2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Target Assignment for Integrated Search and Tracking by Active Robot Networks

Eric W. Frew, *IEEE Member* and Jack Elston

***Abstract*—This paper presents a new task assignment algorithm that integrates area search and target tracking. A new tracking metric is derived that encodes the ability of each robot to reach an unsensed target before the uncertainty in its position passes a given upper bound. Target activation and tracking assignment using this method enable broader participation by individual robots in a cooperative surveillance architecture. Area search is added to the cooperative surveillance architecture through the use of a coordinated coverage map that represents the probability of new targets being detected in regions of the environment. A key feature of the coverage map is that it is updated by all robots, not just robots performing area search. Thus, tracking robots aid the search process. Simulation results validate the new integrated assignment process.**

I. Introduction

Active sensing in robot networks takes advantage of robot mobility to optimize or improve information gathering activities. For some applications, such as persistent surveillance, proactive motion and cooperation between multiple vehicles can improve performance in terms of execution time, search efficiency, and system flexibility. For other applications, such as bearing-only tracking, active sensing is necessary to provide basic sensing observability. Future robotic sensor networks are expected to provide ubiquitous perception for a wide variety of applications including in-situ volumetric sensing [1], multi-target surveillance [2], and chemical plume tracking [3].

Cooperative search, acquisition, and tracking (CSAT) refers to the specific scenario where a robot sensor network actively searches or patrols a region of interest, identifies and assigns targets of interest for further tracking, and follows the assigned targets while performing coordinated geolocalization. Each of these three sub-tasks has been studied extensively in recent years, however, no architecture has been developed to integrate them into a unified framework.

Area search and *coverage control* refer to the task of traversing a region of interest with a sensor with a finite footprint in order to achieve a desired performance objective. In general, a coverage objective is defined that combines the percentage of area traversed by the sensor footprint with the quality of the sensing itself. When communication and range limitations are taken into account and the sensor is mobile, distributed control algorithms can be used to achieve optimal deployment [4]. When the target velocity is assumed bounded and sensing occurs within a finite limit, optimal strategies can be derived to guarantee capture under certain conditions [5]. Use of a recursive Baye's rule enables search over a grid such that the impact of any given visit of the sensor is correctly incorporated and plans can be made to maximize probability of detection [6].

Related to the problem of area search is *task assignment* and *resource allocation*. The area search problem can be decomposed by discretizing the environment and assigning different regions to different vehicles. The general task of assigning multiple robots to multiple spatially distributed locations is dominated by tight task coupling, limited information, and high degree of uncertainty. Fundamental difficulties arise due to the combinatorial nature of the assignment process and the complexity of optimal trajectory design for nonholonomic robots in cluttered environments. A variety of methods have been applied to cooperative task assignment including genetic algorithms, heuristic traveling salesman methods, and mixed integer linear programming.

The final component of a CSAT framework is cooperative tracking of moving ground targets. Here, the goal is to determine the position and velocity of a moving ground target through cooperative estimation and control. Nonlinearity of the estimation process adds a state dependence to estimator quality that can be exploited [7] to improve the process. Different approaches that explicitly optimize an information-theoretic measure of estimator performance [7] or use heuristics [8] to achieve a near-optimal relative configuration have been used to derive tracking controllers.

This paper presents a new formulation of a CSAT control architecture that combines all three components. In particular, elements of target tracking and coverage control are combined to specify desired levels of coverage and tracking (e.g. position estimate uncertainty) that must be maintained by the team. When new targets are detected in the environment, a naive approach would assign one or more vehicles to track the target until it leaves the region, in a "man-to-man" fashion. An intelligent adversary would simply send in more targets than vehicles until the area was no longer being patrolled. The approach presented here provides a "zone defense" where targets are tracked only until their state is adequately determined. The target is then free to move undetected until a second limit in position uncertainty is reached, after which it is reacquired. A new assignment metric is described that encodes the ability of each robot to reach a target before a prespecified limit is reached. This metric is used to activate targets for tracking and to assign robots to targets. Simulation results verify the new integrated search and tracking architecture.

