Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Search for Dynamic Targets with Uncertain Probability Maps

Luca F. Bertuccelli and Jonathan P. How

Aerospace Controls Laboratory
Massachusetts Institute of Technology
{lucab, jhow}@mit.edu

Abstract— **This paper extends a recently developed statistical framework for UAV search with uncertain probability maps to the case of dynamic targets. The probabilities used to encode the information about the environment are typically assumed to be exactly known in the search theory literature, but they are often the result of prior information that is both erroneous and delayed, and will likely be poorly known to mission designers. Our previous work developed a new framework that accounted for the uncertainty in the probability maps for stationary targets, and this paper extends the approach to more realistic dynamic environments. The dynamic case considers probabilistic target motion, creating Uncertain Probability Maps (UPMs) that take into account both poor knowledge of the probabilities and the propagation of their uncertainty through the environment. A key result of this paper is a new algorithm for implementing UPM's in real-time, and it is shown in various simulations that this algorithm leads to more cautious information updates that are less susceptible to false alarms. The paper also provides insights on the impact of the design parameters on the responsiveness of the new algorithm. Several numerical examples are presented to demonstrate the effectiveness of the new framework.**

I. Introduction

UAV search and attack missions have been extensively studied in the context of cooperative control [2], [3], [4], [10], [12], [13], and UAV search operations have been investigated in detail in Refs. [12], [15], [16], [18] (and the references therein). The environment is generally discretized in cells that are described by a probability of target existence. The vehicles store a *probability map*, which contains this probability for each cell and are typically assumed to be exactly known.

There are many examples of map building in the literature that are comparable to the methods presented here. Refs. [6], [7] consider the coordinated multi-vehicle search for a single non-evading target. Ref. [13] consider the case of dynamic undistinguishable obstacles, and update their map based on a probabilistic motion model and a measurement update step. Ref. [17] considers updating a map with a finite number of cells based on sensor observations. A key observation is that most of the work considered in this previous work assumes that the probabilities are known precisely. In reality, because these probabilities summarize the *information* at any given time of the state of each cell, and have been obtained from prior information that could be delayed or wrong (e.g., due to decoys and ghosting), they are not well known, and the overall map is thus uncertain. Assuming that these probabilities are well known could impact the probability map update, and ultimately adversely affect the search task itself.

Ref. [5] introduced a statistical framework based on Beta distributions that presented key analytical results for predicting the amount of time a vehicle should spend in a cell to increase the confidence of target existence in that cell. The probability associated with each cell was described by a Beta distribution that was updated based on whether a target was detected or not by the sensor. A Bernoulli distribution was used to model the image processing, which is an appropriate abstraction for target search applications because it returns a "Yes" (target detected) or "No" (no target detected) with specified success/failure rates based on correct detection and false alarm probabilities. This is in contrast to a sensor model described by a Gaussian sensor likelihood, which is more appropriate for tracking applications that involve an estimate of the target location and state.

The goal of this paper is to extend the approach in Ref. [5] to the case of dynamic targets, which requires the addition of a prediction step to account for target motion. This step requires taking the linear combination of Beta random variables, which is approximated by a new Beta distribution with appropriate modifications to the shape parameters. Note that other methods could be chosen to describe the uncertainty, such as using truncated Gaussian distribution or describing the uncertainty with "p-boxes" [9]. These distributions must be chosen wisely, however, since the measurement update following the prediction can be computationally intensive if the distributions are not conjugate. For example, using a truncated Gaussian PDF or "p-boxes" to describe the uncertain probabilities and passing them through Bayesian updates with a Bernoulli sensor model, would require additional computation: typically, one would need to sample the prior and numerically generate the posterior distribution in the update step. Furthermore, the posterior distribution may not have a closed form and thus would require additional computation to propagate it forward in the next measurement update via other sampling techniques.

