Efficient Methods of Non-myopic Sensor Management for Multitarget Tracking

Chris Kreucher[1], Alfred O. Hero III[2], Keith Kastella[1], and Daniel Chang[1]

[1] General Dynamics Advanced Information Systems, Ann Arbor, MI
{Christopher.Kreucher, Keith.Kastella, Daniel.Chang}@gd-ais.com
[2] University of Michigan Department of EECS, Ann Arbor, MI
hero@eecs.umich.edu

Abstract. This paper develops two efficient methods of non-myopic (long-term) sensor management and investigates the benefit in the setting of multitarget tracking. The underlying tracking methodology is based on recursive estimation of a Joint Multitarget Probability Density (JMPD), which is implemented using particle filtering methods. The myopic sensor management scheme is predicated on maximizing the *expected* Rényi Information Divergence between the current JMPD and the JMPD after a measurement has been made. A full non-myopic strategy based on this information theoretic method is calculated using Monte Carlo methods for a model problem. Since this is computationally intractable when looking more than a small number of time steps ahead, two alternative strategies are investigated. First, we develop an information-directed search algorithm which focusses the Monte Carlo evaluations on action sequences that are most informative. Second, we give two approximate methods which replace the value-to-go with an easily computed function which captures the long term value of the current action. The performance of these methods is compared to the myopic scheme in terms of tracking performance and computational requirements.

1 Introduction

The problem of sensor management is to determine the best way to task a sensor or group of sensors when each sensor may have many modes and search patterns. Typically, the sensors are used to gain information about the kinematic state (e.g. position and velocity) and identification of a group of targets. Applications of sensor management are often military in nature [1], but also include things such as wireless networking [2] and robot path planning [17]. There are many

[3] This material is based upon work supported by the United States Air Force under Contract No. F33615-02-C-1199, AFRL contract SPO900-96-D-0080, and by ARO-DARPA MURI Grant DAAD19-02-1-0262. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

objectives that the sensor manager may be tuned to meet, e.g. minimization of track loss, probability of target detection, minimization of track error/covariance, and identification accuracy. Each of these different objectives taken alone may lead to a different sensor allocation strategy [1].

Sensor management schemes may be either myopic (i.e. short term) or non-myopic (i.e. long-term). Many researchers have investigated myopic schemes using, for example, an information-based sensor selection strategy [16][3][4]. It is believed that long-term sensor scheduling will out perform short-term methods in situations where the dynamics of the scenario are rapidly and predictably changing. For example, when targets and/or sensor platforms are moving the visibility of a target from a sensor changes with time and this property may make long-term planning advantageous.

The long-term sensor scheduling problem has been approached with a Markov decision process strategy. However, a complete long-term scheduling solution suffers from combinatorial explosion when solving practical problems of even moderate size. Researchers have thus worked at approximate solution techniques. For Example, Krishnamurthy [5][6] uses a multi-arm bandit formulation involving hidden Markov models. Since the optimal approach has prohibitive computational complexity, suboptimal approximate methods are developed and some simple numerical examples involving a small number of targets moving among a small number of discrete states are presented. Even with the proposed suboptimal solutions, the problem is still very challenging numerically. Bertsekas and Castanon [7] formulate heuristics for the solution of a stochastic scheduling problem corresponding to sensor scheduling. They implement a rollout algorithm based on their heuristics to approximate the stochastic dynamic programming algorithm. Castanon [8][9] formulates the problem of classifying a large number of stationary objects with a multi-mode sensor based on a combination of stochastic dynamic programming and optimization techniques.

In this paper, we detail a multi-target tracking situation in which non-myopic scheduling should out perform myopic scheduling. This scenario involves a moving sensor which, due to terrain elevation, results in part of the surveillance region being not visible at each time step. We contrast the sensor scheduling decisions made by a myopic scheduler with that of a non-myopic scheduler in terms of the resulting track error. As the full non-myopic solution requires computational time exponential in the number of time steps forward that the algorithm plans, we present two alternative schemes. First, we give an information-directed path searching scheme which reduces the complexity of the full Monte Carlo (MC) search and yields similar results. Second, we present approximate methods which replace the value-to-go by a function of visibility which captures the long-term value benefit of an action. This function is clearly motivated in the case of intervisibility constraints, but may not be motivated in general.

This paper proceeds as follows. First, Section 2 is an overview of Bayesian multiple target tracking and our particle filter implementation. Second, in Section 3, we give the details of our information-based method of myopic sensor management. Third, in Section 4, we provide a motivating example of a scenario

in which non-myopic sensor management ought to provide benefit. Fourth, in Section 5, we detail the full MC approach to non-myopic sensor management, and note the intractability for long time-scale problems. Additionally, we show an information-directed method of selectively searching trajectories, which results in similar performance as the full MC method at reduced computational cost. Furthermore, we detail techniques that approximate the non-myopic strategy with a computational cost similar to that of the myopic strategy by replacing the value-to-go with a function of future visibility that captures the long-term value of an action. Fifth, in Section 6, we provide simulation results comparing the myopic, non-myopic, and approximate techniques in terms of track error and computational burden.

2 Bayesian Multi-target Tracking

Here we briefly describe recursive Bayesian multiple target tracking and our particle filter implementation. A more detailed presentation of our methodology may be found in [11]. The central element of our target tracking methodology is the Joint Multitarget Probability Density (JMPD).