Eric W. Frew is an Assistant Professor in the Aerospace Engineering Sciences Department at the University of Colorado, Boulder. eric.frew@colorado.edu
Jack Elston is a Graduate Student in the Aerospace Engineering Sciences Department at the University of Colorado, Boulder. jack.elston@colorado.edu

II. SYSTEM MODELS

A robot sensor network is defined as a set R of autonomous robots that are *mobile*, can *sense* information from the environment or targets, and are capable of wireless multi-hop *communication* with other robots in the network. Let $r_i \in R$ denote the i^{th} robot with state (position and velocity) $\mathbf{x}_i = [x_i, y_i, \dot{x}_i, \dot{y}_i]^T$ whose dynamics are governed by $\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i)$ where $\mathbf{u}_i$ is the control input and $\mathbf{w}_i$ is process noise. This work considers the nonholonomic kinematic model which is often used to describe the guidance layer behavior of many types of vehicles including mobile ground robots, underwater vehicles, and unmanned aircraft:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} u_{1,i}cos(\psi_i) \\ u_{1,i}sin(\psi_i) \\ u_{2,i} \end{bmatrix} \qquad \begin{matrix} v_{min} \leq u_{1,i} \leq v_{max} \\ \\ |u_{2,i}| \leq \omega_{max}. \end{matrix} \tag{1}$$

The target set T is defined to contain all targets t_j in the environment that could be sensed by the network R. Targets are described by the state vector $\mathbf{x}_j = [x_j, y_j, \dot{x}_j, \dot{y}_j]^T$ with dynamics $\dot{\mathbf{x}}_j = \mathbf{f}_j(\mathbf{x}_j, \mathbf{u}_j, \mathbf{w}_j)$. Target state estimates $\overline{\mathbf{x}}_j$ with covariance matrices P_j are maintained by some sensor fusion system.

For this work we consider the field of view or coverage area of the sensor and the relative position and/or velocity measurements that can be taken. Sensor coverage is modeled as the geometric area within which a sensor can receive a measurement from an object. We define the function $F_s(\mathbf{x}_i)$ as the sensor footprint, i.e. the area of the environment from which a sensor can receive a (position or velocity) measurement $\mathbf{z}_{ij}$ of some target t_j. The specific form of the function $F_s(\mathbf{x}_i)$ depends on the type of sensor being used

Typical sensors on mobile robots measure the range and bearing to target objects. These measurements come from a variety of sensors including monocular computer vision, sonar, radar, and time-of-arrival systems. Given a target located at position $\mathbf{p}_j$ the measurement obtained by a UA at position $\mathbf{p}_i$ is:

$$\mathbf{z}_{ij} = h(\mathbf{p}_i, \mathbf{p}_j, \mathbf{v}) = \left[r, \beta, \dot{r}, \dot{\beta}\right]^T + \mathbf{v} \tag{2}$$

where $r = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ is the range to the target, $\beta = \arctan\left(\frac{y_i - y_j}{x_i - x_j}\right)$ is the target bearing, and $\mathbf{v}$ is zero-mean Gaussian noise with covariance

$$E[\mathbf{v}\mathbf{v}^T] = \mathbf{R} = diag\left[\sigma_r^2, \sigma_\beta^2, \sigma_{\dot{r}}^2, \sigma_{\dot{\beta}}^2\right]. \tag{3}$$

Different sensors can be modeled by taking different values of the variance terms. For instance, bearings-only sensing occurs when $\sigma_r^{-2} = \sigma_{\dot{r}}^{-2} = \sigma_{\dot{\beta}}^{-2} = 0$.

The overall motivation of this work is development of a hierarchical control system to perform cooperative search, acquisition, and tracking. In particular, the motivating application is the use of a robot sensor network to patrol a region of interest in order to provide a given level of sensor coverage or to guarantee deterministic detection of target objects; to coordinate coverage and tracking data between robots in order to improve team performance; to assign robots to track new targets or to re-acquire old targets that have not been sensed over a finite amount of time; and to track and geo-locate ground targets to some specified level of performance. This work presented here focuses on a centralized target assignment process that achieves these goals.

III. ASSIGNMENT FOR INTEGRATED SEARCH AND TRACKING

The key feature of the algorithm described here is integration of search and tracking into a single task assignment framework. We assume the goal of the robot sensor network is to search a given region in order to detect moving targets and to track the targets in order to maintain a specified level of geo-localization error. Since robot sensing is limited, a tradeoff exists between the goals of area search and target localization. The steps presented in this section include 1.) determination of targets to track; 2.) assignment of active targets to available robots; and 3.) search by the remaining robots.

A. Tracking Metric

The target assignment problem presented here varies from traditional problems in several important ways. First, target tracking is not the only mode of behavior for the vehicles since they may also be in search mode. Second, targets are not removed from consideration once they are visited. The goal of the tracking system is to maintain the target position error within bounds so targets must be repeatedly visited. Finally, travel distance is not an appropriate assignment metric since we wish to maintain uncertainty below a specified limit.

