Multi-step Look-Ahead Policy for Autonomous Cooperative Surveillance by UAVs in Hostile Environments

X. TIAN
Y. BAR-SHALOM
K. R. PATTIPATI

In this paper a real-time cooperative path decision algorithm for UAV surveillance is proposed. The surveillance mission includes multiple objectives: i) navigate the UAVs safely in a hostile environment; ii) search for new targets in the surveillance region; iii) classify the detected targets; iv) maintain tracks on the detected targets. To handle these competing objectives, a *layered decision framework* is proposed, in which different objectives are deemed relevant at different decision layers according to their priorities. Compared to previous work, in which multiple objectives are integrated into a single global objective function, this layered decision framework allows detailed specification of the desired performance for each objective and guarantees that an objective with high priority will be better satisfied by eliminating possible compromises from other less important ones. In addition, specific path decision strategies that are suited to the individual objectives can be used at different decision layers. An important objective of the path decision algorithm is to navigate the UAV safely in the hostile environment. To achieve this, it is shown necessary to increase the time horizon of the path decisions. In order to overcome the computational explosion of an optimal multi-step look-ahead path decision strategy, a Rollout Policy is proposed. This policy has moderate complexity and, when used in the layered decision framework, it is able to find safe paths effectively and efficiently. When the number of UAVs is large, the formation of UAV decision groups based on a nearest neighbor rule is proposed to control the complexity of the path decision algorithm. Further flexibility of assigning different objectives to the UAVs is also discussed. Simulation results show that the proposed path decision algorithm can guide the group of UAVs efficiently and safely for the multi-objective surveillance mission.

Manuscript received January 10, 2008; revised October 7, 2008; released for publication October 30, 2008.

Refereeing of this contribution was handled by Stephane Paradis.

Authors' address: Dept. of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157, E-mail: ({xin.tian, ybs, Krishna}@engr.uconn.edu).

1557-6418/10/$17.00 © 2010 JAIF

1. INTRODUCTION

Recently a considerable amount of research effort has been directed toward the navigation and cooperative control of groups of unmanned (or uninhabited) aerial vehicles (UAVs). The advantages of UAVs include greater mobility, removal of risk to human operators, potentially lower cost, smaller size/weight, and the possibility of effective coordination. These features make them ideal for repetitive or dangerous tasks in both military and civilian applications [17]. A number of UAV management algorithms that serve various applications can be found in the literature. Ref. [14] addresses the problem of cooperatively controlling multiple UAVs so that they reach a predetermined target location simultaneously, while maximizing the survivability of the UAVs against exposed threats and adhering to the fuel constraints. A hierarchical decision mechanism is proposed in which at the team level the estimated time until arrival is computed and at the UAV level path planning is performed. In [5] a similar approach, which includes a Voronoi diagram in path planning, is used for the simultaneous intercept problem in the presence of dynamic threats. Similar approaches can be found in [6, 13, 15]. In [8] another hybrid control structure is proposed for the simultaneous intercept problem. Ref. [22] studies the task assignment problem for a group of UAVs.

We focus on the surveillance application of UAVs. The scenario considered involves a group of UAVs that search and track ground moving targets in a hostile environment. The objectives in the surveillance mission include: i) navigate the UAVs safely in a hostile surveillance environment; ii) search for new targets; iii) classify the detected targets; iv) maintain tracks on the detected targets. The conventional method of handling multiple objectives is to construct a combined objective function, e.g., the weighted sum approach used in [16, 23]. Our previous work [18] also uses the weighted sum approach, where the path decision problem is formulated as a nonlinear programming problem and solved by optimizing the global objective function over the continuous control variables (turn rates of the UAVs). However, there are several drawbacks to this. First, since different objectives have different meanings, the weighted sum of the objective functions is difficult to interpret and validate. Second, when the path decisions are made by maximizing a combined global objective function, it is hard to specify the requirements for the individual objectives. For example, for the objective tracking, it would be more reasonable to require the RMS errors of a given target to be no larger than a predefined level as opposed to requiring the errors to be as small as possible. Third, the simultaneous impact of multiple objectives on the path decisions could compromise the satisfaction of one or another objective in an unpredictable manner. As shown in Section 4, the survival probabilities of the UAVs can drop significantly when a combined global objective function is used for path decisions. Ref. [11] proposes an algorithm for the design of the weights for the weighted sum approach in order to achieve a desirable tradeoff in the different objectives. However, the computational requirements of this algorithm are too involved for complex cooperative tasks and may preclude the possibility of multi-step look-ahead policies.

In this paper, a novel approach—*layered decision framework*—is proposed to handle multiple objectives in a surveillance mission. In the layered decision framework, instead of combining different objectives into a single objective function, multiple objectives are in separate decision layers according to their priorities. The control options[1] are evaluated first in the top decision layer, which results in a subset of controls that yields satisfactory results for the primary objective. Then, this subset of controls are passed down to the next decision layer for further selection. Proceeding through the decision layers, the control options are sifted and reduced to the final decision with the best overall performance. Major benefits of this approach include: i) it allows the specification of desired performance for each individual objective in different layers; ii) an objective with a higher priority will be better satisfied by eliminating possible compromises from other less important ones; iii) depending on the nature of the objective, suitable path decision strategies can be used at each decision layer, which may lead to significant savings in computation. iv) computation can be saved when the path decisions can be made through some of the decision layers, because there is no need to evaluate the remaining less important ones.

In the surveillance problem considered, the objective of safe navigation is assigned the highest priority; this is based on the premise that the safety of a UAV is more important than gathering one extra measurement. The problem of navigating a single UAV in a hostile environment while chasing a target has been studied in [25], in which the UAV tries to stay within a defined proximity of its target while avoiding restricted regions and obstacles. A gradient search algorithm with a geometry based strategy is used for the path decisions. In the present paper, the threats come not only from fixed positions, but also from moving targets. A different approach based on a Rollout Policy [3] is proposed for path decisions. This is used in the decision layer for safe navigation and is shown to be able to find safe path decisions effectively with moderate complexity. For other surveillance objectives including—search, classification and tracking, following [18], specific objective functions are constructed based on certain information criteria. These objective functions, along with suitable path decision strategies, form the rest of the decision layers.

[1]The control variables are discretized into a set of control options. As shown later, this also facilitates the multi-step look-ahead path decision policy.

Two additional features are also incorporated into the path decision algorithm, which make it more efficient and flexible for a large number of UAVs. One is the formation of decision groups based on a nearest neighbor rule. By doing this, the complexity of the algorithm increases linearly with the number of UAVs. Another is to assign different objectives to UAVs, since it is common to require the UAVs to focus on different tasks in the surveillance region.

The paper is organized as follows. Section 2 formulates the surveillance problem. Section 3 is devoted to the layered decision framework for surveillance with multiple objectives. In Section 4, the problem of safe navigation in a hostile environment is studied. The multi-step look-ahead path decision strategy is proposed using a Rollout Policy, and it is shown to be effective in solving the problem of safe navigation. In Section 5, the construction of small decision groups and how to assign different objectives to the UAVs are discussed. Simulation results are also presented to show the effectiveness of the algorithm. Section 6 presents the conclusions.

2. THE SURVEILLANCE MODELS AND OBJECTIVE FUNCTIONS

In this paper, the surveillance scenario follows mostly [19]. To make this paper self-contained, all models used are described in this section, including UAV specifications, models for threats in the surveillance region, as well as, tracking, search and classification models. It is worth mentioning that, for the sake of simplicity, we assume the surveillance mission takes place in a 2-D plane, namely, altitudes of the UAVs are not taken into account. This, however, does not compromise the main ideas of the paper, which are the *layered decision framework* and the *multi-step look-ahead path decision policy* for UAV navigation.