Estimating the JMPD provides a means for tracking an unknown number of targets in a Bayesian setting. The statistics model uses the joint multitarget conditional probability density $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k|\mathbf{Z}^k)$ as the probability density for exactly T targets with state vectors s $\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ at time k based on a set of observations $\mathbf{Z}^k$. The number of targets T is a variable to be estimated simultaneously with the states. The observation set $\mathbf{Z}^k$ refers to the collection of measurements up to and including time k, i.e. $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, ...\mathbf{z}^k\}$, where each of the $\mathbf{z}^i$ may be a single measurement or a vector of measurements made at time i. As is typically done, we denote the multitarget state vector by $\mathbf{X}$, i.e. $\mathbf{X} = [\mathbf{x}_1,\ \mathbf{x}_2,\ ...,\ \mathbf{x}_{T-1},\ \mathbf{x}_T]$, where $\mathbf{X}$ is defined for all T, $T = 1...\infty$.

For simplicity, the simulations presented in this paper treat only the case where the number of targets is known and fixed, and the states of the targets are one-dimensional. Details regarding multitarget tracking and myopic sensor management when the state of each target is four dimensional and many targets are tracked, may be found in [11] and [3].

The temporal update of the posterior likelihood of the JMPD proceeds according to the usual rules of Bayesian filtering. Given a model of how the JMPD evolves over time $p(\mathbf{X}^k|\mathbf{X}^{k-1})$, we compute the prediction density via

$$p(\mathbf{X}^k|\mathbf{Z}^{k-1}) = \int d\mathbf{X}^{k-1} p(\mathbf{X}^k|\mathbf{X}^{k-1}) p(\mathbf{X}^{k-1}|\mathbf{Z}^{k-1}) \tag{1}$$

$p(\mathbf{X}^k|\mathbf{Z}^{k-1})$ is referred to as the prior or prediction density at time k, as it is the density at time k conditioned on measurements up to and including time $k-1$. The time evolution of the JMPD may be a collection of target kinematic models or may involve target birth and death.

Given a model of the sensor, $p(\mathbf{z}^k|\mathbf{X}^k)$, and assuming conditional independence of the measurements given the state, Bayes' rule is used to update the posterior density as new measurements $\mathbf{z}^k$ arrive via

$$p(\mathbf{X}^k|\mathbf{Z}^k) = \frac{p(\mathbf{z}^k|\mathbf{X}^k)p(\mathbf{X}^k|\mathbf{Z}^{k-1})}{p(\mathbf{z}^k|\mathbf{Z}^{k-1})} \tag{2}$$

$p(\mathbf{X}^k|\mathbf{Z}^k)$ is referred to as the posterior or updated density at time k as it is the density at time k conditioned on all measurements up to and including time k.

The sample space of $\mathbf{X}^k$ is very large. It contains all possible configurations of state vectors $\mathbf{x}_i$ for all possible values of T. Discretization of the JMPD on a grid has computational burden exponential in the number of targets and grid cells allotted to each state. We find that a particle filter based implementation allows for computational tractability. To implement JMPD via a particle filter (PF), we approximate the joint multitarget probability density $p(\mathbf{X}|\mathbf{Z})$ by a set of N_{part} weighted samples (particles), $p(\mathbf{X}|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p)$.

Particle filtering is a method of approximately solving the prediction and update equations by simulation. The particle filter recursion implemented here follows the standard paradigm of particle proposal, weight update and resampling [15], but utilizes an adaptive sampling strategy that dramatically reduces the number of particles required to track multiple targets (see [11] for details).

3 Information Based Myopic Sensor Management

In this section, we detail our information-based myopic sensor management algorithm. The setting in which we are interested is where a collection of moving targets is to be tracked using a fixed number of sensor dwells at each time step.

As others have realized [16][10], a good measure of the quality of each sensing action is the reduction in entropy of the posterior distribution that is expected to be induced by the measurement. Therefore, at each instance when a sensor is available, we use an information-based method to compute the best sensing action to take. This is done by first enumerating all possible sensing actions. A sensing action may consist of choosing a particular mode (e.g. SAR mode or GMTI mode), a particular dwell point/pointing angle, or a combination of the two. Next, the *expected* information gain is calculated for each possible action, and the action that yields the maximum expected information gain is taken.

The calculation of information gain between two densities f_1 and f_0 is done using the Rényi information divergence [12][14], also known as the α-divergence:

$$D_\alpha(f_1||f_0) = \frac{1}{\alpha - 1} ln \int f_1^\alpha(x) f_0^{1-\alpha}(x) dx \tag{3}$$

The α parameter in eq. (3) is used to adjust how heavily one emphasizes the tails of the distributions f_1 and f_0. In the limit as $\alpha \to 1$, the α-divergence converges to the Kullback-Leibler Divergence. In our application, we are interested in computing the divergence between the predicted density $p(\mathbf{X}^k|\mathbf{Z}^{k-1})$

and the updated density after a measurement is made, $p(\mathbf{X}^k|\mathbf{Z}^k)$. Our particle filter approximation of the density simplifies eq. (3) to

$$D_\alpha\left(p(\mathbf{X}^k|\mathbf{Z}^k)||p(\mathbf{X}^k|\mathbf{Z}^{k-1})\right) = \frac{1}{\alpha-1} ln \frac{1}{p(\mathbf{z})^\alpha} \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X_p})^\alpha \tag{4}$$

where

$$p(\mathbf{z}) = \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X}_p) \tag{5}$$

The sensor model $p(\mathbf{z}|\mathbf{X}_p)$ is used to incorporate everything known about the sensor, including signal to noise ratio, detection probabilities, and whether the locations represented by $\mathbf{X}_p$ are visible to the sensor.