In order to formulate the assignment problem a new performance metric is developed that incorporates the growth of the target position uncertainty when targets are not in view. Most geolocalization systems use recursive filters to maintain an estimate of the target state and its uncertainty in the form of a covariance matrix [9]. When not in view of any sensors the covariance matrix grows according to

$$\mathbf{P}_{k+1} = \mathbf{\Phi} \cdot \mathbf{P}_k \cdot \mathbf{\Phi}^T + \mathbf{Q} \tag{4}$$

where $\mathbf{P}_k$ is the covariance matrix at discrete time k, $\mathbf{\Phi}$ is the state transition matrix, and $\mathbf{Q}$ is the process noise covariance matrix. The square root of the spectral norm (the maximum singular value) of the position component of the estimate error covariance matrix is used as a measure of the uncertainty in the target's position, i.e. we define $\rho(\mathbf{P_p}) = \sqrt{\sigma_{max}(\mathbf{P_p})}$. The desired level of target uncertainty is specified by the limit ρ^* on the allowable magnitude of ρ, effectively limiting the radius of a circle that contains the target with some confidence defined by ρ^*. Given the current measure of target position uncertainty ρ_k, the time remaining before the desired uncertainty level is reached is

$$\Delta t_{\rho^*} = \frac{\Delta\rho}{\dot{\rho}_{ave}(\mathbf{P}_k)} = \frac{(\rho^* - \rho_k)}{\dot{\rho}_{ave}(\mathbf{P}_k)} \tag{5}$$

where

$$\dot{\rho}_{ave}(\mathbf{P}_k) = \frac{1}{\Delta t_{\rho^*}} \cdot \int_{\tau=k}^{\tau=k+\Delta t_{\rho^*}} \dot{\rho}(\tau) d\tau \tag{6}$$

is the average time rate of change of ρ over the time interval Δt_{ρ^*}.

The ability of robot r_i to reach target t_j before it violates the desired uncertainty bound can be determined by comparing Eq. 5 to the time $\Delta t_{i,j}$ needed by the robot to reach the target. As long as the constraint

$$\Delta t_{\rho^*} > \Delta t_{i,j} \tag{7}$$

is satisfied the target can be reached before the uncertainty limit is violated. This constraint can be turned into an assignment metric by considering the time remaining to perform other tasks, e.g. searching, before r_i must head toward the target

$$e_1(\mathbf{x}_{i,k}, \overline{\mathbf{p}}_t, \mathbf{P}_k) = \Delta t_{\rho^*} - \Delta t_{i,j} = \frac{(\rho^* - \rho_k)}{\dot{\rho}_{ave}(\mathbf{P}_k)} - \Delta t_{i,j}. \tag{8}$$

Equation 8 can be simplified by making assumptions on the robot motion. For example, assuming robot r_i moves at a constant speed in a straight line to the target, it would take $\Delta t_{i,j} = d_{i,j}/v_i$ to reach the target estimate where $d_{i,j} = \|\mathbf{p}_i - \overline{\mathbf{p}}_j\|$ and v_i is the robot speed. Substituting this time into Eq. 7 and rearranging terms gives a new constraint on the distance the robot can be from the target while still being able to reach it before the specified limit ρ^* is reached

$$d_{i,j} \leq \frac{v_i}{\dot{\rho}_{ave}(\mathbf{P}_k)} \cdot (\rho^* - \rho_k). \tag{9}$$

In general, the constraint on the distance a robot can be from a target and still reach the target in time is

$$d_{i,j} \leq \frac{v_{i,ave}}{\dot{\rho}_{ave}(\mathbf{P}_k)} \cdot (\rho^* - \rho_k) \tag{10}$$

where

$$v_{i,ave} = \frac{1}{\Delta t_{i,j}} \cdot \int_{\tau=k}^{\tau=k+\Delta t_{i,j}} \sqrt{\|\dot{\mathbf{x}}_\mathbf{j}(\tau) - \dot{\mathbf{x}}_\mathbf{i}(\tau)\|^2 d\tau} \tag{11}$$

is the average relative speed between the robot and the target over the time interval $\Delta t_{i,j}$. The constraint of Eq. 10 is turned into a performance metric by re-arranging terms

$$e_2(\mathbf{x}_{i,k}, \overline{\mathbf{p}}_t, \mathbf{P}_k) = \frac{v_{i,ave}}{\dot{\rho}_{ave}(\mathbf{P}_k)} \cdot (\rho^* - \rho_k) - d_{i,j}. \tag{12}$$

Interception of the j^{th} target by the i^{th} robot is then possible as long as $e_2 \geq 0$.