2.1. UAV Characteristics

Assume that fixed wing unmanned aerial vehicles are used for surveillance. The UAVs can fly only within a speed interval and have limited maneuverability. Following the formulation in [18], it is assumed that the UAVs move with a constant speed V_{uav} and the maximum turn rate the UAVs can take is Φ_{max}. Unlike in [18], the control of the UAVs is discretized into D levels, namely the UAVs can only take turn rates from a finite set. For example, when $D = 3$ the control set is $\{-\Phi_{max}, 0, \Phi_{max}\}$. It is assumed that the path decisions are made every T seconds. For cooperation, the UAVs need to exchange information of their states and measurements from the onboard sensors. In this paper, a centralized data processing framework is used, that is all the information from the UAV network is available for data fusion and path decisions. While the proposed path decision algorithm works best in a centralized setting, it can be used in a distributed system by treating

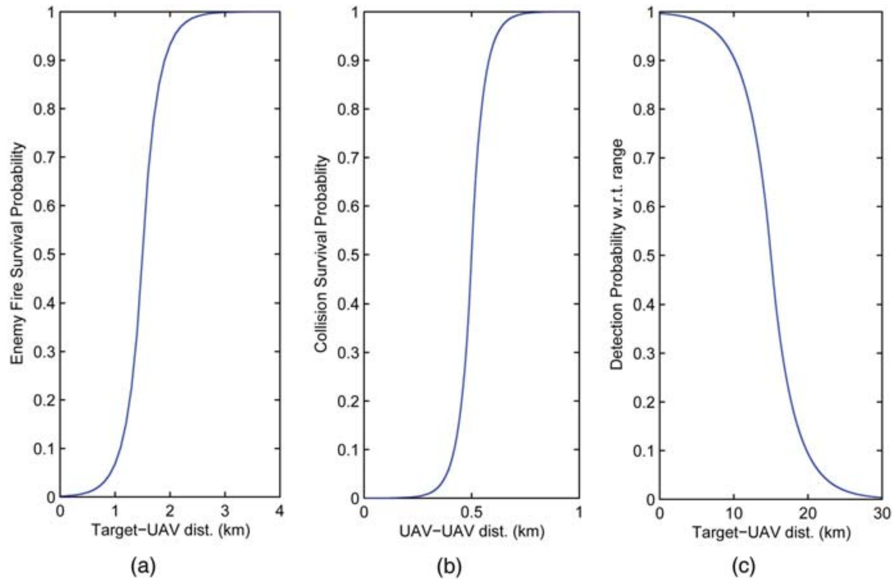


Fig. 1. Detection and survival probabilities as functions of distance to target.

each individual UAV as a duplication of the decision center. The issue of synchronizing information among distributed agents (UAVs) in a distributed system is beyond the scope of this paper.

2.2. The Model of the Threats in the Surveillance Region

In the surveillance, UAV losses may happen due to hostile fire and collisions among the UAVs. As in [18], these potential threats are incorporated into the survival probabilities of the UAVs. The survival probability of UAV s equals the product of target-fire survival probability $\pi_S^1(s)$, stationary-attack survival probability $\pi_S^2(s)$, and collision survival probability $\pi_S^3(s)$, i.e.,

$$\pi_S(s) = \pi_S^1(s)\pi_S^2(s)\pi_S^3(s) \tag{1}$$

where $\pi_S^1(s)$, is the product of target-fire survival probabilities of UAV s in view of each target j, i.e.,

$$\pi_S^1(s) = \prod_j \pi_S^1(s,j). \tag{2}$$

Similarly, for the attacks from stationary threats, survival probability of the UAV is

$$\pi_S^2(s) = \prod_l \pi_S^2(s,l) \tag{3}$$

and $\pi_S^3(s)$ is the product of collision survival probabilities corresponding to all other UAVs,

$$\pi_S^3(s) = \prod_{i:i\neq s} \pi_S^3(s,i). \tag{4}$$

The nature of these survival probabilities is application-dependent. In this paper, the probabilities are modeled as functions of distance as shown in Fig. 1: (a) for π_S^1 and π_S^2 and (b) for π_S^3. For safe navigation, the survival probabilities of the UAVs should be above a threshold,[2] e.g., 0.9, which is a design parameter of the algorithm.

2.3. The Tracking Model

Using a 2-D model, the kinematic state of the target is defined as

$$X = [x \quad \dot{x} \quad y \quad \dot{y}]'. \tag{5}$$

The target motion is modeled by the Discrete White Noise Acceleration (DWNA) model [1]. The UAVs are assumed to be equipped with Ground Moving Target Indicator (GMTI) radars, which measure the locations of moving ground targets as well as their radial velocities (Doppler). A 2-D measurement model is used

$$r_m = r + w_r \tag{6}$$

$$\alpha_m = \alpha + w_\alpha \tag{7}$$

$$\dot{r}_m = \dot{r} + w_{\dot{r}} \tag{8}$$

in which w_r, w_α and $w_{\dot{r}}$ are Gaussian noise with standard deviations σ_r, σ_α and $\sigma_{\dot{r}}$ respectively. Applying the Polar to Cartesian conversion [1], the measurement is converted to

$$Z_m = [x_m \quad y_m \quad \dot{r}_m] \tag{9}$$

where

$$x_m = r_m \cos\alpha_m \tag{10}$$

$$y_m = r_m \sin\alpha_m. \tag{11}$$

[2]This threshold serves as a soft boundary, the path decision algorithm should be able to keep the survival probabilities above or close to this safety bound.

The noise in the converted measurement is zero-mean[3] with covariance matrix

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} & 0 \\ R_{1,2} & R_{2,2} & 0 \\ 0 & 0 & \sigma_{\dot{r}}^2 \end{bmatrix} \tag{12}$$

where

$$R_{1,1} = r_m^2\sigma_\alpha^2 \sin\alpha_m^2 + \sigma_r^2\cos\alpha_m^2 \tag{13}$$

$$R_{2,2} = r_m^2\sigma_\alpha^2 \cos\alpha_m^2 + \sigma_r^2\sin\alpha_m^2 \tag{14}$$

$$R_{1,2} = (\sigma_r^2 - r_m^2\sigma_\alpha^2)\sin\alpha_m\cos\alpha_m. \tag{15}$$

The observation matrix corresponding to (9) is (see, e.g., [24])

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \cos\alpha & 0 & \sin\alpha \end{bmatrix} \tag{16}$$

where α_m can be used in H, as shown in [24].

The detection probability is a function of range and range rate of the target with respect to the GMTI radar. Denoted as π_D, the detection probability is given by

$$\pi_D = \pi_D^1(r)\pi_D^2(\dot{r}) \tag{17}$$

$\pi_D^1(r)$ is shown in Fig. 1(c). For a GMTI radar, if the range rate for a target falls below a threshold $\dot{r}_{\min}$ then the target will not be detected. Hence,

$$\pi_D^2(\dot{r}) = 1 - \text{Prob}\{-\dot{r}_{\min} < \dot{r} < \dot{r}_{\max}\}. \tag{18}$$

For tracking, it is assumed that the UAVs obtain measurements from the detected targets every T seconds. Following [18], at decision time kT, the expected track update for target j at time $(k+1)T$ is

$$\hat{I}_j(k+1 \mid k+1) = I_j(k+1 \mid k) + \sum_{s=1}^{N} \hat{\pi}_D(s,j,k+1)\hat{H}(s,j,k+1)' \times \hat{R}(s,j,k+1)^{-1}\hat{H}(s,j,k+1) \tag{19}$$

in which I_j denotes the information matrix from the track of target j, namely, $I_j = P_j^{-1}$; N is the number of the UAVs; s is the index of the UAV. The "hat" marks indicate the values are expectations that depend on the relative positions of target j to the UAVs at $k+1$. Clearly, $\hat{I}_j(k+1 \mid k+1)$ is a function of the collective path decisions (controls) of the UAVs at k. To evaluate the expected quality of the track, the expected mean square position error can be used, since it is directly related to the RMS position error (components 1 and 3 of the state vector). For target j one has,

$$\widehat{\text{MSE}}(j,k+1) = \hat{P}_j(k+1 \mid k+1)_{(1,1)} + \hat{P}_j(k+1 \mid k+1)_{(3,3)}. \tag{20}$$

[3]Since the condition for the unbiasedness conversion [1] is satisfied, the noises in x_m and y_m can be assumed to be zero-mean.

The construction of the objective function for tracking is based on (20). Further details will be discussed in Section 3.

2.4. The Model for Search

Studies on the problem of cooperative search using multiple autonomous UAVs can be found in [10, 9, 16]. For different applications, formulations of the problem may change. In the surveillance problem considered, the surveillance region is divided into a number of sectors. It is assumed that each UAV scans a fixed number N_s of such sectors in each period of its operation. As in [18], in each sector, the arrival of new targets is modeled as a Poisson process.