We want to perform the measurement that makes the divergence between the current density and the density after a new measurement largest. This indicates that the action has maximally increased the information content of the measurement updated density with respect to the density before a measurement was made. To this end, we calculate the expected value of eq. (4) for each of the N possible sensing actions and choose the action that maximizes the expectation. We use a_i, $i = 1...N$ to refer to the possible sensing actions under consideration, including but not limited to sensor mode selection and sensor beam positioning.

The expected value of eq. (4) may be written as an integral over all possible outcomes z_{a_i} when performing sensing action a_i:

$$< D_\alpha >_{a_i} = \int d\mathbf{z}_{a_i} p(\mathbf{z}_{a_i}|\mathbf{Z}^{k-1}) D_\alpha\left(p(\mathbf{X}^k|\mathbf{Z}^k)||p(\mathbf{X}^k|\mathbf{Z}^{k-1})\right) \tag{6}$$

In the special case where measurements are thresholded and are therefore either detections or no-detections this integral reduces to

$$< D_\alpha >_{a_i} = \frac{1}{\alpha-1} \sum_{z_{a_i}=0}^{1} p(z_{a_i}) ln \frac{1}{p(z_{a_i})^\alpha} \sum_{p=1}^{N_{part}} w_p p(z_{a_i}|\mathbf{X}_p)^\alpha \tag{7}$$

In summary, our method of myopic sensor management proceeds as illustrated in Figure 1. At each occasion where a sensing action is to be made, we evaluate the expected information gain as given by eq. (7) for each possible sensing action a_i. We then perform the sensing action that gives maximal expected information gain. Computationally, the value of eq. (7) can be calculated for N possible sensing actions in $O(NN_{part})$.

4 Non-Myopic Sensor Management : Motivating Example

It is expected that in some situations a non-myopic sensor management strategy will provide sensor tasking decisions that are better than the myopic strategy. In this section, we detail such a scenario. We consider the problem where at each

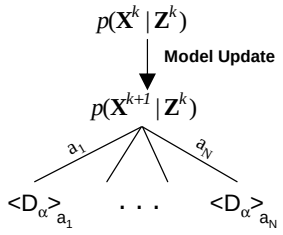


Fig. 1. Myopic Sensor Management

time step an airborne sensor is able to image a portion of a ground surveillance area in order to determine the location of a set of moving ground targets. The surveillance area has a known terrain elevation, and the sensor has a known flight pattern (Figure 2).

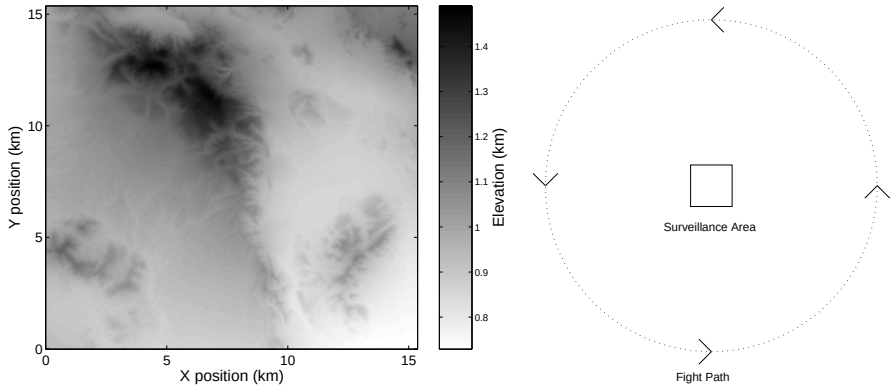


Fig. 2. Left: The elevation of the surveillance area. Right: The sensor flight path.

At each time step, the sensor position relative to the surveillance area causes certain portions of the ground to be unobservable due to terrain elevation between the sensor and the ground. Given the sensor position and the terrain elevation, at any time we can compute a visibility mask which determines whether a particular spot on the ground can be seen by the sensor. As an example, in Figure 3, we give the visibility maps that are computed from a sensor positioned below and to the right of the surveillance area.

The visibility constraint enters directly into our sensor management formulation given in Section 3, through the $p(\mathbf{z})$. This factor will make the (myopic) sensor manager give no gain to looking at an invisible cell, and hence no dwells will be made in those cells.

The situation in which we expect non-myopic sensor management to aid in target tracking is when a target becomes invisible to the sensor for a brief amount

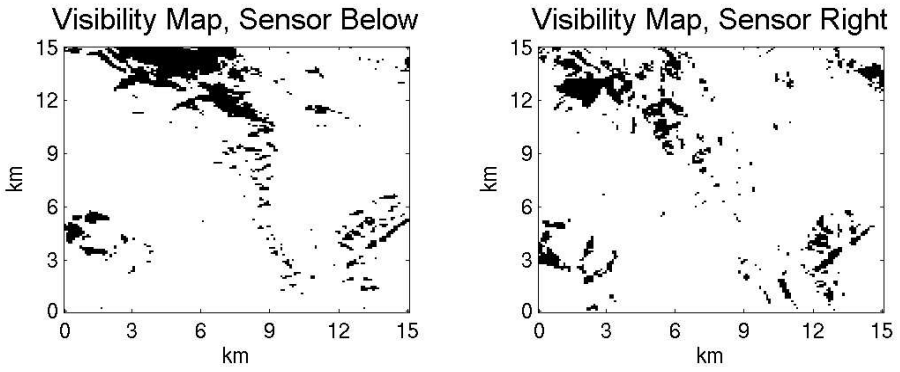


Fig. 3. Visibility masks for a sensor positioned below and to the right of the surveillance area. Non-visible areas of the surveillance region are black. Visible areas are white.

of time and then reemerges. In this case, extra sensor dwells immediately before the target enters into the obscured area (at the expense of not interrogating other targets) will sharpen the estimate of the target location. This sharpened estimate will allow for better prediction of where the target will emerge and cause extra dwells to be focussed there. We illustrate this graphically with a six time-step vignette in Figure 4.