B. Approximate Tracking Metric

In some cases $v_{i,ave}$ and $\dot{\rho}_{ave}(\mathbf{P}_k)$ can be calculated analytically based on the robot guidance laws or target estimation process. Even when these terms cannot be determined analytically, they can be calculated by simulating the robot motion and target estimation processes forward in time. In practice this simulation can require significant computational resources that may not available. Thus, an approximation of Eq. 10 or Eq. 12 is helpful.

Let $\overline{v}_i \leq v_{i,ave}$ and $\overline{\dot{\rho}} \geq \dot{\rho}_{j,ave}(\mathbf{P}_{j,k})$ be under- and over-approximations, respectively, of the average speed of the robot and average time rate of change of ρ_j. Given these new variables it is easy to see that

$$\frac{\overline{v}_i}{\overline{\dot{\rho}}} \cdot (\rho^* - \rho_k) \leq \frac{v_{i,ave}}{\dot{\rho}_{ave}(\mathbf{P}_k)} \cdot (\rho^* - \rho_k). \tag{13}$$

Consider the new constraint

$$d_{i,j} \leq \frac{\overline{v}_i}{\overline{\dot{\rho}}} \cdot (\rho^* - \rho_k). \tag{14}$$

It is straightforward to see that Eq. 14 implies Eq. 10 and that imposing the new constraint gives a conservative bound on the distance the robot is allowed to move away from the target. Likewise, a new metric can now be derived from the new constraint

$$e_3(\mathbf{x}_{i,k}, \overline{\mathbf{p}}_t, \mathbf{P}_k) = \frac{\overline{v}_i}{\overline{\dot{\rho}}} \cdot (\rho^* - \rho_k) - d_{i,j}. \tag{15}$$

It is also straightforward to see that $e_3 \leq e_2$, i.e. the new metric is conservative. Note that for the nonholonomic kinematic model used here $v_i \geq v_{i,ave}$ since the robots cannot change direction instantaneously, thus setting $\overline{v}_i = v_i$ as in Eq. 9 is not a conservative approximation.

Approximating the average robot speed is difficult because of the nonholonomic kinematics of the motion model and the fact that the target may moving. If we let $v_{rel,ij}$ be the average speed of the optimal straight-line intercept course between r_i and t_j, we take $\overline{v}_{i,j} = \gamma \cdot v_{rel,ij}$ where $\gamma < 1$.

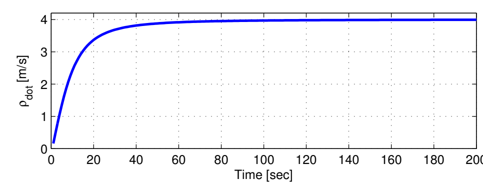

Fig. 1. Time rate of change $\dot{\rho}_k$ of uncertainty norm versus time for constant velocity target model with no process noise.

In order to derive the bound $\overline{\dot{\rho}}$, recall the recursive relationship (Eq. 4) for the covariance of the target state. Let the covariance matrix be written in block diagonal form

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{xx,k} & \mathbf{P}_{xv,k} \\ \mathbf{P}_{vx,k} & \mathbf{P}_{vv,k} \end{bmatrix} \tag{16}$$

where $\mathbf{P}_{xx,k}$ is the position component of the covariance matrix, $\mathbf{P}_{vv,k}$ is the velocity component, and $\mathbf{P}_{xv,k} = \mathbf{P}_{xv,k}^T$ contains cross-correlations. We are interested in the position component of the covariance and are using $\rho(\mathbf{P}_{xx,k})$ to define the desired target bound. For the case of a constant velocity target model with no process noise, e.g. $\mathbf{Q} = \mathbf{0}$, the position component of the covariance is updated only by the velocity and cross-correlation terms in Eq. 16 and it can be shown (see Fig. 1) that $\dot{\rho}(\mathbf{P}_{xx,k}) \to \rho(\mathbf{P}_{vv,k})$ as $k \to \infty$ so we can take $\overline{\dot{\rho}} = \rho(\mathbf{P}_{vv,k})$. When $\mathbf{Q} \neq \mathbf{0}$, $\dot{\rho}(\mathbf{P}_{xx,k})$ grows as $k \to \infty$ and a bound on $\dot{\rho}(\mathbf{P}_{xx,k})$ cannot be computed easily.