Let $P_{m,n}(k)$ denote the probability that there is no new target in sector $\{m,n\}$ and $\lambda_{m,n}$ denote the Poisson parameter (expected spatial density of new targets) of this sector. At the kth decision time one has

$$P_{m,n}(k) = P_{m,n}(k-1)\exp^{-\lambda_{m,n}T}. \tag{21}$$

If the sector is scanned by UAV s at k with a detection probability of $\tilde{\pi}_D(m,n,s,k)$, it follows (assuming there are no false alarms) that the updated probability $P_{m,n}(k^+)$ is given by

$$P_{m,n}(k^+) = \begin{cases} \dfrac{P_{m,n}(k)}{P_{m,n}(k) + [1-P_{m,n}(k)][1-\tilde{\pi}_D(m,n,s,k)]} & \text{if scanned and no target was detected} \\ 1 & \text{if scanned and a target was detected} \end{cases}. \tag{22}$$

An intuitive interpretation of (22) is as follows. When scanned, no target is detected with probability $P_{m,n}(k) + [1-P_{m,n}(k)][1-\tilde{\pi}_D(m,n,s,k)]$. So, the updated probability $P_{m,n}(k^+)$ is as given in the first probability of (22). If a target is detected, no new target is in that sector with probability 1. From (22), the payoff of a specific scan can be calculated as

$$\begin{aligned} \Delta(m,n,s,k) &= \text{E}[P_{m,n}(k^+)] - P_{m,n}(k) \\ &= [1-P_{m,n}(k)]\tilde{\pi}_D(m,n,s,k). \end{aligned} \tag{23}$$

For a single UAV its scan decision can be made by selecting the most profitable (largest Δ given by (23)) N_s sectors, which favors the sectors that are more likely to have new targets (low $P_{m,n}(k)$) and the potential new targets are more likely to be detected (high $\tilde{\pi}_D(m,n,s,k)$). In the multiple UAV case, the optimal scan decision is a complicated assignment problem. However, a near-optimal solution can be found using simple heuristics. Since the UAVs tend to operate in different regions (to produce good coverage to the whole surveillance area), their N_s best sectors to scan are very unlikely to overlap, which allows the UAVs to make their scan decisions independently; rare conflicts can be resolved by making their scan decisions sequentially in the order of the UAV indices.

2.5. The Model for Target Classification

An important objective of surveillance is to classify the detected targets. Studies of optimal search with joint detection and classification can be found in [20, 6, 12]. An integrated algorithm for tracking and classification with data association is presented in [2]. In this paper, following [19], the classification and tracking are considered as different problems as it is assumed that the GMTI radar mounted on the UAV does not provide classification information; instead, it is assumed that classification information is provided by a closed circuit digital (CCD) camera. A classifier associated with the camera processes the data from the camera. The outputs are class decisions and their associated class confusion matrix. The class confusion matrix gives the probabilities of class decision output given the actual class of the target. It is assumed that the class confusion matrix is only a function of target-UAV distance, i.e., spatial diversity does not improve classification results. Let $\zeta(j,s,k)$ denote the output of the classifier on UAV s at the kth decision time for target j and $C(j,s,k)$ be the corresponding class confusion matrix. Element $c_{ab}(j,s,k)$ of $C(j,s,k)$ is given by

$$c_{ab}(j,s,k) = \mathrm{P}(\zeta(j,s,k) = b \mid \kappa_j = a) \tag{24}$$

in which κ_j denotes the true class of the target.

To facilitate classification, when a new target is detected, the UAV closest to that target is assigned to perform the classification. The UAV will start to use the classification sensor when it gets close enough to the target. Notice that classification is a special case when the UAVs are focusing on different objectives. Such needs are common in multiple UAV surveillance. For example, some of the UAVs may focus on tracking while others focus on search. In Section 5.2, the problem of assigning different objectives to the UAVs will be discussed.

For a detected target, a class probability vector is used as the state for classification. Let μ_j denote the class probability vector for target j which can be initialized, e.g., as a uniform distribution over all possible classes. If the output of the classifier is $\zeta(j,s,k) = b$, then μ_j is updated as [2]

$$\mu_j^+ = \frac{C_b(j,s,k) \otimes \mu_j}{C_b(j,s,k)'\mu_j} \tag{25}$$

where $C_b(j,s,k)$ is the bth column of the class confusion matrix and $\otimes$ is the Schur-Hadamard product (term by term). The classification of target j is completed when

$$\max\{\mu_j\} > \tau_{\mathrm{CLS}} \tag{26}$$

in which τ_{CLS} is a confidence threshold, e.g., 0.95.

TABLE I
Decision Layers in the Path Decision Algorithm for Surveillance
(s is the index of the UAVs and j is the index of targets)

Objective	Decision Layer (priority)	Satisfactory Level	Evaluation Criterion for the Accomplishment
Safe Navigation	1	τ_{PS}	$\min\{\pi_S(s), \tau_{\mathrm{PS}}\}$
Classification	2	τ_{CLS}	$\min\{\max\{\mu_j\}, \tau_{\mathrm{CLS}}\}$
Tracking	3	$\tau_{\mathrm{MSE}}(j)$	$\max\{\mathrm{MSE}(j), \tau_{\mathrm{MSE}}(j)\}$
Search	4	τ_{PNNT}	$\min\{P_{m,n}, \tau_{\mathrm{PNNT}}\}$

3. LAYERED DECISION FRAMEWORK FOR SURVEILLANCE MISSION WITH MULTIPLE OBJECTIVES

In this paper, a layered decision framework is used for handling multiple objectives, in which each objective occupies a decision layer according to its priority. A decision layer consists of: i) the objective; ii) a function that evaluates the degree of accomplishment of the objective; iii) a satisfactory level, at which point no further improvement on the objective is necessary. Table I shows an example of arrangement of the decision layers.

In the layered decision framework, an objective with a higher priority will be considered first. The key idea is that once a satisfactory level is reached, the "satisfied" objective will have no effect on the path decisions, thus freedom in the path decisions can be passed on to the next decision layer. To illustrate this, consider a simple case of a group of $N = 2$ UAVs tracking two targets while performing search in the surveillance region (classification is omitted in this example). Suppose the control of each UAV is discretized into $D = 3$ levels. Thus, at every decision epoch, the number of control options for the UAV group is $D^N = 9$. For simplicity, the example will stay with one-step look-ahead path decision (multi-step will be introduced later) and all the data in this example are for the purposes of illustration only.[4]

In this example, the control options are first evaluated by the top decision layer of safe navigation. Table II shows the m-best control options ($m = 5$ in this case) indicated by a check mark.[5]

When $m = 1$, this is the control option that yields the best result for the current objective and it is chosen directly as the path decision, since there is no freedom in control left for the remaining decision layers. If $m > 1$, these m best control options will be passed on to the next decision layer of tracking. As shown in Table III, the

[4]In actual simulations, the differences between different control options are much smaller than those shown in this example. However, by always following the best control option, the UAVs will navigate to desired positions by capturing the gradient information of the objective functions.

[5]In Tables II–III, control index (C_1, C_2) denotes a combination of the controls taken by the two UAVs, $C_1 \in \{1,2,3\}$ for UAV 1, $C_2 \in \{1,2,3\}$ for UAV 2.

TABLE II
Decision Layer 1: Control decisions for Safe Navigation with $N = 2$ UAVs and $\tau_{PS} = 0.9$

Control Index (C_1,C_2)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
$\hat{\pi}_S$ (Expected π_S at $k+1$)	0.92 1	0.95 0.98	0.87 0.91	0.99 0.92	0.82 0.87	0.81 0.95	0.93 1	0.83 0.91	0.91 0.92
$\bar{\pi}_S$ $(\min\{\hat{\pi}_S,\tau_{PS}\})$	0.9 0.9	0.9 0.9	0.87 0.9	0.9 0.9	0.82 0.87	0.81 0.9	0.9 0.9	0.83 0.9	0.9 0.9
Control Evaluation $\left(\prod \bar{\pi}_S(s)\right)$	0.81 √	0.81 √	0.783	0.81 √	0.713	0.729	0.81 √	0.747	0.81 √

TABLE III
Decision Layer 2: Control decisions for Tracking with $\tau_{MSE} = 25$ m^2

Control Index (C_1,C_2)	(1,1)	(1,2)	(2,1)	(3,1)	(3,3)
$\widehat{\mathrm{MSE}}$(m^2) (Expected MSE at $k+1$)	17 23	20 22	30 27	32 29	23 31
$\overline{\mathrm{MSE}}$(m^2) $(\max\{\widehat{\mathrm{MSE}},\tau_{MSE}\})$	25 25	25 25	30 27	32 29	25 31
Control Evaluation $\left(\sum \overline{\mathrm{MSE}}(j)\right)$	50 √	50 √	57	61	56

output of the second decision layer is a further reduced set of controls indicated by columns with a check mark. If the size of this reduced control set is greater than 1, it will be passed to the next decision layer for further selection.

The path decision algorithm ends when the best control option is found. The uniqueness of the final path decision can be guaranteed by simply setting the "satisfactory level" of the last decision layer to the "ideal" one. In this example, the last decision layer is "Search," thus τ_{PNNT} can be set to 1, which is an "ideal" level that can never be simultaneously achieved at all the sectors due to the limited scan capability of the UAV group. A similar procedure as in Tables II and III can be used for "Search" and it is omitted here for conciseness.