5 Non-Myopic Sensor Management : Computational Methods

In this section, we present three methods of performing non-myopic sensor management based on information-theoretic measures. For simplicity of exposition, all methods are described in the two-step non-myopic situation.

The first method is a straightforward Monte Carlo (MC) technique that considers all two step action sequences (a_i^k, a_j^{k+1}) and computes the expected information gain for the action sequence by repeatedly simulating its application and computing the average gain acquired. While straightforward to describe, this method has a computational burden of $O(N^T)$, where N is the number of actions at each time and T is the number of time steps the algorithm looks ahead.

The second method is a MC technique that adaptively decides which paths through the action space to investigate. Given a fixed computational budget, we use an information-directed algorithm to decide which paths deserve more or less attention by the search algorithm.

The third method is an approximate technique that replaces the value-to-go with a function of visibility that captures the long term value of a current action. This function makes actions that are rewarding over the long term more desirable to chose at the current time step, thus approximating the non-myopic decision. This algorithm is $O(N)$.

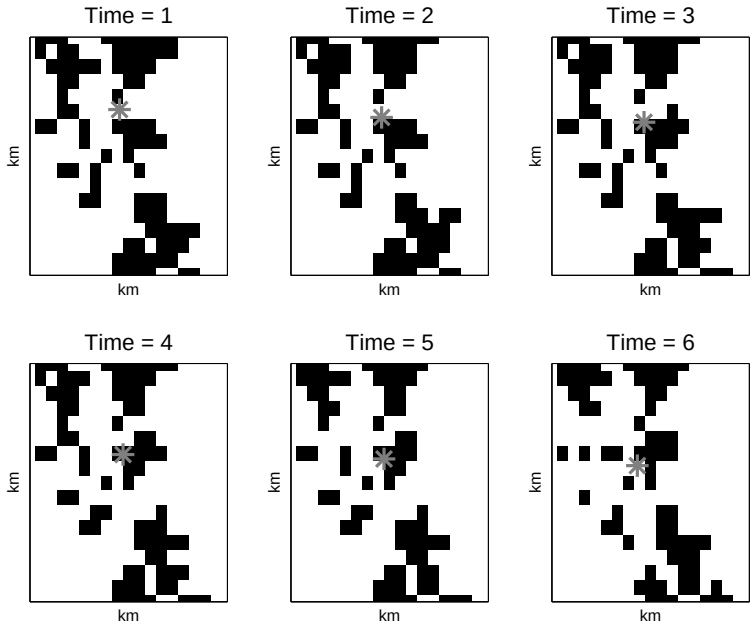


Fig. 4. A six time step vignette in which the target moves through an obscured area. The target is depicted by a gray asterisk. Obscured areas are in black and visible areas are in white. Extra sensor dwells just before becoming obscured (time = 1) should aid in relocalization after the target emerges (time = 6)

This section proceeds as follows. First, we give a Markov Decision Process expression for the non-myopic benefit of choosing an action at the current time. Next, we detail the straightforward but computationally intractable MC method for solution. Third, we show the improved information-directed search which produces similar results at reduced computational cost. Finally, we detail a simple approximate technique for incorporating long-term gains into the current decision that has computational burden on the order of that of the myopic strategy.

5.1 Notation and Preliminaries

The value of state s at time k will be denoted by $V_k(s)$. In our case, the value is computed based on the action that yields the maximum expected amount of information gained. We will use $c(s,a)$ as shorthand for the myopic expected gain associated with an action a in state s, that is

$$c(s,a) \doteq < D_\alpha \left(p(\mathbf{X}^k|\mathbf{Z}^k) || p(\mathbf{X}^k|\mathbf{Z}^{k-1}) \right) >_a \tag{8}$$

where s is used as a surrogate for $p(\mathbf{X}^k|\mathbf{Z}^{k-1})$.

The Bellman equation in the discounted reward scenario is then written as

$$V_k(s) = \max_a \{c(s,a) + \gamma E_{s'}[V_{k+1}(s')]\} \tag{9}$$

where $E_{s'}[V_{k+1}(s')] = \sum_{j \in S} p(j|s,a) V_{k+1}(j)$.

The optimal non-myopic action $\hat{a}$ is then given by

$$\hat{a} = arg \max_a \{c(s,a) + \gamma E_{s'}[V_{k+1}(s')]\} \tag{10}$$

5.2 Monte Carlo Rollout for Non-myopic Sensor Management

A straightforward but computationally intractable way of solving eq. (10) is via MC rollout techniques. Here we use "rollout" in the manner of Tesauro [13], which is as a synonym for repeatedly playing out a given position in order to calculate the expected reward starting from that position. For simplicity of exposition, we first describe only the two-step non-myopic solution in this section and comment on the extension to multiple time steps at the end.

The two-step rollout procedure is shown graphically in Figure 5. We first predict the target density at the measurement time $(k+1)$ by performing model update as in the myopic scheme (Figure 1). The prediction density, $p(\mathbf{X}^{k+1}|\mathbf{Z}^k)$ is used to determine all possible actions at time $k+1$, $a_1^{k+1} \cdots a_N^{k+1}$. In practice, only those actions with non-zero expected myopic gain are considered.