By choosing to use the Beta distribution in the Uncertain Probability Map (UPM), the sampling is avoided altogether. In particular, while the proposed approach requires an approximation in the prediction step that is similar to the techniques discussed previously (except the pure Gaussian), the advantage of using the Beta distribution are underscored in the measurement step. Furthermore, the approximation in the prediction step does not add any significant com-

1-4244-0210-7/06/$20.00 ©2006 IEEE

putational load because it can be written as a closed-form update to the mean and variance of the Beta distributions associated with each cell. As a result, the prediction and measurement update stages can be expressed as a recursion on the means and variances of the distributions, which are very simple to implement and conceptually very similar to conventional filtering methods. This paper derives this recursion algebraically, analyzes the properties of the approach, and demonstrates it on full set of simulations.

This paper is organized as follows: Section II describes the notation used in this paper; the mathematical details of the propagation step are presented in Section III. For computational tractability, this step requires an approximation that is developed in Section IV. The new recursive form for the map updates and the response times of the UPMs are investigated Section V, and full scenario simulations are shown in Section VI.

II. Definitions

The following notation is used:

- $A \in \mathcal{R}^{MN,MN} = \alpha_{ij}$ denotes the transition probability of the target or the probability that the target will move to cell $i \in 1,\ldots,MN$ at time $k+1$ from cell $j \in 1,\ldots,MN$ at time step k. For a discretized $M \times N$ environment, the state transition matrix may not be fully populated since transitions between two non-adjacent cells may not be allowed to account for feasible target dynamics, and hence the transitions between these cells can be set to 0. In general, though, the state transition matrix is fully populated.
- The likelihood distribution is given by a series of observations that register target detection or target absence. It is described by a Bernoulli model that outputs $Y=1$ if a target is detected, and $Y=0$ if a target is not detected.

$$P(Y_{k+1}|p) \quad = \quad Mp^{\gamma_1-1}(1-p)^{\gamma_2-1}$$

where γ_1 (γ_2) denote the number of times a target is detected (and not detected, respectively), and $M = \frac{\Gamma(\gamma_1+\gamma_2)}{\Gamma(\gamma_1)\ \Gamma(\gamma_2)}$. This type of model is a simplified but reasonable method to approximate the image detection process. The intricacies of the image processing are abstracted away by treating the outputs ("yes" a target was detected, or "no" not detected) as binary measurements with an error probability that is representative of the image classification errors; this is analogous to a classical binary detection problem.

The general problem of updating the probability maps is presented in the next section.

III. General Problem

This section introduces the general information update steps in probability maps. The prediction and measurement update steps are first introduced for maps that assume precise knowledge of the probabilities, and then presented for the UPMs.

A. Prediction

When the environment is discretized into a finite number of cells, the prediction step can be described as follows

$$\underline{P}_{k+1|k} = A\ \underline{P}_{k|k}, \quad \underline{P}_{k|k} = \hat{\underline{P}}_{k|k} \tag{1}$$

where the stacked vector of probabilities is, $\underline{P}_{k+1|k} = [P_1(Y_{k+1|k})\ \ P_2(Y_{k+1|k})\ldots\ \ P_N(Y_{k+1|k})]^T$, and A represents the state transition matrix. $\hat{\underline{P}}_{k|k}$ denotes that the prior probabilities (or at least their point estimates) are known. Following directly from Eq. 1, each element $P_i(Y_{k+1|k})$ is evaluated as

$$\begin{aligned} P_i(Y_{k+1|k}) &= \sum_{j=1}^{N} \Pr(i_{k+1}|j_k) P_j(Y_{k|k}) \\ &= \sum_{j=1}^{N} \alpha_{ij} P_j(Y_{k|k}), \quad \sum_{i=1}^{N} \alpha_{ij} = 1 \end{aligned} \tag{2}$$

Note that because the total number of targets is unknown, the sum of the probabilities, $\sum_j P_j(Y_{k|k}) \neq 1$; in this case, the state transition matrix is used to diffuse the information in a step similar to Markov Chain theory.

The problem is more difficult when the probabilities are not known precisely however, and are assumed to lie in uncertainty sets: that is, $P_j(Y_{k|k}) \in \tilde{P}_j(Y_{k|k}) \subseteq \mathcal{P}$. The prediction step of Eq. 1 then becomes

$$\underline{P}_{k+1|k} = A\ \underline{P}_{k|k}, \quad \underline{P}_{k|k} \in \tilde{\underline{P}}_{k|k} \tag{3}$$

This uncertainty can be the realistic result of delayed measurements, or a mission planner's desire to incorporate adversarial intent into the problem. Various approaches exist for encoding this uncertainty, but as discussed in Ref. [5], few of these are conjugate through a Bayesian measurement update, and thus the posterior PDF must be computed numerically. In general this could be a computationally intensive approach, and not amenable to real-time implementation. The Beta distribution maintains computational tractability in the Bayesian measurement update by ensuring that the update is available in closed form. Note that, unlike the summation of i.i.d. Gaussian random variables (where the distribution of the sum is also normal), the Beta distribution is not in general guaranteed to be preserved in the sum of Beta distributed random variables. There are some relevant special cases, however, where this assumption is reasonable and effective. For example, with Beta distributions described by shape parameters greater than 5, the Beta distribution is Gaussian in shape and the approximation is very good. The approximation also works well for large sums of Beta distributed random variables (where a "Central Limit"-like theorem is invoked).

The Beta distribution is defined as

$$\tilde{P}(p) = \frac{\Gamma(b+c)}{\Gamma(b)\ \Gamma(c)} p^{b-1}(1-p)^{c-1}, \quad p \in [0,1] \tag{4}$$

Here, b and c are shape parameters that determine the overall shape of the distribution, allowing a designer to choose a sharply peaked distribution or larger variance distributions (the uniform distribution arises with $b = c = 1$). The Beta distribution was used in Ref. [5] to describe the uncertain probability of a target existing in a cell. Key advantages of using the Beta are its support is in the interval $[0, 1]$ and assuming a Bernoulli sensor model, the Beta is a conjugate distribution during measurement updates.

Using this model for the uncertainty description of the probabilities, Eq. 3 can be modified to be the linear combination of Beta distributed random variables, where the i^{th} entry will be given by

$$\tilde{P}_i(Y_{k+1|k}) = \sum_{j=1}^{N} \alpha_{ij} \tilde{P}_j(Y_{k|k}) \tag{5}$$

IV. Approximation

Since the $\tilde{P}_j(Y_{k|k})$ are uncertain, and assumed to be described by a Beta distribution, the problem becomes one of taking linear combinations of Beta distributed random variables. The resulting distribution is then assumed to be Beta in form to maintain computational tractability in the Bayesian update.

Consider the case when each variable, $P_j(p, k)$, has a weighting a_j. The new distribution is given by $P_T(p, k) = \sum_j P_j(p, k) a_j$, and is assumed to be a Beta distribution with mean $\overline{p}_j$ and standard deviation σ_j^2. The mean of the new (and assumed Beta) distribution is

$$\overline{P}_T = E\,[P_T] = \sum_j \overline{p}_j a_j \tag{6}$$

and the variance of the new distribution is given by

$$\sigma_P^2 = E(P_T - \overline{P}_T)^2 = \sum_j a_j^2 \sigma_j^2 \tag{7}$$

where the random variables are assumed to be independent. Note that the independence assumption is an approximation, and since the random variables are correlated, it will generally result in a slightly higher variance than if the correlation were explicitly included. From numerous numerical experiments that we ran however, the difference in variance between the correlated and (assumed) independent random variables resulted in similar distributions, therefore allowing us to maintain this assumption.

Now, since a Beta distribution

$$P_T(p, k) = \frac{\Gamma(b_{k+1|k} + c_{k+1|k})}{\Gamma(b_{k+1|k})\ \Gamma(c_{k+1|k})} p^{b_{k+1|k}-1}(1-p)^{c_{k+1|k}-1}$$

has principal moments

$$\text{Mean:} \qquad \overline{P}_T = \frac{b_{k+1|k}}{b_{k+1|k} + c_{k+1|k}}$$

$$\text{Var:} \qquad \sigma_P^2 = \frac{b_{k+1|k} c_{k+1|k}}{(b_{k+1|k} + c_{k+1|k})^2\ (b_{k+1|k} + c_{k+1|k} + 1)}$$

these equations can be inverted to solve for $b_{k+1|k}$ and $c_{k+1|k}$ (see appendix)

$$\begin{aligned} b_{k+1|k} &= \overline{P}_T \left[\frac{(1 - \overline{P}_T)\overline{P}_T}{\sigma_P^2} - 1 \right] \\ c_{k+1|k} &= \frac{1 - \overline{P}_T}{\overline{P}_T}\ b \end{aligned}$$

The Beta distribution that approximates the new distribution has parameters

$$\begin{aligned} b_{k+1|k} &= \sum_j \overline{p}_j a_j \left[\frac{(1 - \sum_j \overline{p}_j a_j) \sum_j \overline{p}_j a_j}{\sum_j \sigma_j^2 a_j^2} - 1 \right] \\ c_{k+1|k} &= \left(\frac{1 - \sum_j \overline{p}_j a_j}{\sum_j \overline{p}_j a_j} \right)\ b_k \end{aligned} \tag{8}$$

This new Beta distribution can now be used in the measurement update that follows the prediction step. Since the measurement update step preserves the Beta distribution, only two parameters need to be kept for each cell at each time step, and thus a conventional Bayesian measurement update can be performed.

Note that this approach uses the most information available to determine the approximation. Since the resultant Beta distribution is described by two new shape parameters, using the two moments to determine the new shape parameters leads to a unique solution.

A. Recursive Formulation

The propagation and measurement update steps for the UPM discussed in the earlier sections were based on the shape parameters (b and c) in each cell. While convenient to work with, a perhaps more convenient way to express the propagation and update steps is in a recursive form in terms of mean and variance of the distributions. The mean-variance recursive form for the updates of the UPM is presented next. The prediction step comes directly with a substitution of variables

$$\begin{aligned} \mu_{k|k}(j) &= \overline{p}_j\ , \quad \mu_{k+1|k} = \overline{P}_T \\ \sigma_{k|k}^2(j) &= \sigma_j^2\ , \quad \sigma_{k+1|k}^2 = \sigma_P^2 \end{aligned}$$

and the prediction form of the estimator (Eq. 7) is
Prediction

$$\begin{aligned} \mu_{k+1|k} &= \sum_j a_j\ \mu_{k|k}(j) \\ \sigma_{k+1|k}^2 &= \sum_j a_j^2\ \sigma_{k|k}^2(j) \end{aligned}$$

A recursive form for the measurement update step are shown next (the steps of the derivation are shown in the Appendix). The main difference when compared to the prediction step is that the measurement update recursion depends on whether a target was detected ($\delta_k = 1$) or not ($\delta_k = 0$).
Measurement Update

$$\mu_{k+1|k+1} = \begin{cases} \mu_{k+1|k} + \left(\dfrac{\sigma_{k+1|k}^2}{\mu_{k+1|k}} \right), & \delta_k = 1 \\[2ex] \mu_{k+1|k} - \left(\dfrac{\sigma_{k+1|k}^2}{1 - \mu_{k+1|k}} \right), & \delta_k = 0 \end{cases}$$

$$\sigma^2_{k+1|k+1} = \begin{cases} \dfrac{\mu_{k+1|k+1}\left(1-\mu_{k+1|k+1}\right)}{\dfrac{\mu_{k+1|k}(1-\mu_{k+1|k})}{\sigma^2_{k+1|k}}+1}, & \delta_k = 1 \\ \dfrac{\mu_{k+1|k+1}\left(1-\mu_{k+1|k+1}\right)}{\dfrac{\mu_{k+1|k}(1-\mu_{k+1|k})}{\sigma^2_{k+1|k}}+1}, & \delta_k = 0 \end{cases}$$

Note that the update step is nonlinear in the state estim $\mu_{k+1|k}$, and the covariance update is estimate-dependent (ι like a Kalman filter, where the covariance can be evalua off-line). Regardless, these equations represent a *recurs* approach to update the imprecise probability maps direc in terms of the mean and covariance of the distribution desired, the shape parameters of each distribution can backed out immediately by applying the results in Appen A. The key point for these recursions is that they prov a very intuitive way of updating the uncertainty in the probability maps. If a target is detected, the estimate $\mu_{k+1|k}$ will be shifted to the right towards 1 by a factor directly related to the variance, and inversely proportional to the estimate itself. Likewise, if the target is not detected, the estimate will be shifted toward 0. An added benefit of this proposed formulation is that the estimate of the probability will be guaranteed to lie between 0 and 1.

V. Map Update Analysis

This section analyzes the impact of the state transition A on the map update. Since this matrix drives the diffusion process based on an assumed model of the target motion, it is a key parameter that affects the overall performance of the map update. Typically, A is derived from knowledge of the motion of targets in the environment, or from implicit knowledge of the environment itself. Researchers frequently treat the A matrix as known, based on prior information of the target motion (see Ref. [11] and references therein for online identification of this matrix). The key point is that the choice of A drives the response time of the UPM update.

The impact of the state transition can be best understood with a numerical example. 9 cells (in a 3×3 array) and a total of 5 targets were considered in a scenario of 50 time steps. The original 5 targets were assumed to exist in cells 1, 2, 4, 7, and 9. They were stationary until time step 40, when two targets disappeared, and the remaining three targets occupied cells 3, 5, and 9. The sensor error distribution was assumed Bernoulli, with a probability of correct detection, 0.95, and a probability of false alarm of 0.15. The decision thresholds (confidence boundaries) that confirmed the presence or absence of a target were set at 0.75 and 0.25 respectively. Four choices of state transition matrices $A_i, i = 1, \ldots, 4$ were analyzed. The mean of the diagonal entries of A_1 (with $N = 9$) was

$$\bar{a} = \frac{1}{N}\sum_{j=1}^{N} a_{jj} = 0.35 \tag{9}$$

Hence, while there was a large weighting on the target staying in the original cell (this weighting is inversely related

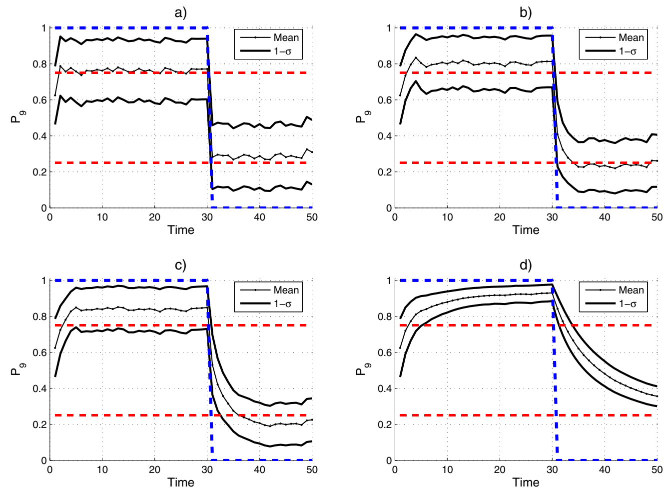

Fig. 1. Numerical simulations for various scale factors on the state transition matrix: a) Fully dynamic motion: scale factor = 1; b) Partially dynamic motion: scale factor = 5; c) Slower motion: scale factor = 20; d) Stationary: scale factor = 50. As the state transition matrix approaches the identity matrix, the response of the estimator is substantially slowed down.

to the mean sojourn time of the target in the cell), the off-diagonal entries were still significantly large enough to impact the diffusion process. The remaining matrices were generated by scaling up the diagonal entries of A_1 by a large factor, and re-normalizing the matrix (scale factors were 5, 20, and 50 respectively). This corresponded to an increase in the mean sojourn time of the target in each cell. 50 Monte Carlo simulations that created random realizations of the observations were performed. At each time step, a unique observation and propagation were made.

The simulation was done for all nine cells, but the time response for cell 4 is highlighted in Figure 1. This figure shows the responses for different choices on the scaling factors; scaling up the principal diagonal gives extra weighting to the diagonal elements, thereby making the state transition matrix an identity matrix (implying a stationary environment).

Figure 1(a) shows the response to the target leaving the cell at time 30 with an A_1 matrix that assumes a very dynamic environment, since the scale factor is unity. The time response is very fast, with a change detected immediately in one time step. Note that this fast response comes at a price: the steady-state limit of the estimate before time 30 is on the order of 0.79, just barely above the decision threshold for target detection of 0.75. Figure 1(d) shows the response when the targets were assumed virtually stationary. Note that at time 30, the response (from target “present” to target “absent”) takes longer than case (a); the steady-state estimate (at the times before 30 sec) was greater than for case (a), however, asymptotically reaching the limit of 0.95. Figures 1(c) and (d) show intermediate choices for the scale factors, and there is a clear trade-off between estimate response time and steady-state convergence of the estimate: the faster the response, the lower the steady-state estimate.

The key point of these simulations is that they demonstrate that the UPM updates have inherent time constants in responding to the new information obtained by the sensors. This is a critical observation for updating the UPMs: the update process of the map may be sluggish or fast depending on the assumed model of the environment. Each has its advantages and disadvantages. A responsive map update might also be more susceptible to sensing errors in the detection processes (thereby detecting more targets than actually exist), but will likely also require less time to confirm the presence or absence of a target. A more cautious response could be more immune to sensing errors, but may take longer time to confirm the existence of a target.

We have compared the UPM approach to conventional map update techniques that perform the Bayesian measurement update on point estimates of the probabilities (such as in Ref. [18]), by adding the propagation step of Eq. 2. The key observation was that the map updates were very responsive *regardless of the assumed dynamic model* A_i. The adverse affect of this behavior will be shown more explicitly in Section VI, where the map updates are very responsive to false alarms. Intuitively, an operator would likely prefer more cautious UPM update that are more immune to sensing errors, rather than very responsive updates that could erroneously identify targets. The implications for the ultimate search task is that vehicles could be sent to explore regions in space where no targets exist, thereby wasting valuable resources.

VI. Scenario

This new formulation of the UPM was embedded in the Receding Horizon Task Assignment (RHTA) framework [1]. RHTA is a computationally tractable algorithm that assigns vehicles to tasks while accounting for the distances between them; this models the realistic objective of visiting the *nearest* cells that are most likely to contain targets (while accounting for the uncertainty via the UPM).

The search strategy was applied to an environment (discretized in a 10×10 cell array with side lengths of 100 m) with 7 targets. The targets were assumed to transition probabilistically from cell to cell, with a mean sojourn time of approximately every 50-75 seconds in the 500-second mission. This time was chosen to be representative of the UAV physical capabilities; since the UAV took 3 seconds to visit each cell, a target transition every 50-75 seconds allowed the UAV to visit approximately 18 cells per target transition. The probability of correct detection was set at 0.90, while the probability of false alarm was set at 0.10. The key objective of the simulation was to compare the proposed UPM framework with the Bayesian Point Estimate (BPE) search approach that assumes point estimates for the probabilities. Both approaches applied the same dynamical model assuming perfect knowledge of the state transition matrix A. However, since the prior probabilities are imprecise, and the sensing mechanism is not perfect, the preference was to be more cautious in the UPM map updates. The initial conditions for the probabilities were randomized for each cell, $P_i(0|0) = \text{Beta}(b_i, c_i)$, and random realizations from this distribution were generated for each run. Note

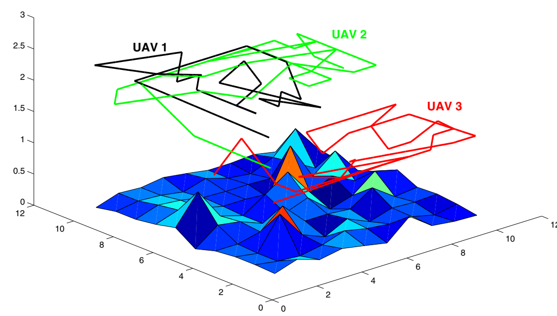

Fig. 2. Sample simulation demonstrating the 3 UAV trajectories overlayed on the probability map of the environment at time 499. Note that 6 of the 7 targets have been found in this particular simulation.

that this initial randomization created the imprecise prior probabilities, which are used for generating control decisions for the search. The total number of targets in the environment was not known a priori.

Two key parameters were tracked across 10 mission length simulations:

1) N_F – the total number of targets actually found;
2) N_{TF} – the total number of targets each vehicle *declared* were found.

A target was declared found if the probability, P_i in cell i satisfied $P_i > 0.8$. Hence, $N_F = (P_i > 0.8 \wedge$ target was in cell $i)$, while $N_{TF} = (P_i > 0.8)$. The results from these simulations are shown in Figure 3. For the number of targets actually found by the BPE and UPM approaches (top), note that the transient response of the first 50 seconds indicates that the UPM approach is slightly slower in finding the first 3–4 targets. The steady-state behavior for N_F is similar for both the BPE and UPM approach, however, and both methods find the same number of targets.

The bottom plot in Figure 3 displays the total targets *declared found*. Note that while the UPM approach (red) shows a close relation to the true number of targets found, the BPE approach on average declares significantly more targets found than were actually found. There is cause for concern here because, without access to the true location of the targets, the probability map is the only information available to unambiguously declare target *absence* or *presence* in a particular cell. Over predicting the number and location of targets would lead to inefficient strike and/or surveillance plans.

The reason why the BPE approach is overly optimistic is due to the responsiveness of the updates, while the UPM approach is more cautious in updating the probabilities, effectively requiring more looks per cell before declaring a target "present" or "absent". The fact that the UPM updates more cautiously does not impact the total number of targets found, however, since the UPM approach found the same number of targets as the BPE approach at steady-state.

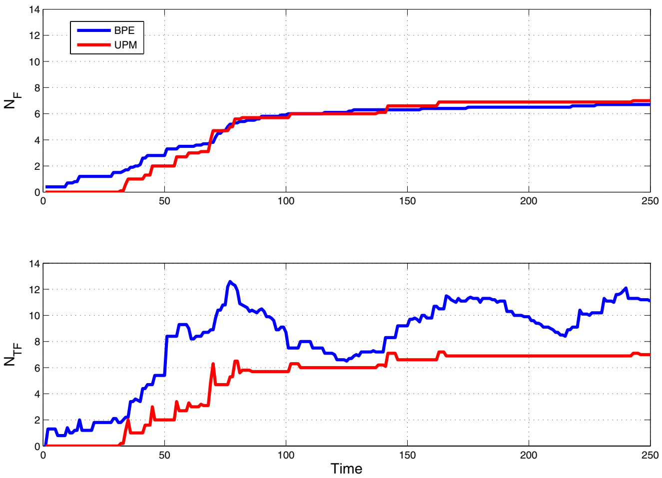

Fig. 3. Top: Total number of targets truly found; Bottom: Total number of targets thought to be found by the vehicles. Note that using the Beta formulation results in a more realistic map, while the Bayes update scheme results in an overly optimistic map.

Hence, the inclusion of the uncertainty, while adding some latency to the responsiveness of the map updates, did not affect the overall objective of finding the targets in the environment.

VII. Conclusion

This paper extended a previous formulation for search problems in uncertain environments to the case of dynamic targets. The proposed approach uses Beta distributions to describe the uncertainty in the probability of target existence in the cells. The new form of the algorithm in this paper updates these Beta distributions using a recursive method that includes both propagation and measurement update steps. By developing an approximation for the combination of the Beta random variables encountered in the propagation phase in terms of updates in the mean and variance, the recursive algorithm is very similar to a classical recursive filter. Comparisons of the overall algorithm with current approaches that use point estimates for these probabilities showed that the new approach generates more consistent maps which will ultimately result in more efficient search and strike tasks. Future research will investigate how to handle inconsistent maps among UAVs.

Acknowledgments

The first author was supported by a National Defense, Science, and Engineering Graduate Fellowship. The research was funded in part by AFOSR Grant # FA9550-04-1-0458.

Appendix A

Since $\overline{P}_T = \frac{b_k}{b_k + c_k}$, then

$$c_k = \frac{1 - \overline{P}_T}{\overline{P}_T}\, b_k \qquad (10)$$

Then, $b_k + c_k$ is:

$$b_k + c_k = b_k \left(1 + \frac{1 - \overline{P}_T}{\overline{P}_T}\right) = \frac{b_k}{\overline{P}_T} \qquad (11)$$

and

$$b_k + c_k + 1 = b_k \left(1 + \frac{1 - \overline{P}_T}{\overline{P}_T}\right) + 1 = \frac{b_k + \overline{P}_T}{\overline{P}_T} \qquad (12)$$

The variance, $\sigma_P^2 = \frac{b_k c_k}{(b_k+c_k)^2(b_k+c_k+1)}$ is then given by:

$$\sigma^2 = \frac{\overline{P}_T^2 (1 - \overline{P}_T)}{b_k + \overline{P}_T} \qquad (13)$$

and by solving for b_k

$$b_k = \overline{P}_T \left[\frac{(1 - \overline{P}_T)\overline{P}_T}{\sigma^2} - 1 \right] \qquad (14)$$

References

[1] Alighanbari, M. *Task Assignment Algorithms for Teams of UAVs in Dynamic Environments*, Massachusetts Institute of Technology, M. S. Thesis, 2004.

[2] Alighanbari, M., L. F. Bertuccelli, and J. P. How. *Filter-Embedded UAV Task Assignment Algorithms For Dynamic Environments* AIAA Guidance, Navigation and Control Conference, Aug 2004.

[3] Bellingham, J. S., M. Tillerson, M. Alighanbari and J. P. How, *Co-operative Path Planning for Multiple UAVs in Dynamic and Uncertain Environments,* Proceeding of the IEEE Conference on Decision and Control, Dec. 2002.

[4] Bertuccelli, L. F. , M. Alighabari, and J. P. How. *Robust Planning For Coupled Cooperative UAV Missions,* IEEE Conference on Decision and Control, 2004.

[5] Bertuccelli, L. F. and J. P. How. *Robust UAV Search for Environments with Imprecise Probability Maps*, to appear at the IEEE Conference on Decision and Control, 2005.

[6] Bourgault, F., T. Furukawa, and H. F. Durrant-Whyte. *Coordinated Decentralized Search for a Lost Target In a Bayesian World,* IEEE/RSJ International Conf. on Intelligent Robots and Systems, October 2003.

[7] Bourgault, F., T. Furukawa, and H. F. Durrant-Whyte. *Process Model, Constraints, and the Coordinated Search Strategy,* IEEE International Conf. on Robotics and Automation, April 2004.

[8] Hahnel, D., R. Triebel, W. Burgard, and S. Thrun. *Map Building with Mobile Robots in Dynamic Environments,* IEEE ICRA, 2003.

[9] Ferson, S., V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz. *Constructing Probability Boxes and Dempster-Shafer Structures,* Sandia National Laboratories. SAND2002-4015, 2002.

[10] Flint, M., E. Fernandez-Gaucherand, and M. Polycarpou. *Cooperative Control for UAVs Searching Risky Environments for Targets,* IEEE Conference and Decision and Control, 2003.

[11] Jilkov, V. and X. R. Li. *Online Bayesian Estimation of Transition Probabilities for Markovian Jump Systems*, IEEE Trans. of Signal Processing, Vol. 52, No. 6, June 2004.

[12] Jin, Y., A. Minai, and M. Polycarpou. *Cooperative Real-Time Search and Task Allocation in UAV Teams*, IEEE Conference on Decision and Control, 2003.

[13] Jun, M. and R. D'Andrea. *Probability Map Building of Uncertain Dynamic Environments with Indistinguishable Obstacles,* IEEE American Control Conference, 2003.

[14] Pollock, S. M. *A Simple Model of Search for a Moving Target,* Operations Research, Vol. 18(5), Pg. 883-903, 1970.

[15] Polycarpou, M., Y. Yang, and K. Passino. *A Cooperative Search Framework for Distributed Agents*, IEEE International Symposium on Intelligent Control, 2001.

[16] Stone, L. *Theory of Optimal Search*, 2^{nd}*ed.*, Military Applications Society, 2004.

[17] Thrun, S. *Learning Occupancy Grid Maps with Forward Sensor Models,* "Autonomous Robots", Vol. 15. Kluwer, 2003.

[18] Yang, Y., A. Minai, and M. Polycarpou. *Evidential Map-Building Approaches for Multi-UAV Cooperative Search*, American Control Conference, 2005.