Compared to the weighted sum approach, the layered decision framework has the following advantages:

- Multiple objectives in the surveillance are clearly delineated. Thus, objectives with higher priorities are free from possible compromises from the less important ones. Section 4 will show that this is particularly important for the objective of safe navigation.
- For each objective, the "satisfactory" levels allow the path decision algorithm to be sensitive to the entities (e.g., targets in the tracking layer, sectors in the search layer) that demand more attention. Take tracking for example. The objective function is a combination of sub-objectives related to the tracks of all the targets. The use of the satisfactory level τ_{MSE} eliminates the impact of those sufficiently accurate tracks and allows the inaccurate tracks have more influences on the path decisions.
- The layered decision framework allows different path decision strategies to be used for the objectives. For example, depending on the nature of the objectives, they may or may not benefit from multi-step look-ahead strategies. Significant computation cost can be saved by decomposing the objectives among multiple decision layers.
- When a path decision is determined by the first few decision layers, the remaining layers do not need to be evaluated.

4. MULTI-STEP LOOK-AHEAD PATH DECISION STRATEGY FOR UAV NAVIGATION

An important objective for the path decision algorithm is to navigate the UAV group safely in the surveillance region. As specified in Section 2.2, the threats to the UAVs are modeled in terms of survival probabilities (1). In [18] the survival probabilities of the UAVs are incorporated into the global objective function through the track update as

$$\hat{I}_j(k+1 \mid k+1) = I_j(k+1 \mid k) + \sum_{s=1}^{N} \hat{\pi}_S(s,k+1)\hat{\pi}_D(s,j,k+1) \times \hat{H}(s,j,k+1)'\hat{R}(s,j,k+1)^{-1}\hat{H}(s,j,k+1) \tag{27}$$

which is a variation of (19). If the UAV survival probabilities, $\hat{\pi}_S(s,k+1)$, drop, there will be a reduction in the expected information gain. As a result, the path decision algorithm tends to avoid drops in the survival probabilities of the UAVs. While this formulation intuitively makes sense, it turns out to be incapable of preventing the UAV survival probabilities from significant drops. There are two reasons for this problem. First, tracking and safe navigation are two competing objectives. Particularly when a UAV is tracking a single target it tends to get close to the target, while safe navigation requires the UAV to keep adequate distance from the target. The combination of competing objectives into a single global objective function can lead to unpredictable compromises. Second, due to limited maneuverability of the UAV, a one-step look-ahead path decision strategy can result in late detections of potential safety risks. In the rest of this section, a multi-step

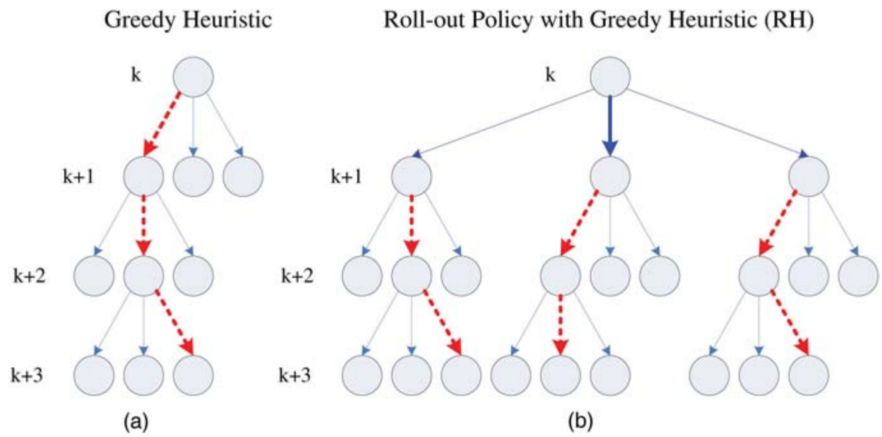


Fig. 2. Greedy Heuristic and Rollout Policy.

look-ahead path decision strategy is proposed based on the Rollout Policy [3]. When used in the decision layer of safe navigation, it is shown to produce significantly improved results.

4.1. Multi-step Look-ahead Path Decision and Rollout Policy

By discretizing the controls of the UAVs, a multi-step look-ahead path decision for the UAV group can be modeled as a combinatorial optimization problem. However, the problem is NP-hard, e.g., for a UAV group that consists of N UAVs, the optimal solution for a K-step look-ahead path decision needs to consider D^{NK} possible paths, which can be far too expensive for a real-time algorithm even with modest N and K. Instead of seeking the optimal solution, a suboptimal solution requiring less computation is much more desirable. The Rollout policy proposed in [3] is a suboptimal solution to the combinatorial optimization problems. Based on a heuristic solution to the problem (called a base heuristic), the Rollout policy is guaranteed to find a solution that is no worse than the base heuristic. Successful applications of the Rollout policy can be found in [4, 21], in which it works surprisingly well by producing near-optimal solutions.

In [3], the Rollout Policy was introduced in a Dynamic Programming (DP) context. Consider a problem with a finite set of feasible solutions and a cost function $g(u)$, $u \in U$. Each u has K components, namely, $u = (u_1, u_2, \ldots, u_K)$. In the K-step look-ahead path decision algorithm, the components $u_1, \ldots, u_K$ correspond to the controls at different times. An i-tuple $(u_1, u_2, \ldots, u_i)$, $i < K$, consisting of i components of the solution is called an i-solution. The optimal solution $u^* = (u_1^*, u_2^*, \ldots, u_K^*)$ can be obtained via DP, which gives

$$u_i^* = \arg\left\{\min_{u_i \in U_i(u_1^*, u_2^*, \ldots, u_{i-1}^*)} J^*(u_1^*, u_2^*, \ldots, u_{i-1}^*, u_i)\right\}, \quad i = 1, 2, \ldots, K \qquad (28)$$

where J^* is the optimal cost-to-go function for any i-solution. However, the evaluation of J^* is, in most cases, not feasible. In the Rollout policy, a base heuristic algorithm H is used. From any i-solution $u = (u_1, u_2, \ldots, u_i)$, the heuristic algorithm H can generate a complete K-solution $u = (u_1, u_2, \ldots, u_K)$ whose cost is denoted by $h(u_1, u_2, \ldots, u_i)$. The suboptimal solution $\tilde{u} = (\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_K)$ is found by replacing J^* in (28) with the heuristic cost-to-go function h, namely,

$$\tilde{u}_i = \arg\left\{\min_{u_i \in U_i(\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_{i-1})} h(\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_{i-1}, u_i)\right\}, \quad i = 1, 2, \ldots, K. \qquad (29)$$

For the K-step look-ahead path decision strategy, at k, the control that produces the best control sequence from $k+1$ to $k+K$ is selected; the Greedy heuristic, which is equivalent to the one-step look-ahead path decision, is used as the base heuristic to generate the control sequences. Fig. 2 illustrates the greedy heuristic and its corresponding Rollout policy in a 3-step look-ahead path decision strategy for a single UAV.

Assume that at each node, there are 3 controls (turn rates) available for the UAV. Using the Greedy heuristic, the control that leads to the next "node" with the best immediate result will be selected. Fig. 2(a) shows the path (control sequence) from k to $k+K$ $(=k+3)$ generated by Greedy heuristic (highlighted by the thick dashed arrows). In the Rollout policy, instead of starting from k, the greedy heuristic starts from $k+1$ to generate the remaining paths to $k+3$. The control at k that produces the best path to $k+3$ (highlighted by the thick dashed arrows in Fig. 2(b)) will be selected as the control decision. Note that the evaluations of the paths from k to $k+K$ are based on the information available at k and the procedure is repeated at every decision time with updated information. Compared to the exhaustive search which requires one to evaluate $\sum_{i=1}^{K} D^{N\cdot i}$ "nodes," the Rollout policy only evaluates $D^N + (K-1)D^{2N}$ nodes. The computational cost increases linearly with the decision horizon K.