Figure 1 shows $\dot{\rho}(\mathbf{P}_{xx,k})$ versus time for a constant velocity target with no process noise. The initial covariance matrix is

$$\mathbf{P}_0 = diag\left[50^2, 20^2, 4^2, 2^2\right]. \tag{17}$$

The plot shows that $\dot{\rho}(\mathbf{P}_{xx,k}) \to \rho(\mathbf{P}_{vv,k}) = 4$ m/s.

C. Tracking Assignment

The next step in the assignment process is determining which targets need to be active. As long as $e_{3,ij} > 0$ for some robot at $\mathbf{p}_i$ the target t_j can be reached before it surpasses the specified uncertainty bound. Ideally the system only "activates" a target when $e_{3,ij} \leq 0$ for all $\mathbf{p}_i$. However, any delay would cause the target to violate the uncertainty limit and only one robot (the last for which $e_{3,ij} = 0$) would be capable of reaching the target in time. In order to add margin to the assignment process and insure that all robots could reach the active targets, we set a limit $\epsilon > 0$ and define the active target set

$$T_{active} = \{t_j | \min_i(e_{3,ij}) \leq \epsilon\} \tag{18}$$

Robot target assignment is performed using binary integer programming to minimize

$$f = \sum_{i=1}^{n} \sum_{t_j \in T_{active}} e_{i,j} \cdot b_{i,j} \tag{19}$$

subject to the constraint that every target is assigned to only one UA when there are more UA than targets, or every UA is assigned to only one target when there are more targets.

D. Search Assignment

In addition to assigning robots to targets for tracking, the task assignment process also coordinates area search by the robots. A non-empty subset $S \subseteq R$ of the robot sensor network will be designated as search robots only and will not be considered in the target assignment process. Additionally, robots not assigned to track a specific target will be included in the area search process.

Area search is coordinated through a discretized coverage map. A cellular decomposition is used to divide the world W into a finite set of n_c non-overlapping cells $A_c \subset W$. For this work we assume an open, i.e obstacle-free, environment decomposed into a simple grid. Each cell A_c is denoted by its center position $\mathbf{p}_c$ and represented by a single coverage variable a_c that represents the probability of the presence of a target in that cell. Let N_c (or $N(\mathbf{p}_c)$) be the neighbor set of cells adjacent to A_c. The coverage variable is updated by the rule

$$a_{c,k+1} = f(a_{c,k}, N_c, R). \tag{20}$$

When the footprint of a robot overlaps a cell, the coverage variable is decreased by a set amount. For most examples we consider ideal detection and thus reset the coverage variable to zero. An important aspect of the coverage map is that it is updated by *all* robot sensors, even ones in other task modes.

In order to simplify use of the coverage map for task assignment and to guarantee exploration of the entire world we require the following conditions:

C 1: $\mathbf{x}_i \in A_c \rightarrow A_c \subseteq F_s(\mathbf{p}_i)$

C 2: $f(a_{c,k}, N_c, R) = 0$ iff $\exists i$ such that $\mathbf{x}_{i,k} \in A_{c,k}$, otherwise $f(a_{c,k}, N_c, R) \geq a_{c,k} > 0$

C 3: if $a_{i,k} \leq a_{j,k}$ and neither cell is visited at time k, then $a_{i,k+1} \leq a_{j,k+1}$

The first condition insures that once a robot enters a cell it can sense the entire cell. This allows the coverage map to be updated by considering only the position of the robots without worrying about their geometric footprints. The second condition states that the coverage variable is reset to zero if and only if a cell is occupied (and from C1, sensed entirely) by a robot. Otherwise the coverage variable is always increased after each update. Combined with the second condition, the final condition insures that the coverage variable in one cell cannot increase fast enough to overtake the coverage variable of another cell with greater value. A result of this condition is that the cell with the maximum coverage variable will always have the maximum coverage variable until it is entered by a robot and the variable is reset to zero.

If there are n_s robots in search mode, area search is converted into an assignment process by identifying the n_s cells with the largest probability of containing the target. A standard binary integer program is then used to assign the robots to the cells based on straight line distances. This type of assignment process does not consider the area coverage that occurs enroute to the destinations. However, it encourages the robot sensor network to explore the entire space without getting stuck in locally flat regions typical of gradient-based approaches. As long as $n_c > 2n_s$ then there will always be n_s cells with non-zero (positive) coverage variables to serve as destinations for the robots. Conditions C1-C3 insure that the entire world is visited periodically. If $a_{m,k} = \max_{i=1,\cdots,n_c} a_{i,k}$ and $a_{m,k+1} = 0$, i.e. it is visited at time k, then C1-C3 imply that every other cell that was unoccupied at time k must be visited before $a_{m,k+T} = \max_{i=1,\cdots,n_c} a_{i,k+T}$ again. Since this applies to all cells for all k, this implies that every cell must eventually become the maximum cell or be visited again while a robot is enroute to another cell. In either case, every cell is repeatedly visited with a maximum time interval between visits determined by the size and shape of the world.

In order to consider moving targets, the coverage variable $a_{c,k+1}(\mathbf{p}_c)$ for the cell with center position $\mathbf{p_c}$ is recursively updated as a function of the previous value $a_{c,k}(\mathbf{p_c})$ and the values of the 8-connected neighbor $N_k^8(\mathbf{p}_c)$ variables

$$a_{c,k+1}(\mathbf{p_c}) = f(a_{c,k}(\mathbf{p_c}), N_k^8(\mathbf{p_c}), R). \tag{21}$$

In this work we fix the coverage variables of the perimeter to a value of one. This simulates the continued possibility that a target enters the world. The remaining cells in the grid are updated using a simple linear combination of the values of the 8-connected neighbors. For a grid decomposition represented by a 2-D matrix A_k where $a_{ij,k}$ is the coverage variable for the grid location (i, j) at time k, the update step is realized as convolution the matrix of A_k with the kernel

$$K = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{bmatrix}. \tag{22}$$

IV. SIMULATIONS

This section presents simulation results for 9 aerial robots searching an area that contains 4 moving targets. The aerial robots move with a nominal airspeed of $v_0 = 20$ m/s, have a maximum turn rate $\omega_{max} = 0.2$ rad/sec, and are allowed deviations of $\Delta v = 5$ m/s for coordination. The environment is 5000 meters wide and 8000 meters long. Initial positions of the 9 aircraft and 4 targets are randomly selected. Ground targets move with constant velocity with speed randomly chosen on the interval $[4, 10]$ and random initial headings. Targets change direction only upon reaching the boundary of the environment. Estimation of a target occurs once it it detected by moving into the sensor footprint of one of the robots.

For estimation, target motion is modeled as constant velocity with accelerations acting as process noise with variance $0.05 m/s^2$. In this work we assume each robot measures the bearing to the target with error covariance $\sigma_\theta = 5$ rad. A two stage approach is used for the measurement update step. First, measurements for the two robots are combined to form an instantaneous estimate of the target position. The measurement function is inverted (assuming some maximum range r_{max}) to obtain a local estimate of the target position $\hat{\mathbf{p}}_{ij,k} = h^{-1}(\mathbf{z}_{ij,k}, \mathbf{x}_{i,k})$ and approximate the error covariance matrix of that local estimate as

$$\mathbf{P}_{j,k} = (\mathbf{H}_{j,k}^T \cdot \sigma_\theta \cdot \mathbf{H}_{j,k})^{-1} \tag{23}$$

where $\mathbf{H}_{j,k}$ is the derivative of the measurement function with respect to the target position. If more than one robot senses the target their local estimates are combined to give the instantaneous position estimate

$$\hat{\mathbf{p}}_{j,k} = \hat{\mathbf{P}}_{j,k} \cdot \sum_{i=1}^{n} \mathbf{P}_{j,k}^{-1} \cdot \hat{\mathbf{p}}_{ij,k} \tag{24}$$

with new error covariance matrix

$$\hat{\mathbf{P}}_{j,k} = (\sum_{i=1}^{n} \mathbf{P}_{j,k}^{-1})^{-1} \tag{25}$$

where n is the number of robots sensing the target. Second, the full target state is estimated using $\hat{\mathbf{p}}_{j,k}$ as the measurement vector with noise covariance $\hat{\mathbf{P}}_{j,k}$ in a Kalman filter.

When multiple robots are assigned to track the same target a cooperative stand-off tracking algorithm is used to coordinate their motion. The coordination approach is based on a Lyapunov guidance vector field that produces essentially globally stable tracking to a closed loiter pattern [8]. For the work in this paper the loiter pattern is determined by the uncertainty ellipsoid associated with the target position covariance matrix.

Three different simulations are used to demonstrate the utility of the approach presented here. The first scenario uses the new algorithm presented in this paper. The task assignment algorithm attempts to keep the position error ρ_k between 20 and 50 meters. Each active target is assigned to two robots who track the target until $\rho \leq 20$m and all remaining robots are assigned to search. Since there are more than twice the number of robots than targets, at least one robot is always in the SEARCH mode. In order to use the approximate tracking metric of Eq. 14 the average robot and target error velocities are set to $\overline{v} = 0.8 v_0$ and $\dot{\overline{\rho}} = 2\sqrt{\rho(\mathbf{P}_v)}$, respectively. Figure 2

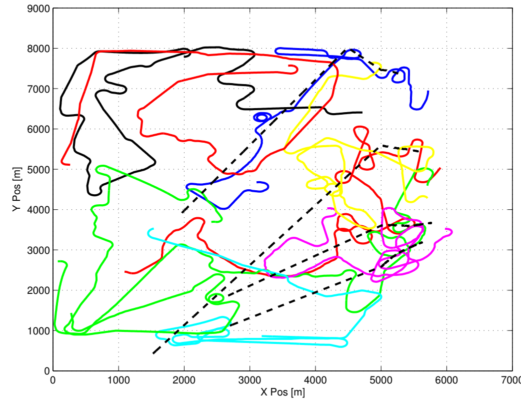

Fig. 2. Paths of 9 UA (solid lines) searching and tracking 4 moving targets (dashed lines) using the integrated assignment algorithm.

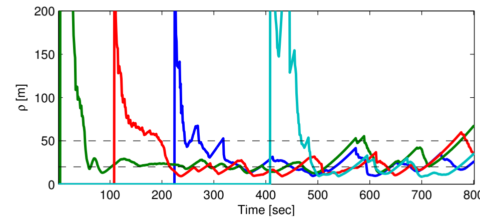

Fig. 3. Square root of the maximum singular value of the position error covariance matrix versus time.

shows the paths of the 9 robots and 4 targets over the course of an 800s simulation and Fig. 3 shows a plot of ρ for each target as a function of time. The dashed lines denote the upper and lower bounds of the desired position error.

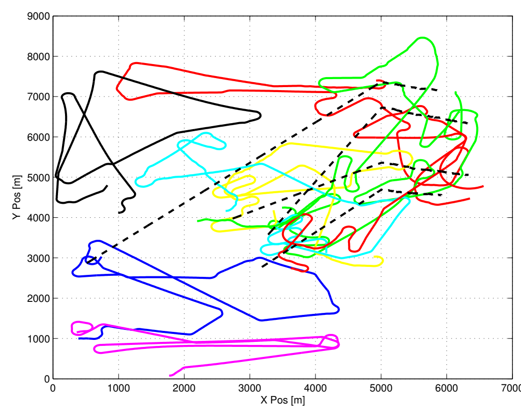

Fig. 4. Paths of 9 UA (solid lines) searching and tracking 4 moving targets (dashed lines) using the original assignment metric.

The second scenario uses a similar approach to the first. However, instead of using the new metric to activate targets, tracking is only initiated once the target uncertainty has passed above the upper bound (still 50 m). Figure 4 shows the paths of the robots and targets and Fig. 5 shows a plot of ρ for each target as a function of time.

The final scenario uses an algorithm that always assigns two robots to track any targets present in the environment. Once all targets are assigned, one robot is still left to perform area

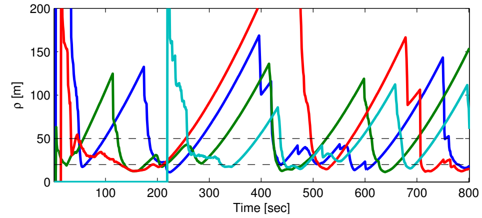

Fig. 5. Square root of the maximum singular value of the position error covariance matrix versus time.

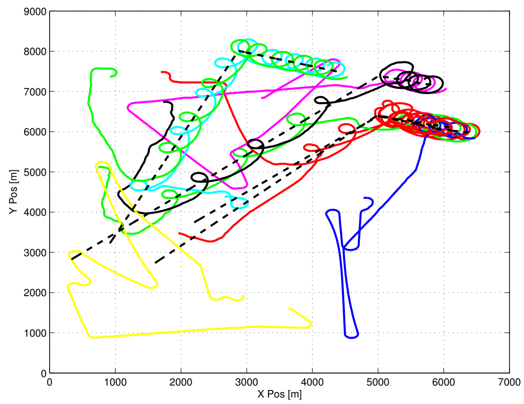

Fig. 6. Paths of 9 UA (solid lines) searching and tracking 4 moving targets (dashed lines) by continually orbiting around target once assigned.

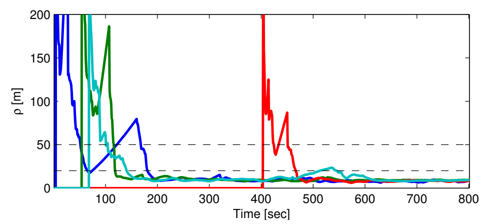

Fig. 7. Square root of the maximum singular value of the position error covariance matrix versus time.

search. While this approach clearly provides the best tracking performance, it comes at the expense of area coverage. In fact, this naive strategy is easily defeated by an intelligent adversary that introduces more targets than robots into the environment, drawing their attention away from the coverage task. Figures 6 and Fig. 7 show the paths and target uncertainty versus time, respectively.

The plots of the robot paths (Fig. 2 and Fig. 4) do not differ noticeably between the first two scenarios. In either case, the aerial robot network is able to cover, i.e. search over, a large percentage of the environment. In contrast, for the third case all but one of the robots maintain close proximity to their assigned targets and much of the environment is unexplored by the end of the simulation (Fig. 6). As expected, the third scenario provides the best tracking error (Fig. 7), keeping ρ at approximately 10 meters in steady state, well below the desired minimum. Likewise, the second approach also performs as expected; allowing the uncertainty to grow past the desired bound. This is expected since targets are not reactivated until they pass this mark. In many cases the target uncertainty grows up to three times the specified bound before it is finally sensed again (Fig. 5). The new approach described here is able to provide significant coverage of the environment while also keeping the target uncertainty below the specified bound most of the time (Fig. 3). Since the approximations $\overline{v}$ and $\dot{\overline{\rho}}$ were set heuristically, they are not always conservative and the target uncertainty occasionally grows above the given limit.

V. Conclusion

This paper presented a new task assignment algorithm that integrates area search and target tracking. A new tracking metric encodes the ability of each robot to reach an unsensed target before the uncertainty in its position passes a given upper bound. This allows broader participation by tracking robots in a cooperative surveillance architecture. Simulation results show that using the new tracking metric to activate and assign targets enables the robot sensor network to keep targets within desired uncertainty bounds while still exploring large portions of the environment.

Future work will consider a hierarchical decomposition of the robot sensor network. In particular, the assignment process will be extended for a heterogeneous aerial robot network consisting of mothership vehicles that carry smaller daughtership micro air vehicles. Hierarchical task assignment that combines distributed negotiation between mothership vehicles and centralized coordination between daughtership sub-teams is under development.

References

[1] B. Argrow, D. Lawrence, and E. Rasmussen, "Uav systems for sensor dispersal, telemetry, and visualization in hazardous environments," in *43rd Aerospace Sciences Meeting and Exhibit*, January 10-13 2005.

[2] A. Makarenko, A. Brooks, S. Williams, H. Durrant-Whyte, and B. Grocholsky, "A decentralized architecture for active sensor networks," in *IEEE International Conference on Robotics and Automation*, vol. 2, New Orleans, LA, USA, April 26 - May 1 2004, pp. 1097–102.

[3] M. A. Kovacina, D. Palmer, G. Yang, and R. Vaidyanathan, "Multi-agent control algorithms for chemical cloud detection and mapping using unmanned air vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, Lausanne, Switzerland, Sep 30 - Oct 4 2002, pp. 2782–2788.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[5] T. G. McGee and J. K. Hedrick, "Guaranteed strategies to search for mobile evaders in the plane," in *Proceedings of the American Control Conference*, Minneapolis, MN, 2006, pp. 2819 – 2824.

[6] L. Bertuccelli and J. How, "Bayesian forecasting in multi-vehicle search operations," in *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, 2006.

[7] M. L. Hernandez, "Optimal sensor trajectories in bearings-only tracking," in *Proceedings of the Seventh International Conference on Information Fusion*, vol. 2, Stockholm, Sweden, Jun 28 - Jul1 2004, pp. 893–900.

[8] E. W. Frew, "Cooperative stand-off tracking of uncertain moving targets using active robot networks," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, May 2007.

[9] M. E. Campbell and W. W. Whitacre, "Cooperative tracking using vision measurements on seascan uavs," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 6613–626, July 2007.