For each action at time $k+1$, we perform the following two steps repeatedly to generate a MC average of the information gain, which is used to approximate the expected value. First, the action is simulated resulting in a measurement $\hat{z}^{k+1}$. The density of $\hat{z}^{k+1}$ is formed from $p(\mathbf{X}^{k+1}|\mathbf{Z}^k)$, as in eq. (5). The simulated measurement is then used to update the density and form $p(\mathbf{X}^{k+1}|\mathbf{Z}^k, \hat{z}^{k+1})$. The realized gain in information from this measurement is calculated between the densities $p(\mathbf{X}^{k+1}|\mathbf{Z}^k)$ and $p(\mathbf{X}^{k+1}|\mathbf{Z}^k, \hat{z}^{k+1})$ using eq. (3).

This predicted posterior is then model updated to form the prediction density at time $k+2$, $p(\mathbf{X}^{k+2}|\mathbf{Z}^k, \hat{z}^{k+1})$. At this point, the expected one-step (myopic) gains for each possible action at time $k+2$ is generated using eq. (7). The value of action a_i^{k+1} is then the actual realized gain from time step $k+1$ to time step $k+2$ plus the mean of the expected gain at time $k+2$. We call this 2-step procedure searching the path (or trajectory) associated with the action a_i^{k+1}.

The extension to looking more than two time steps into the future is straightforward but computationally prohibitive. For example, a three-step rollout would perform an additional simulation step using $p(\mathbf{X}^{k+2}|\mathbf{Z}^k, \hat{z}^{k+1})$ to simulate a measurement $\hat{z}^{k+2}$ at time $k+2$. This would generate a predicted posterior at time $k+2$, $p(\mathbf{X}^{k+2}|\mathbf{Z}^k, \hat{z}^{k+1}, \hat{z}^{k+2})$. A model update would form $p(\mathbf{X}^{k+3}|\mathbf{Z}^k, \hat{z}^{k+1}, \hat{z}^{k+2})$, and the expected myopic gain at time $k+3$ would be calculated. This procedure would be repeated for each action at time $k+1$ many times to generate a MC average of the expected gain for making that measurement.

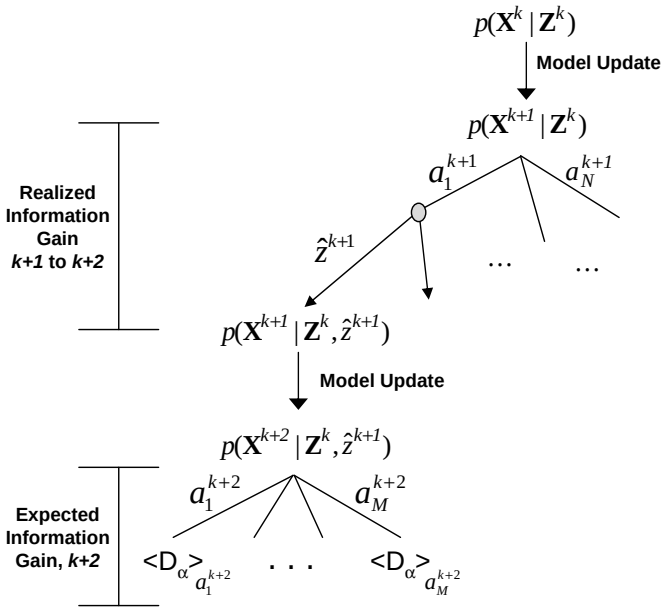


Fig. 5. The two-step non-myopic algorithm is rolled out for all possible actions at time $k+1$. The value of an action at time $k+1$ is taken to be the realized gain from the action plus the expected gain at the next step. This procedure is run many times to generate a MC average of the two-step gain for each action a_i^{k+1}.

5.3 Adaptive Trajectory Selection for Improved MC Rollout

In this section, we describe a method of performing the MC rollout discussed above where we restrict ourselves to searching down the tree only a small number of times. Given this computational budget, we wish to adaptively determine the best trajectories to investigate.

At time $k+1$, there are N possible actions. Each action corresponds to the first step in a trajectory down the tree. Associated with each action is an expected (long-term) gain in information for actually executing that action, and we wish to determine this as precisely as possible. In section 5.2 we determined this gain by simply searching down each path many times and using the empirical average of information gain as a surrogate for the expected information gain. We then executed the action with the largest long-term gain in information.

Here, rather than looking down the trajectory associated with each action an equal number of times, we wish to select the paths to search so as to best estimate the expected information gain with a fixed number of samples. We propose to select the best trajectory to simulate by computing the gain in information about the trajectory that making an additional simulation will garner.

We define the density $p_{a_i}(g|G_{a_i})$ to be a density on the expected long-term gain in information g if we were to actually take action a_i, conditioned on the long-term information gains simulated so far from searching down trajectories

starting with action a_i, G_{a_i}. Of course, at beginning of each decision epoch, we will have not searched any trajectories yet and so $G_{a_i} = \emptyset$. Our goal is to determine $p_{a_i}(g|G_{a_i})$ for all actions a_i as accurately as possible using a fixed search budget, so that when we actually task the sensor we are tasking it to make the action that maximizes the expected long-term gain in information. In Section 5.2 we concerned ourselves with estimating only the expected value of g rather than $p_{a_i}(g|G_{a_i})$ itself. We estimated the expected value of $p_{a_i}(g|G_{a_i})$ by searching down the trajectory associated with each a_i a large and constant number of times and calculated the empirical average of information gain. Here, we focus on looking at each trajectory a variable number of times so that given we search only a fixed number of trajectories, we have the best estimate for $p_{a_i}(g|G_{a_i})$ possible for all a_i. We are determining an automatic method to prune trajectories – i.e. to decide which paths that are not worth further investigation, and which paths deserve greater attention.

At the onset, we have N possible actions and no idea which action is the best to take. We propose to construct the initial density on the expected long-term information gain for actually taking action a_i by looking down the trajectory associated with action a_i a small number of times (M) to generate samples from the density $p_{a_i}(g|G_{a_i})$. These samples from $p_{a_i}(g|G_{a_i})$ will be used to approximate $p_{a_i}(g|G_{a_i})$ in a particle filter like manner, e.g. $p_{a_i}(g|G_{a_i}) = \frac{1}{M}\sum_{p=1}^{M}\delta(g-g_p)$.

We wish to simulate an additional K trajectories to improve our estimate of the distribution of the expected long term information gain when taking action a_i, $p_{a_i}(g|G_{a_i})$. In section 5.2, we simply looked at each trajectory an equal number of times. Here, we use an information directed method for selecting which trajectory to investigate for each of the K investigations we make. The method proceeds as follows. For each action a_i, we compute the expected gain in information with respect to $p_{a_i}(g|G_{a_i})$ that making one additional simulation of that action will garner. We then investigate that path that generates the largest expected gain in information. We repeat this procedure for all K investigations that we are to make.

Formally, we can compute the expected gain in information for investigating action a_i as follows. Before investigating a new path, we have a density $p_{a_i}(g|G_{a_i})$. Assume that we have decided to investigate a particular action this investigation generated a new realization of the expected long-term gain $\hat{g}$. The updated density (by Bayes' rule) becomes

$$p_{a_i}(g|G_{a_i}, \hat{g}) = \frac{p_{a_i}(\hat{g}|g)p_{a_i}(g|G_{a_i})}{p_{a_i}(\hat{g}|G_{a_i})} \tag{11}$$

Using the Alpha-Divergence metric (eq. 3), and a method identical to that of section 3, we can determine that the expected gain in information between $p_{a_i}(g|G_{a_i})$ and $p_{a_i}(g|G_{a_i}, \hat{g})$ for searching the trajectory starting with action a_i is proportional to the entropy of the distribution associated with that action,

$$\int_g p_{a_i}(g|G_{a_i}) ln\left(p_{a_i}(g|G_{a_i})\right) dg \tag{12}$$

This yields the very intuitive result that the best trajectory to search is the trajectory associated with the highest uncertainty.

5.4 Approximating the Value-to-go

In this section, we investigate an approximate method of determining the long-term gain associated with each action. As mentioned above, the optimal method for choosing the action to make at the current time, $\hat{a}$ is by evaluating

$$\hat{a} = arg\max_{a}\{c(s,a) + \gamma E_{s'}[V_{k+1}(s')]\} \tag{13}$$

We propose here to approximate the value-to-go $E_{s'}[V_{k+1}(s')]$ by a function of visibility $N_a(s)$ which captures the long term reward of action s and is trivially computable. This yields the approximate methods

$$\hat{a} = arg\max_{a}\{\hat{c}(s,a) + \gamma N(s,a)\} \tag{14}$$

where $\hat{c}$ represents a normalized myopic gain, and

$$\hat{a} = arg\max_{a}\{c(s,a) * N(s,a)\} \tag{15}$$

The form of eq. (14) looks like a direct approximation of the optimal solution of eq. (10). The myopic gain $c(s,a)$ is normalized so that the maximum expected gain is 1. The approximation of eq. (15) is also natural, as the Rényi information divergence adheres to the geometric mean rather than the algebraic mean.

$N(s,a)$ should depress the value of actions associated with low long-term rewards while amplifying actions associated with large long-term rewards. In our current setting, the dynamics of the problem are changing through the time varying visibility map. We specialize to "actions" corresponding to measuring cells. If we denote by $Vis_k(c)$ the visibility of cell c at time k, a visibility induced long-term penalty/reward for measuring cell c may be

$$N_k(s,c) = \prod_{j=1}^{J} \gamma^j (1 - Vis_{k+j}(c)) \tag{16}$$

This long-term penalty/reward works to boost the value of measuring a cell that is to have reduced visibility in the future, while suppressing the value of measuring a cell that is to have good visibility in the future. For the two step non-myopic scheduler, the $N_k(s,c)$ becomes simply $N_k(s,c) = \gamma(1 - Vis_{k+1}(c))$. Algorithms implemented with this penalty/reward remain $O(N)$.

In our simulations, we assume the elevation of the terrain and sensor trajectory are known precisely, and therefore the future visibility is exactly calculable. This easily extends to the case where only an estimate of visibility is present, in which case eq. (16) might use the conditional expectation $E(Vis_{j+k}(c)|\mathbf{Z}^k)$ which would allow the most up-to-date estimate of visibility to be used.

6 Simulation Results

We investigate the following model problem, which is inspired by the scenario depicted in Section 4. There are two targets which are each described by a one-dimensional position. Target 1 is initially positioned at $x = 2.1$ and Target 2 is initially positioned at $x = 14.9$. The targets move slowly with respect to the sensor resolution so that they stay in the same cell for the entire simulation.

For each dwell, the sensor may measure any one of 16 cells, each of which is 1 unit wide. The cell locations are fixed and centered at $.5, 1.5, \cdots, 15.5$ units. The sensor is allowed to make three (not necessarily distinct) dwells per time step. The sensor receives binary returns from the cell interrogated, which are independent from dwell to dwell. In cells that are occupied, a detection is received with probability P_d (set here at 0.9). In cells that are unoccupied a detection is received with probability P_f (set here at .01). This corresponds to a signal to noise ratio of 16dB, assuming Rayleigh distributed threshold detected returns.

At the onset, the positions of the two targets are known only probabilistically to the filter. The filter is initialized with the probability of target 1 location uniformly distributed across sensor cells $\{2 \cdots 6\}$ and the probability of target 2 location uniformly distributed across sensor cells $\{11 \cdots 15\}$.

The visibility of the various sensor cells is constructed to change in the following manner. At time step 1, all cells are visible to the sensor. At time steps $2, 3, 4$, cells $\{11 \cdots 15\}$ are invisible to the sensor. At time step 5 all cells are visible to the sensor again. This model problem closely emulates the situation where a target is initially visible to the sensor, becomes obscured, and then reemerges from the obscuration. This scenario can benefit from non-myopic scheduling as looking in the cells that are about to become obscured preferentially will minimize total track error at the end of the simulation.

	0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16
	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7	Cell 8	Cell 9	Cell 10	Cell 11	Cell 12	Cell 13	Cell 14	Cell 15	Cell 16
Time 1		X													X	
Time 2																
Time 3																
Time 4																
Time 5																

Fig. 6. An illustration of the model problem. At the onset, the filter has its position estimates of target 1 and target 2 uniformly distributed across cells {2...6} and {11...15}, respectively. At time 1 all cells are visible to the sensor. At time 2,3, and 4 cells {11...15} are obscured. This situation emulates the the situation where one target is initially visible to the sensor, becomes obscured and then reemerges.

We anticipate at time 1 the myopic strategy, having no information about the future visibility, will choose cells uniformly from the set $\{2...6\} \cup \{11...15\}$. As a result, target 1 and target 2 will on the average be given equal attention. The non-myopic strategy, however, should preferentially choose cells from {11...15} as they are to become invisible while cells {2...6} are not.

6.1 Results Using Information Directed Trajectory Interrogation

We present a comparison between uniform searching of all paths suggested in Section 5.2 with the information-directed search algorithm of Section 5.3 in Figure 7. Performance is compared in terms of median error after time 4 versus number of paths searched (which measures algorithm complexity). As expected, as either algorithm searches more paths, better decisions are made resulting in lower tracking error. However, uniform search requires more paths to be searched to yield a desired error since it wastes investigations on paths of little value.

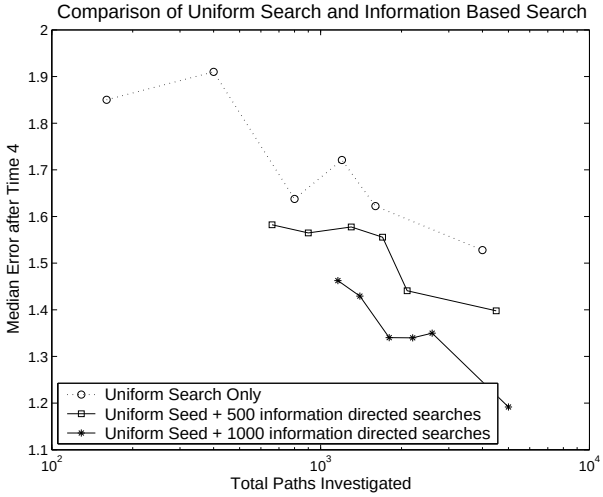
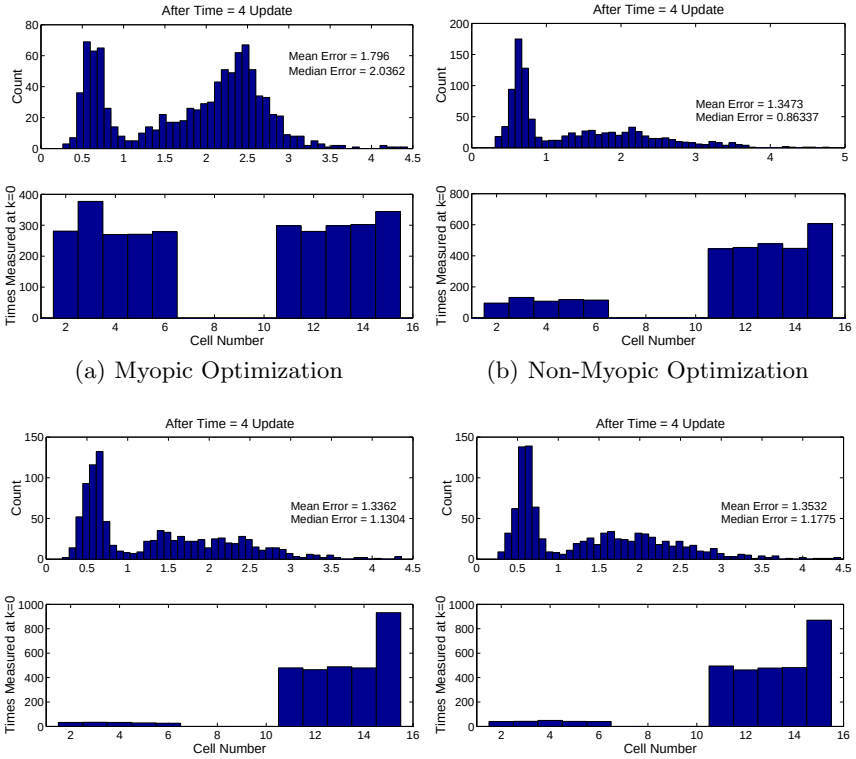


Fig. 7. A comparison between uniform MC and information-directed search. This graph contains three curves. The top curve contains the results of searching each path equally (uniform search) where the number of searches of each path is varied between 10 and 250 to produce the six points on the curve. The bottom two curves are each seeded with uniform search and followed by information-directed searches. The number of uniform searches is again varied between 10 and 250 to produce the six points on each curve. The results are plotted against the total number of paths searched, which is a measure of algorithm complexity. For a given number of paths searched, information-directed search yields better performance.

6.2 Results Using Approximation of Value-to-go

We compare here the performance of the myopic sensor management, exhaustive MC search and the two approximate methods based on replacing the value-to-go with a function of visibility. in Figure 8. For each method, we show two measures of performance. First, we provide a histogram of the errors after time 4 along with the mean and median error. Second, we provide a histogram of the sensor cell decisions made at time 1 for each of the methods.



(a) Myopic Optimization (b) Non-Myopic Optimization

(c) Non-Myopic Approximation (eq. 14) (d) Non-Myopic Approximation (eq. 15)

Fig. 8. The tracking performance of myopic, non-myopic MC, and approximate methods. For each method we illustrate the performance with two metrics. First (top) we histogram tracking error. Non-myopic MC and approximate methods outperform the myopic method. Second (bot), we show the allocation of sensor resources at time = 0. The non-myopic and approximate methods correctly look at the region that is about to become obscured whereas the myopic method allocates dwells uniformly.

The non-myopic strategy of Section 5.2 dramatically reduces target localization error at time step 4. Furthermore, the approximate techniques yield results similar to the MC non-myopic strategy at a computational cost similar to the myopic strategy. The performance of the various strategies is summarized in Table 1 below.

References

1. S. Musick and R. Malhotra, "Chasing the Elusive Sensor Manager", *Proceedings of NAECON*, Dayton, OH, May 1994, pp. 606-613.
2. J. Liu, P. Cheung, L. Guibas, and F. Zhao, "A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks", ACM International Workshop on Wireless Sensor Networks and Applications Workshop, Atlanta, September 2002.

Table 1. The performance of the four sensor management strategies considered.

Method	CPU Time (sec)	Mean Error (cells)	Median Error (cells)
Myopic	0.31	1.80	2.04
MC Non-myopic	102.54	1.35	0.86
Non-myopic approx. eq. (14)	0.32	1.34	1.13
Non-myopic approx. eq. (15)	0.32	1.35	1.18

3. C. Kreucher, K. Kastella, and A. O. Hero III, "Sensor Management Using Relevance Feedback Learning", under review at *IEEE Transactions on Signal Processing*.
4. C. Kreucher, K. Kastella, and A. O. Hero III, "Multi-target Sensor Management Using Alpha-Divergence Measures", *The Second Annual Symposium on Information Processing in Sensor Networks (IPSN '03)*, April 22-23, 2003.
5. V. Krishnamurthy, "Algorithms for Optimal Scheduling and Management of Hidden Markov Model Sensors", *IEEE Transactions on Signal Processing*, Vol. 50, no. 6, pp. 1382-1397, June 2002.
6. V. Krishnamurthy and D. Evans, "Hidden Markov Model Multiarm Bandits: A Methodology for Beam Scheduling in Multitarget Tracking", *IEEE Transactions on Signal Processing*, Vol. 49, no. 12, pp. 2893-2908, December 2001.
7. D. P. Bertsekas and D. Castanon, "Rollout Algorithms for Stochastic Scheduling Problems", *Journal of Heuristics*, Vol. 5, no. 1, pp. 89-108, 1999.
8. D. Castanon, "Approximate Dynamic Programming for Sensor Management", *Proceedings of the 1997 Conference on Decision and Control*, 1997.
9. D. Castanon, "Optimal Search Strategies for Dynamic Hypothesis Testing", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 7, pp. 1130-1138, 1995.
10. K. J. Hintz, "A Measure of the Information Gain Attributable to Cueing", *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 21, no. 2, pp. 237-244, March/April 1991.
11. C. Kreucher, K. Kastella and A. O. Hero III, "Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density", SPIE Annual Meeting, San Diego, California, August 2003.
12. A. Rényi, "On measures of entropy and information", Proc. 4th Berkeley Symp. Math. Stat. and Prob., volume 1, pp. 547-561, 1961.
13. G. Tesauro, and G. R. Galperin, "On-Line Policy Improvement Using Monte Carlo Search", The 1996 Neural Information Processing Systems Conference, Denver, CO.
14. A. O. Hero, B. Ma, O. Michel and J. Gorman, "Applications of entropic spanning graphs," IEEE Signal Processing Magazine (Special Issue on Mathematics in Imaging), Vol 19, No. 5, pp 85-95, Sept. 2002.
15. Doucet, A. de Freitas, N., and Gordon, N. "Sequential Monte Carlo Methods in Practice", Springer Publishing, New York, 2001.
16. F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration", IEEE Signal Processing Magazine, March 2002, pp. 61-72.
17. V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic Path Planning in Sensor-Based Terrain Acquisition, IEEE Transactions on Robotics and Automation, vol. 6, no. 4, pp. 462-472, August 1990.