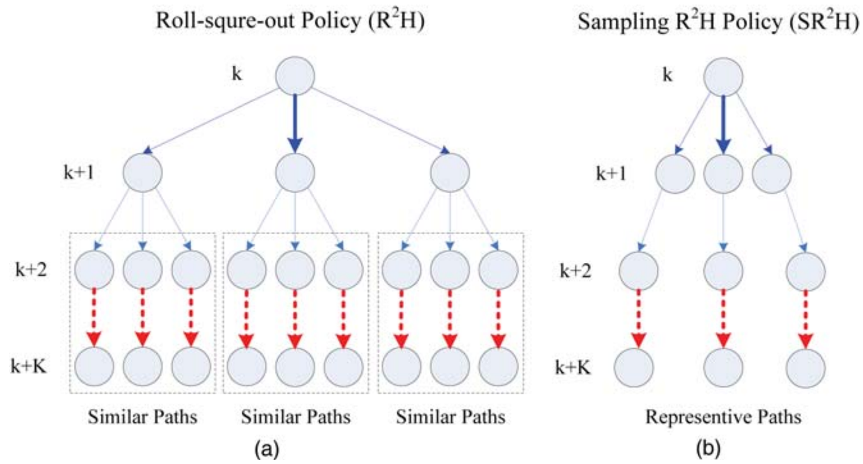


Fig. 3. Roll-square-out Policy and Sampling Roll-square-out Policy.

A variation of the Rollout policy RH is the Roll-square-out policy [3] (denoted as R^2H). As shown in Fig. 3(a), in R^2H the greedy heuristic (indicated by the dashed arrows) starts from $k+2$. R^2H (which needs to evaluate $D^N + D^{2N} + (K-2)D^{3N}$ "nodes") is more expensive than RH, while its results are guaranteed to be no worse than RH. In view of the specific feature of the path decision problem, it is reasonable to assume two close paths will produce similar performance. In Fig. 3(a), the dotted squares mark out 3 similar solution sets. By taking representative sample paths from the similar solution sets, R^2H can be simplified to a sampling R^2H policy (SR^2H). As illustrated in Fig 3(b) SR^2H is much less expensive than R^2H, which requires one to evaluate only $2D^N + (K-2)D^{2N}$ nodes. SR^2H is useful in UAV path decisions, since it increases the volume of the search space for optimal paths (the K-step look-ahead increases the time horizon of the search).

4.2. The Decision Layer for Safe Navigation

The proposed multi-step look-ahead path decision strategy (see Section 4.1) can be used in any decision layer in the layered decision framework (see Section 3). Instead of seeking one best control at k, at each decision layer, the path decision algorithm looks for m best controls which will be passed on to the next decision layer for further selection. An important issue in a K-step look-ahead path decision algorithm is to evaluate and compare the control sequences from k to $k+K$. Figs. 2–3 show that the evaluation of a control sequence from k to $k+K$ requires the evaluations of the nodes from $k+1$ to $k+K$. In the layer of safe navigation, a node at $k+i$ can be evaluated by

$$\hat{J}_S(k+i) = \sum_s \ln(\min\{\hat{\pi}_S(s,k+i), \tau_{PS}\}) \tag{30}$$

where s is the index of the UAVs and τ_{PS} is the satisfactory level introduced in Table II. Accordingly, the evaluation of a control sequence from k to $k+K$ is given by

$$\sum_{i=1}^{K} \hat{J}_S(k+i). \tag{31}$$

In addition, a control sequence is considered to be "safe" if the expected survival probabilities of the UAVs are above τ_{PS} along the path, namely,

$$\min_s\{\hat{\pi}_S(s,k+i)\} \geq \tau_{PS} \qquad \forall \quad i = 1,\ldots,K. \tag{32}$$

Therefore, all "safe" control sequences have the same value (31), namely

$$\sum_{i=1}^{K} \hat{J}_S(k+i) = KN\ln(\tau_{PS}). \tag{33}$$

Based on the above definitions, at the kth decision time, the procedure for a K-step look-ahead path decision algorithm for safe navigation is as follows:

- Use the Rollout Policy to generate control sequences from k to $k+K$.
- If "safe" control sequences that satisfy (33) are detected, pass the corresponding controls at k to the next decision layer.
- If no "safe" sequence is found, use the Sampling Rollout strategy to generate control sequences from k to $k+K$.
- If "safe" sequences are detected, pass the corresponding controls at k to the next decision layer.
- If still no "safe" sequence is found, the value of u_k that leads to the "best" control sequence (evaluated using (31)) is selected. The evaluations in the remaining decision layers are not needed.

4.3. Simulation Results for UAV Safe Navigation: Rollout vs. One-step Look-ahead

Consider first a "toy example" in which one UAV searches for and tracks one target. For simplicity, clas-

TABLE IV
Decision Layers in the Simulation

Objective	Decision Layer (priority)	Satisfactory Level	Evaluation Criterion for the Accomplishment	Strategy for Path Decision
Safe Navigation	1	$\tau_{\text{PS}} = 0.9$	$\min\{\pi_S(s), \tau_{\text{PS}}\}$	multi-step
Tracking	2	$\tau_{\text{MSE}} = 0 \text{ m}^2$	$\max\{\text{MSE}(j), \tau_{\text{MSE}}\}$	one-step
Search	3	$\tau_{\text{PNNT}} = 1$	$\min\{P_{m,n}, \tau_{\text{PNNT}}\}$	one-step

sification is not included here. Table IV shows the decision layers of the path decision algorithm.[6] Note that τ_{MSE} in the tracking layer is set to zero, which means once the target is detected the UAV will "focus" on tracking. The surveillance region is 40 $km \times$ 40 km and is divided into 10×10 sectors. The target starts from $[2000, 14200]$ m with initial velocity $[10, -2]$ m/s. The process noise of the target has intensity $\sqrt{\tilde{q}} = 0.01$ m/s^2. It is assumed that $V_{\text{UAV}} =$ 40 m/s and the control set is $\{-3, 0, 3\}$ deg/s. The on board GMTI radar has measurement standard deviations of [10 m, 1 mrad, 1 m/s]. There are 3 stationary threats located at $[5000, 15000]$ m, $[7000, 7000]$ m and $[20000, 10000]$ m (indicated by the "asterisks"). The circles show the boundaries of the corresponding restricted zones within which the survival probability of the UAV from the threat is below the satisfactory level τ_{PS}. Specifications of the UAV survival probability and the target detection probability follow those in Section 2. Fig. 4 shows trajectories of the UAV and the target in one simulation. In this case the UAV has to circle around the target which is slower while avoiding certain regions.

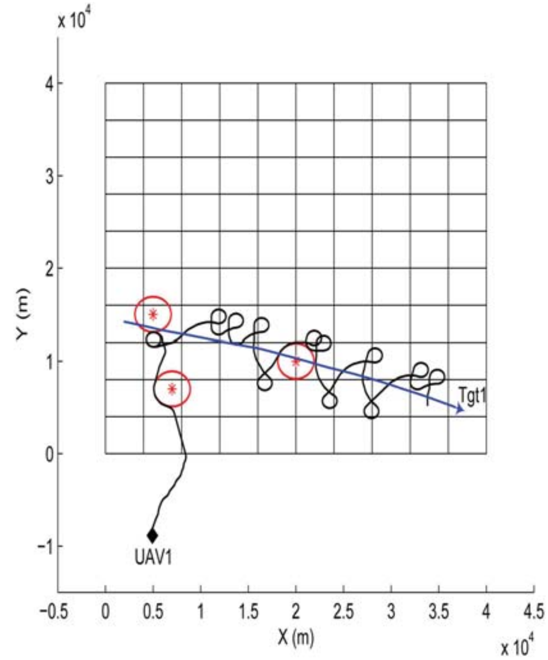


Fig. 4. UAV trajectory in one simulation using the layered decision framework (9-step look-ahead decisions for safe navigation).

For comparison, the combined objective approach, in which the survival probability of the UAV is incorporated into the expected update of the track in (27), is also tested. Notice that in the layered decision framework, safe navigation is treated separately from the objective of tracking; thus, unlike (27), the objective of the expected track update given in (19) does not deal with survival probabilities of the UAVs. A modified version of (27)

$$\begin{aligned}
\hat{I}_j(k+1 \mid k+1) &= \min_s\{\hat{\pi}_S(s,k+1)\} I_j(k+1 \mid k) \\
&\quad + \sum_{s=1}^{N} \hat{\pi}_S(s,k+1)\hat{\pi}_D(s,j,k+1) \\
&\quad \times \hat{H}(s,j,k+1)'\hat{R}(s,j,k+1)^{-1}\hat{H}(s,j,k+1)
\end{aligned} \tag{34}$$

is tested as well, which places greater penalty to the drops in the survival probabilities.

Figs. 5–6 show the minimum survival probability of the UAV over 100 MC runs, in which "combined objective 1" refers to the approach that uses the expected update in (27) as the objective function and "combined objective 2" refers to the approach that uses the expected update (34) as the objective function. As shown in Fig. 5, the one-step look-ahead path decision strategy can not meet the requirement for safe navigation, no matter which objective function for path decision is used. In Fig. 6, although a 9-step look-ahead path decision strategy is used, significant drops in the survival probability of the UAV are still observed in the two combined objective approaches. However the 9-step look-ahead path decision strategy with the layered decision framework is able to keep the survival probability of the UAV close to the satisfactory threshold $\tau_{\text{PS}} = 0.9$. The rare drop to 0.8 occurred only once in the 100 runs. Fig. 7 compares the RMS position errors of the algorithms. Notice that, around the 100th decision time, the layered decision framework has larger RMS position errors than those of the combined objective function approaches, but the drops in the survival probability are avoided, as shown in Fig. 6. This is an example where an objective with higher priority

[6]If the tactical value of the information is very high, safe navigation can be moved to a layer with lower priority.

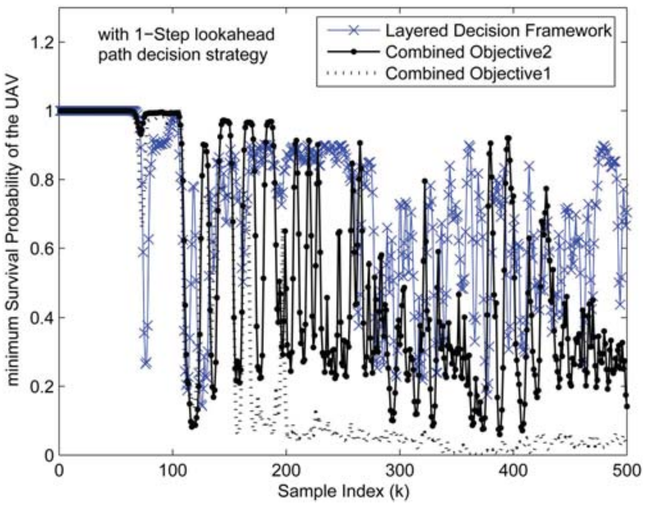


Fig. 5. Minimum survival probability (one-step look-ahead, 100 MC runs).

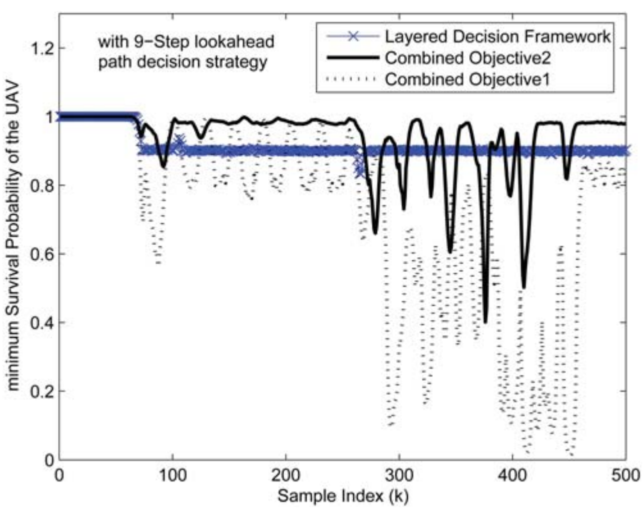


Fig. 6. Minimum survival probability (9-step look-ahead, 100 MC runs).

(safe navigation) will not be compromised by objectives with lower priorities (tracking and search), which is a desirable feature of the layered decision framework. Also notice that, most of the time, the three approaches have no significant differences in the RMS position errors.

5. MULTIPLE UAV COOPERATIVE PATH DECISION ALGORITHM FOR SURVEILLANCE MISSIONS

The multi-step look-ahead path decision algorithm proposed in Section 4 has no limitation on the number of UAVs. However, its complexity increases geometrically with respect to the number of UAVs. To keep the complexity of the path decision algorithm under control, clustering of UAVs into small decision groups will be discussed. Another feature also incorporated is to allow the UAVs to focus on different tasks in the surveillance mission.

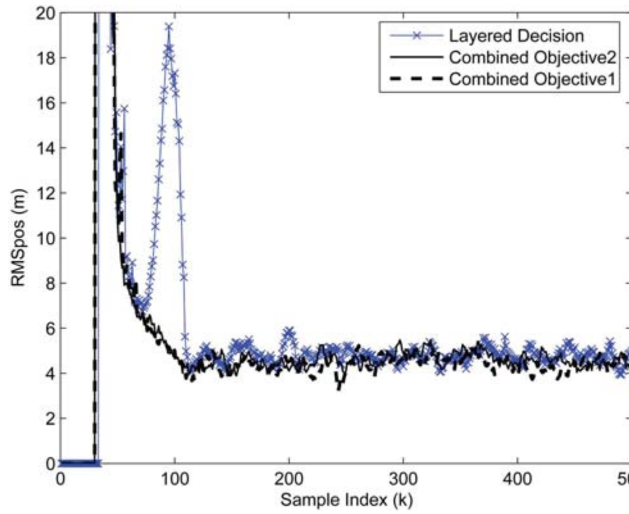


Fig. 7. RMS position error of the track (100 MC runs).

5.1. Formation of Decision Groups Based on a Nearest Neighbor Rule

As discussed in Section 4.1, using the Roll-policy, the K-step look-ahead path decision algorithm needs to evaluate $D^N + (K-1)D^{2N}$ nodes. However, as the number of UAVs increases, the complexity of the algorithm increases geometrically. To avoid this explosion in complexity, the formation of small path decision groups is proposed. For the cooperative path decision problem, it is reasonable to assume that the larger the distance between two UAVs, the less their path decisions are coupled. Thus, to control the number of UAVs involved in each path decision, it is reasonable to: i) set a maximum distance $\text{Dist}_{\max}$ beyond which the two UAVs' path decisions are decoupled; ii) construct small groups for path decisions with maximum number of N_g UAVs based on a Nearest Neighbor Rule (NNR). The NNR can be found in chapter 10 of [7], where it was used for the problem of hierarchical clustering. Fig. 8 is an example of the formation of decision groups for a group of 7 UAVs with $N_g = 3$. The procedure is as follows:

1. Find the "nearest neighbors"[7] of all the ungrouped UAVs. (As shown in Fig. 8, the arrows start from the UAVs point to their "nearest neighbors.")
2. The two UAVs that have the shortest distance to each other form a basic decision group ({UAV 3, UAV 5} in this example).
3. This decision group increases by including a UAV whose "nearest neighbor" is in the decision group. (In this example, both UAV 7 and 4's "nearest neighbors" are inside the basic decision group {UAV 3,

[7]The "nearest neighbor" of a UAV is defined as the closest UAV within a range of $\text{Dist}_{\max}$. Notice that, in the example, the distance of UAV 6 to all the other UAVs is above $\text{Dist}_{\max}$. Therefore, UAV 6 has no "nearest neighbor." Consequently, it forms a decision group by itself.

TABLE V
Decision Layers in the Simulation

Objective	Decision Layer (priority)	Satisfactory Level	Evaluation Criterion for the Accomplishment	Strategy for Path Decision
Safe Navigation	1	$\tau_{\mathrm{PS}} = 0.9$	$\min\{\pi_S(s), \tau_{\mathrm{PS}}\}$	multi-step
Classification	2	$\tau_{\mathrm{CLS}} = 0.95$	$\min\{\max\{\mu_j\}, \tau_{\mathrm{CLS}}\}$	one-step
Tracking	3	τ_{MSE}	$\max\{\mathrm{MSE}(j), \tau_{\mathrm{MSE}}\}$	one-step
Search	4	$\tau_{\mathrm{PNNT}} = 1$	$\min\{P_{m,n}, \tau_{\mathrm{PNNT}}\}$	one-step

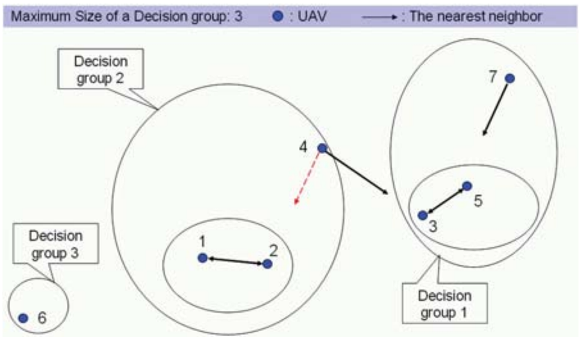


Fig. 8. Formation of decision groups.

UAV 5}; UAV 7 is first selected since it has a shorter distance to UAV 5 than the distance between UAV 4 and UAV 3.)

4. Repeat 3 on the current decision group, until it reaches the capacity limit N_g or there is no unassigned UAV that should be added based on the NNR. (In this example, {UAV 3, UAV 5, UAV 7} form the first decision group.)
5. Repeat this procedure from step 1 for the ungrouped UAVs, until all the UAVs are assigned to their respective decision groups.

For the path decisions in a decision group, the UAVs outside it are assumed to use their latest known controls throughout the path decision procedure; thus, their existence will not increase the complexity of the path decisions in this decision group. By incorporating the mechanism of decision group, the complexity of the path decision algorithm only increases linearly with the number of UAVs.

5.2. Cooperative Path Decision for UAVs with Different Objectives

In practical applications, it might be desirable to allow the UAVs to focus on different tasks. The function of assigning different objectives to the UAVs can be conveniently incorporated into the layered decision framework using satisfactory level matrices. This is illustrated by an example of multi-UAV surveillance with heterogeneous objectives, where some of the UAVs are dedicated to tracking, while the other UAVs focus more on other surveillance tasks. In this case, instead of using a satisfactory level in the decision layer of tracking, a satisfactory level matrix τ_{MSE} is used, whose element $\tau_{\mathrm{MSE}}(s,j)$ specifies the satisfactory level of the track accuracy of target j to UAV s. Thus the desired track accuracy of a target can be different for different UAVs. In the path decision algorithm, once track j is sufficiently accurate to UAV s, that is, $\widehat{\mathrm{MSE}}(j,k) \leq \tau_{\mathrm{MSE}}(s,j)$, a default turn rate (0 rad/s) will be used for UAV s when evaluating the sub-objective function (20) for target j. This makes the sub-objective $\mathrm{MSE}(j,k)$ indifferent to the control evaluations of UAV s, so that freedom in the path decision of UAV s can be saved for other "unsatisfied" objectives.

5.3. Simulation Results

The proposed multiple UAV cooperative path decision algorithm is tested in a similar surveillance scenario as in Section 4.3 but with 4 UAVs and 4 targets. The decision layers of the path decision algorithm are shown in Table V. The UAVs start out searching for targets in the surveillance region. When a target is detected, the UAV that is closest to the target will carry out the classification. Meanwhile the UAV group tracks the target cooperatively. As in Section 5.2, the satisfactory level matrix for tracking τ_{MSE} is a $N \times M$ matrix, where N is the number of the UAVs and M is the number of targets. The components in τ_{MSE} can be set dynamically during the surveillance mission. In the simulation, for the sake of simplicity, a predefined matrix

$$\tau_{\mathrm{MSE}} = \begin{bmatrix} 0 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \end{bmatrix} \mathrm{m}^2 \tag{35}$$

is used. By setting $\tau_{\mathrm{MSE}}(1,1) = 0\ \mathrm{m}^2$, UAV 1 will focus on the tracking of target 1 once it is detected, except when there is a target for it to classify. Fig. 9 shows the trajectories in single run of the simulation. Notice that at the early stage of the simulation, UAV 1 moves farther from target 1 to classify target 4, then it always stays close to target 1, while the other 3 UAVs will not try to stay as close to target 1 due to their relatively low requirements in tracking accuracy.

Fig. 10 shows the minimum survival probabilities of the UAVs. Like the results of the single UAV tracking case in Section 4.3, drops in the survival probabilities are very rare. The drops to about 0.75 occurred only twice over the 100 MC runs. Fig. 11 is the RMS position

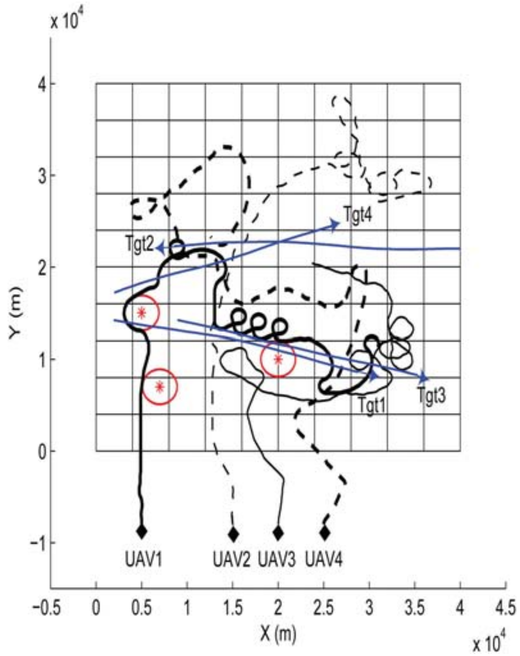


Fig. 9. UAV trajectories in one simulation (three exclusive zones are around the "asterisks").

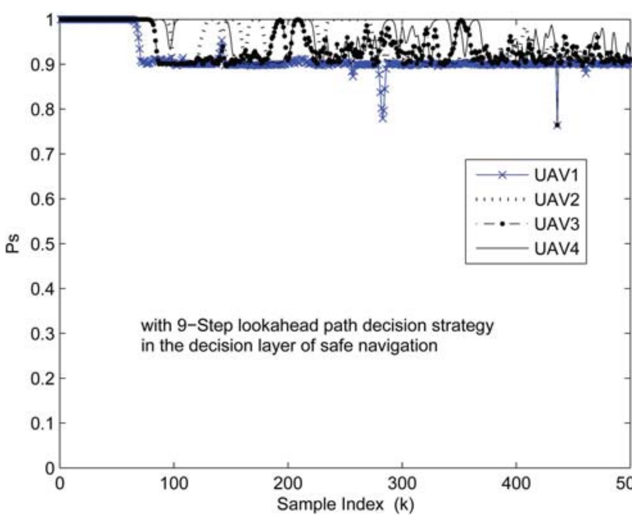


Fig. 10. Minimum survival probabilities of the UAVs (100 MC runs).

error of the targets. The initial zero RMS position errors indicate that the targets were not detected. Targets 1 and 3 were detected around time $k = 20$. Target 4 was detected around $k = 40$ and Target 2 was detected around $k = 70$ (There were some slight variations from run to run). It can be seen that target 1 is more accurately tracked due to the effort of UAV 1. The RMS position errors of the other targets satisfied the desired accuracy of the other UAVs (10 m as defined in (35)) soon after their detections, thus when the objectives with higher priorities (classification and tracking) have been accomplished, UAV 2–4's path decisions are optimized for search as long as the control decisions are "safe" for the UAVs.

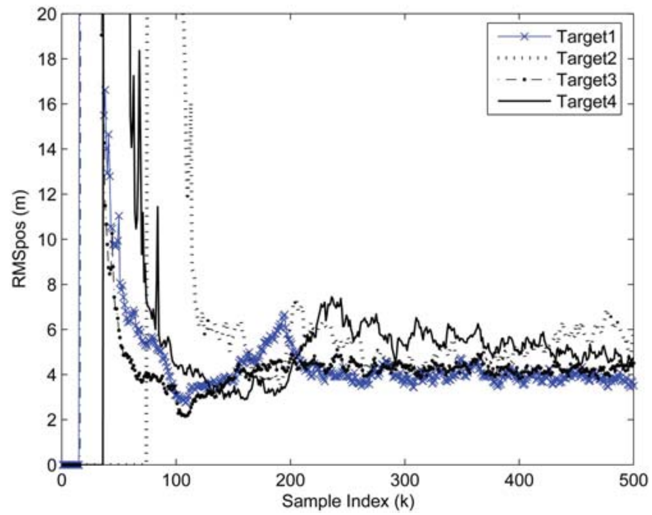


Fig. 11. RMS position error of the tracks (100 MC runs).

To summarize, the proposed path decision algorithm for UAV group is able, with moderate complexity, to i) guide a group of UAVs cooperatively for surveillance missions with multiple objectives, and ii) achieve balanced performance according to the various objective specifications.

6. CONCLUSIONS

For a surveillance mission by a group of UAVs with multiple objectives, generally the UAVs are guided by the gradient information from a certain "combination" of the objective functions. In this paper, the control of the UAV is discretized into a finite set, which amounts to sampling the objective functions over the continuous control space. Comparisons of the sample values are able to capture the gradient information in the objective functions, thus guiding the UAV group for the surveillance task.

More importantly, the discretization of control variables provides extra freedom in dealing with multiple objectives in the surveillance mission. Accordingly, a layered decision framework is proposed. Instead of using a single global objective function that is a weighted sum of all the objectives, different objectives are treated in separate decision layers in the order of their priorities. Compared to the weighted sum approach, the layered decision framework has the following advantages: i) multiple objectives in the surveillance mission are isolated; thus objectives with higher priorities are free from possible compromises from the less important ones; ii) for each objective, the specification of "satisfactory" levels allow the algorithm to be more sensitive to the entities (targets in tracking, sectors in search) that demand more attention; iii) the layered decision framework allows different path decision strategies to be used for the objectives, which makes the algorithm efficient.

The discretized controls also allow the extension of the time horizon of the path decisions, which is particularly important for the safe navigation of the UAVs. Accordingly, a multi-step look-ahead path decision strategy based on the Rollout policy is proposed. When used in the decision layer of safe navigation, this approach produces significantly improved results.

To keep the algorithm computationally feasible for large groups of UAVs, clustering of UAVs into small decision groups is discussed. Further flexibility of assigning different tasks to the UAVs is also incorporated into the path decision algorithm. Simulation results show that the proposed multi-step look-ahead path decision algorithm can effectively guide the UAV group for multi-objective surveillance missions and its performance is superior to the one-step look-ahead combined-objective approach.

REFERENCES

[1] Y. Bar-Shalom, X. R. Li and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
New York: Wiley, 2001.

[2] Y. Bar-Shalom, T. Kirubarajan and C. Gokberk
Tracking with classification-aided multiframe data association.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 3 (July 2005), 868–878.

[3] D. P. Bertsekas, J. N. Tsitsiklis and C. Wu
Rollout algorithms for combinatorial optimization.
Journal of Heuristics, **3**, 3 (Nov. 1997), 245–262.

[4] D. P. Bertsekas and D. A. Castanon
Rollout algorithms for stochastic scheduling problems.
Journal of Heuristics, **5**, 1 (Apr. 1999), 89–108.

[5] R. W. Beard, T. W. McLain, M. A. Goodrich and E. P. Anderson
Coordinated Target assignment and intercept for unmanned air vehicles.
IEEE Transactions on Robotics and Automation, **18**, 6 (Dec. 2002), 911–922.

[6] P. R. Chandler
UAV coperative control.
In *Proceedings of American Control Conference*, June 2001, 50–55.

[7] R. O. Duda, P. E. Hart and D. G. Stork
Pattern Classification (2nd ed.).
New York: Wiley, 2001.

[8] T. Furukawa, F. Bourgault, H. F. Durrant-Whyte and G. Dissanayake
Dynamic allocation and control of coordinated UAVs to engage multiple targets in a time-optimal manner.
In *Proceedings of IEEE International Conference on Robotics and Automation*, Apr. 2004, 2353–2358.

[9] T. Furukawa, F. Bourgault, B. Lavis and H. F. Durrant-Whyte
Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets.
In *Proceedings of Conference on Robotics and Automation*, May 2006, 2521–2526.

[10] G. M. Hoffmann, S. L. Waslander and C. J. Tomlin
Distributed cooperative search using information-theoretic costs for particle filters with quadrotor applications.
In *Proceedings of AIAA Guidance, Navigation and Control Conference*, Keystone, CO, Aug. 2006.

[11] V. P. Jilkov, X. R. Li and D. DelBalzo
Best combination of multiple objectives for UAV search & track path optimization.
In *Proceedings of the 10th International Confernce on Information Fusion*, July 2007.

[12] D. V. Kalbaugh
Optimal search among false contacts.
SIAM Journal of Applied Math, **52**, 6 (Dec. 1992), 1722–1750.

[13] S. M. Li, et al.
Automonous hierachical control of multiple unmanned combat vehicles.
In *Proceedings of American Control Conference*, May 2002, 274–279.

[14] T. W. McLain, P. R. Chandler and M. Pachter
A decomposition strategy for optimal coordination of unmanned air vehicles.
In *Proceedings of American Control Conference*, June 2000, 369–373.

[15] T. W. McLain, P. R. Chandler, S. Rasmussen and M. Pachter
Cooperative control of UAV rendezvous.
In *Proceedings of American Control Conference*, June 2001, 2309–2314.

[16] M. M. Polycarpou, Y. Yang and K. M. Passino
A cooperative search framework for distributed agents.
In *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, Sept. 2001.

[17] D. A. Schoenwald
UAVs: In space, air, water, and on the ground.
IEEE Control Systems Magazine, **20**, 6 (Dec. 2000), 15–18.

[18] A. Sinha, T. Kirubarajan and Y. Bar-Shalom
A distributed approach to autonomous surveillance by multiple cooperative UAVs.
In *Proceedings of SPIE Signal and Data Processing of Small Targets*, Oct. 2005, #5913-64.

[19] A. Sinha, T. Kirubarajan and Y. Bar-Shalom
Autonomous search, tracking and classification by multiple cooperative UAVs.
In *Proceedings of SPIE Conference on Signal Processing, Sensor Fusion, and Target Recognition*, Apr. 2006, #6235-09.

[20] L. D. Stone and J. A. Stanshine
Optimal search using uninterrupted contact investigation.
SIAM Journal on Applied Mathematics, **20**, 2 (Mar. 1971), 241–163.

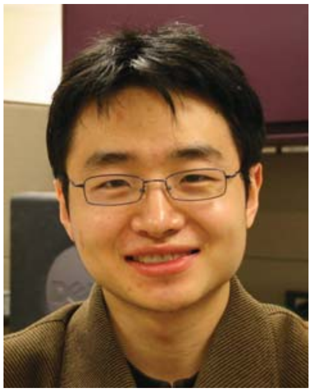
[21] F. Tu and K. R. Pattipati
Rollout strategy for sequential fault diagnosis.
IEEE Transactions of System, Man and Cybernetics, Part A, 33, 1 (Jan. 2003), 86–99.

[22] J. Yan, L. Yan, A. Minai and M. Polycarpou
Balancing search and target response in cooperative unmanned aerial vehichle (UAV) teams.
IEEE Transactions on Systems, Man and Cybernetics, Part B, **36**, 3 (June 2006), 571–587.

[23] Y. Yang, A. Minai and M. Polycarpou
Decentralized cooperative search by networked UAVs in an uncertain environment.
In *Proceedings of American Control Conference*, June 2004, 5558–5563.

[24] S. Yeom, T. Kirubarajan and Y. Bar-Shalom
Track segment association, fine-step IMM and initialization with Doppler for improved track performance.
IEEE Transactions on Aerospace and Electronic Systems, **40**, 2 (Jan. 2004), 293–309.

[25] U. Zengin and A. Dogan
Real-time target tracking for autonomous UAVs in adverarial environments: A gradient search algorithm.
IEEE Transactions on Robotics, **23**, 2 (Apr. 2007), 294–307.



Xin Tian was born in 1980. He received the B.E. degree in 2002 and M.E. degree in 2005, both from the Department of Information and Communication Engineering, Xi'an Jiaotong University, China. He is now a Ph.D. candidate in the Department of Electrical and Computer Engineering, University of Connecticut.

His research interests include statistical signal processing, tracking algorithms, and information fusion.

Yaakov Bar-Shalom (S'63—M'66—SM'80—F'84) was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab.

His current research interests are in estimation theory and target tracking and has published over 370 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate text *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000).

He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas.

During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–08.

In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. He is the recipient of the 2008 IEEE Dennis J. Picard Medal for Radar Technologies and Applications.

Krishna R. Pattipati (S'77—M'80—SM'91—F'95) received the B.Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut in 1977 and 1980, respectively.

From 1980–86 he was employed by ALPHATECH, Inc., Burlington, MA. Since 1986, he has been with the University of Connecticut, where he is a Professor of Electrical and Computer Engineering. His current research interests are in the areas of adaptive organizations for dynamic and uncertain environments, multi-user detection in wireless communications, signal processing and diagnosis techniques for complex system monitoring, and multi-object tracking. Dr. Pattipati has published over 370 articles, primarily in the application of systems theory and optimization (continuous and discrete) techniques to large-scale systems. He has served as a consultant to Alphatech, Inc., Aptima, Inc., and IBM Research and Development. He is a cofounder of Qualtech Systems, Inc., a small business specializing in intelligent diagnostic software tools.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He was elected a Fellow of the IEEE in 1995 for his contributions to *discrete-optimization algorithms for large-scale systems and team decision making*. Dr. Pattipati has served as the Editor-in-Chief of the *IEEE Transactions on SMC: Part B—Cybernetics* during 1998–2001, Vice-President for Technical Activities of the IEEE SMC Society (1998–1999), and as Vice-President for Conferences and Meetings of the IEEE SMC Society (2000–2001). He was corecipient of the Andrew P. Sage award for the Best SMC Transactions Paper for 1999, Barry Carlton award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," the 2003 AAUP Research Excellence Award and the 2005 School of Engineering Teaching Excellence Award at the University of Connecticut. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004 and 2005 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences.