

# A Bayesian Method for Integrated Multitarget Tracking and Sensor Management

Chris Kreucher and Keith Kastella  
Veridian Systems Division  
Ann Arbor, MI  
Christopher.Kreucher@veridian.com  
Keith.Kastella@veridian.com

Alfred O. Hero III  
The University of Michigan  
Department of EECS  
Ann Arbor, MI  
hero@eecs.umich.edu

**Abstract** – *This paper presents an integrated method for target tracking and sensor management, applied to the problem of tracking multiple ground targets. We use a multiple target tracking methodology based on recursive estimation of a Joint Multitarget Probability Density (JMPD) which is implemented using particle filtering methods. This Bayesian method for tracking multiple targets allows nonlinear, non-Gaussian target motion and measurement-to-state coupling. The 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. The Rényi Information Divergence, a generalization of the Kullback-Leibler Distance, provides a way to measure the dissimilarity between two densities. Sensor management then proceeds by evaluating the expected information gain for each of the possible measurement decisions, and selecting to make the measurement that maximizes the expected information gain.*

**Keywords:** Tracking, Nonlinear Filtering, Sensor Management, Bayesian Methods.

## 1 Introduction

The problem of sensor management is to determine the best way to task a sensor where the sensor may have many modes and may be pointed in many directions. This problem has recently enjoyed a great deal of interest [1]. A typical application, and one that we focus on in our model problem, is to direct an electronically scanned aperture (ESA) radar [2]. An ESA provides great flexibility in pointing and mode selection. For example, the beam can be redirected in a few microseconds, enabling targets to be illuminated at will.

One way of designing a sensor management system is by optimizing information flow. This is analogous

to designing a communications system to maximize the channel capacity. Past work in this area has been based on maximizing Kullback-Leibler (KL) divergence. The KL divergence between two densities measures their dissimilarity, and in our application gives an indication of the difference in information between the two. In this work, we employ a more general information measure called the Rényi Information Divergence (also known as the  $\alpha$ -divergence), which reduces to the KL divergence under a certain limit. The  $\alpha$  parameter in the Rényi divergence provides an additional flexibility that allows for emphasis to be placed on specific portions of the densities.

The algorithm that we propose for sensor tasking is thus motivated by information theory. First, we utilize a target tracking algorithm that recursively estimates the joint multitarget probability density (JMPD) for the set of targets under surveillance. We then strive to task the sensor in such a way that the sensing action it makes results in the maximum amount of information gain. To that end, we employ the Rényi information divergence [3] as a measure of distance between two densities. The decision as to how to use a sensor then becomes one of determining which sensing action will maximize the expected information gain between the current joint multitarget probability density and the joint multitarget probability density after a measurement has been made.

The paper is organized as follows. In Section 2, we present the target tracking algorithm that is central to our sensor management scheme. Specifically, we give the details of the JMPD and examine the numerical difficulties involved in directly implementing JMPD on a grid. In Section 3, we present a particle filter (PF) based implementation of JMPD. We see that this implementation provides for computational tractability, allowing realistic simulations to be made with modern computing power. Particle filtering techniques have

been used in multi-target tracking applications in the past [11], although our method differs from traditional methods in several important aspects. First, there is no measurement to target association involved. Furthermore, each particle is a sample from the joint multitarget density and therefore is an estimate of both the number of targets present and their states. Our sensor management scheme, based on maximizing the expected Rényi Information Divergence is given in Section 4. A comparison of the performance of the tracker using sensor management to the tracker using a non-managed scheme on a model problem is given in Section 5. We conclude with some thoughts on future direction in Section 6.

## 2 The Joint Multitarget Probability Density (JMPD)

In this section, we introduce the details of using the Joint Multitarget Probability Density (JMPD) for target tracking. The concept of JMPD was first discussed by Kastella in [4], where a method of tracking multiple targets that moved between discrete cells on a line based on a set of sensor measurements was presented. We generalize the discussion here to deal with targets that have  $N$ -dimensional continuous valued state vectors. In the model problem, we are interested in tracking the position  $(x, y)$  and velocity  $(\dot{x}, \dot{y})$  of multiple targets and so we describe targets by the four dimensional state vector  $[x, \dot{x}, y, \dot{y}]'$ .

JMPD provides a means for tracking an unknown number of targets in a Bayesian setting. The statistical model uses the joint multitarget conditional probability density  $p(\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k | \mathbf{Z}^k)$  as the probability density for exactly  $T$  targets with states  $\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k$  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 of the  $T$  targets. The observation set  $\mathbf{Z}^k$  refers to the collection of measurements up to and including time  $k$ , i.e.  $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^k\}$ , where each of the  $\mathbf{z}^i$  may be a single measurement or a vector of measurements made at time  $i$ .

Each of the state vectors  $\mathbf{x}_i$  in the density  $p(\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k | \mathbf{Z}^k)$  is a vector quantity and may (for example) be of the form  $[x, \dot{x}, y, \dot{y}]'$ . We refer to each of the  $T$  target state vectors  $\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k$  as a partition of the state  $\mathbf{X}$ . For convenience, the density will be written more compactly in the traditional manner as

$$p(\mathbf{X}^k | \mathbf{Z}^k) \quad (1)$$

With the understanding that the state-vector  $\mathbf{X}$  represents a variable number of targets each possessing

their own state vector. As an illustration, some examples illustrating the sample space of  $p$  are

$p(\emptyset | \mathbf{Z})$ , the posterior probability density for no targets in the surveillance volume

$p(\mathbf{x}_1 | \mathbf{Z})$ , the posterior probability density for one target in state  $\mathbf{x}_1$

$p(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{Z})$ , the posterior probability density for two targets in states  $\mathbf{x}_1$  and  $\mathbf{x}_2$

$p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 | \mathbf{Z})$ , the posterior probability density for three targets in states  $\mathbf{x}_1, \mathbf{x}_2$  and  $\mathbf{x}_3$

The temporal update of the posterior likelihood on this density proceeds according to the usual rules of Bayesian filtering. Using a model of target kinematics

$$p(\mathbf{X}^k | \mathbf{X}^{k-1}) \quad (2)$$

we may compute the time-updated or prediction density via

$$p(\mathbf{X}^k | \mathbf{Z}^{k-1}) = \int d\mathbf{X}^{k-1} p(\mathbf{X}^k | \mathbf{X}^{k-1}) p(\mathbf{X}^{k-1} | \mathbf{Z}^{k-1}) \quad (3)$$

Bayes' rule enables us to update the posterior density as new measurements  $\mathbf{z}^k$  arrive as

$$p(\mathbf{X}^k | \mathbf{Z}^k) = \frac{p(\mathbf{z}^k | \mathbf{X}^k) p(\mathbf{X}^k | \mathbf{Z}^{k-1})}{p(\mathbf{z}^k | \mathbf{Z}^{k-1})} \quad (4)$$

Notice that there is no measurement to target association necessary to update the JMPD.

The probabilistic sensor model is denoted by

$$p(\mathbf{z}^k | \mathbf{X}^k) \quad (5)$$

As illustrated above, the sample space of  $\mathbf{X}^k$  is very large – it consists of all possible configurations of state vectors  $\mathbf{x}_i$  for all possible values of  $T$ . The original formulation of JMPD given in [4] approximated the posterior density by discretizing on a grid. It was immediately found that the computational burden in this scenario makes evaluating realistic problems intractable, even when using the simple model of targets moving between discrete locations in one-dimension. In fact, the number grid cells needed grows as  $Locations^{Targets}$ , where  $Locations$  is the number of discrete locations the targets may occupy and  $Targets$  is the number of targets.

Thus, we need a method for approximating the JMPD that leads to more tractable computational burden. In the next section, we show that the Monte Carlo methods collectively known as particle filtering break this computational barrier.

### 3 Particle Filter Implementation of JMPD

We anticipate that a particle filter based implementation of JMPD will break the computational logjam and allow us to investigate more realistic problems. To implement JMPD using particle filter techniques, we first approximate the joint multitarget probability density  $p(\mathbf{X}|\mathbf{Z})$  by a set of  $N_{part}$  weighted samples,  $\mathbf{X}_p$ , ( $p = 1 \dots N_{part}$ ):

$$p(\mathbf{X}|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \quad (6)$$

Here we have suppressed the time superscript  $k$  everywhere for notational simplicity. We will continue to do this whenever time is not relevant to the discussion at hand.

Recall from Section 2 that our multitarget state vector  $\mathbf{X}$  contains  $T$  partitions, each corresponding to a target:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{T-1}, \mathbf{x}_T] \quad (7)$$

Furthermore, the joint multitarget probability density  $p(\mathbf{X}|\mathbf{Z})$  is defined for  $T = 0 \dots \infty$ . Each of the particles  $\mathbf{X}_p$ ,  $p = 1 \dots N_{part}$  is a sample drawn from  $p(\mathbf{X}|\mathbf{Z})$ . Therefore, a particle  $\mathbf{X}_p$  may have  $0, 1, \dots \infty$  partitions, each partition corresponding to a different target. We will denote the number of partitions in particle  $\mathbf{X}_p$  by  $n_p$ , where  $n_p$  may be different for different  $\mathbf{X}_p$ , representing an uncertainty in the estimate of the number of targets in the surveillance region. Since a partition corresponds to a target, the number of partitions that a particle has is that particle's estimate of the number of targets in the surveillance area.

To make our notation more concrete, assume that each target is modeled using the state vector  $\mathbf{x} = [x, \dot{x}, y, \dot{y}]'$ . Then a particular  $\mathbf{X}_p$ , which is tracking  $n_p$  targets, will be given as

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \dots, \mathbf{x}_{p,n_p}] = \begin{pmatrix} x_{p,1} & x_{p,2} & \dots & x_{p,n_p} \\ \dot{x}_{p,1} & \dot{x}_{p,2} & \dots & \dot{x}_{p,n_p} \\ y_{p,1} & y_{p,2} & \dots & y_{p,n_p} \\ \dot{y}_{p,1} & \dot{y}_{p,2} & \dots & \dot{y}_{p,n_p} \end{pmatrix} \quad (8)$$

Where here we expand the notation slightly and use  $x_{p,1}$  to denote the  $x$  position estimate that particle  $p$  has of target 1.

Notice that this method differs from traditional particle filter tracking algorithms where a single particle corresponds to a single target. We find that when each particle is attached to a single target, some targets become particle starved over time. All of the particles

tend to attach to the target receiving the best measurements. Our method explicitly enforces the multitarget nature of the problem by encoding in each particle the estimate of the number of targets and the states of those targets. This helps to alleviate the particle starvation issue.

Note there is a permutation symmetry inherent in JMPD, i.e.  $p(\mathbf{x}_1, \mathbf{x}_2|\mathbf{Z}) = p(\mathbf{x}_2, \mathbf{x}_1|\mathbf{Z})$ . This is particularly relevant when targets are near each other and particle partitions begin to swap allegiances. We will have more to say about this issue in Section 3.5.

In the following subsections, we give the specifics of our particle filter implementation of JMPD.

#### 3.1 Initialization

An uninformed method of initialization would be to randomly choose particle locations in the surveillance area and use the measurements as weights. In the multitarget scenario where the number of targets is unknown, this would involve randomly choosing both the states and number of targets. This is very inefficient, especially for large numbers of targets and large surveillance volumes. Instead, we incorporate the measurement likelihood directly into our initialization.

The basic idea of our initialization technique is to propose particles that have targets which are biased preferentially towards high likelihood regions. The partitions of the multitarget particle are formed by selecting from these high likelihood regions and the bias is removed during resampling.

The method proceeds as follows. First, each target (partition) in each particle is generated by sampling from the likelihood given by the first scan of the region. Particles are given weights based on their likelihood and resampled to remove the bias. There is no guarantee that the initial particle set will have reasonable density near the true multitarget state after just a single scan. In fact, at low signal to noise ratio (SNR), this is unlikely to occur on any given scan. Therefore, after the first scan a portion of the particles are time updated according to the tracking algorithm, and some new particles are generated according to the detection scheme. The particle population is then formed as a combination of newly detected particles and particles that are being tracked. As time goes on, fewer and fewer particles are generated via the detection scheme until finally all particles are being maintained by the tracking algorithm.

#### 3.2 Particle Proposal

Several methods of particle proposal have been investigated. The standard method used, which will be referred to as sampling from the kinematic prior, proposes new particles at time  $k$ ,  $\mathbf{X}_p^k$ , according

to the traditional Sampling-Importance Resampling (SIR) method. For each particle at time  $k-1$ ,  $\mathbf{X}_p^{k-1}$ , a new particle  $\mathbf{X}_p^k$  is generated using the kinematic prior  $p(\mathbf{X}^k|\mathbf{X}^{k-1})$ .

In the case where the targets are indistinguishable, each target in  $\mathbf{X}_p^{k-1}$  behaves according to the same motion model and is proposed according to the Kinematic model. In the more general case where targets are distinguishable (perhaps due to a sensor capable of target classification, as in [10]) different kinematic models may be used for each of the partitions in a particle. This amounts to having a different form of equation (3) for different types of targets.

In addition to this simple method, we have investigated alternate particle proposal techniques, all of which are developed as a means of biasing the proposal process towards the measurements.

The coupled-partition (CP) proposal method proposes  $M$  possible realizations (futures) for each partition of a particle. We use the notation  $\mathbf{X}_{p,j}^k(m)$  to refer to the  $m^{\text{th}}$  proposal for the  $j^{\text{th}}$  partition of the  $p^{\text{th}}$  particle at time  $k$ . See equation (8) for a concrete example of a particle and its partitions. The proposed partitions are then given weights according to the likelihood, and a new particle  $\mathbf{X}_p^k$  is chosen by selecting a representative from each of the proposed partition sets.

Alternatively, we utilize the independent-partition (IP) method given in [9] proposes new partitions and weights each partition independently. Particles at time  $k$ ,  $\mathbf{X}_p^k$ , are formed by selecting partitions from the set of weighted proposed partitions from the particles at time  $k-1$ . This method assumes that the targets states are independent, which is not the case when targets cross.

We have found that an adaptive method, which automatically selects between the IP and CP methods gives the best tradeoff of performance versus efficiency. This method identifies which targets are acting independently and which are not based on their proximity in measurement space. Those partitions (targets) of a particle that are well separated are treated as independent as proposed via the IP method. The remaining partitions that are not well separated are proposed using the CP method.

Finally in any of these methods, target birth and death may be accounted for by modifying the proposal density to incorporate a probability that the proposed particle  $\mathbf{X}_p^k$  has either fewer or more targets than  $\mathbf{X}_p^{k-1}$ . This is useful in scenarios where targets may enter or leave the surveillance area.

### 3.3 Measurement Update

Each proposed particle is given a weight according to its agreement with the measurements, the kinematic

model, and the importance density [6]. Since we are proposing particles based on  $p(\mathbf{X}^k|\mathbf{X}^{k-1})$ , it can be shown that the proper weighting is given by

$$w_p \propto p(\mathbf{z}|\mathbf{X}_p^k) \quad (9)$$

The IP method detailed above requires a slight modification to this weight to account for the bias introduced in proposal process – see [9].

Each particle  $\mathbf{X}_p^k$  simultaneously postulates that a specific number of targets exist in the surveillance region ( $n_p$ ) and that the target states are given by  $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_p-1}, \mathbf{x}_{n_p}]$ . In the case where the measurement set is made up of a scan  $i$  cells (say for example on a XY grid) where the measurement in each cell is independent of the measurements in the other cells, we compute the weight as

$$w_p \propto \prod_i p(z_i|\mathbf{X}_p) \quad (10)$$

where in this notation  $z_i$  refers to the measurement made in cell  $i$ . A particular particle  $\mathbf{X}_p$  will postulate that there are targets in some cells  $i_x$  (not necessarily distinct):

$$i_x = i_1, i_2, \dots, i_{n_p} \quad (11)$$

We denote the measurement density when there are 0 targets present as  $p_0$ , and simplify the weight equation as

$$w_p \propto \prod_{i \notin i_x} p_0(z_i) \prod_{i \in i_x} p(z_i|\mathbf{X}_p) \quad (12)$$

$$w_p \propto \prod_i p_0(z_i) \prod_{i \in i_x} \frac{p(z_i|\mathbf{X}_p)}{p_0(z_i)} \quad (13)$$

$$w_p \propto \prod_{i \in i_x} \frac{p(z_i|\mathbf{X}_p)}{p_0(z_i)} \quad (14)$$

If we let  $O_{i,p}$  (the occupation number) denote the number of targets that a particle  $p$  postulates exist in cell  $i$ , then we write the weight as

$$w_p \propto \prod_{i \in i_x} \frac{p(z_i|O_{i,p})}{p_0(z_i)} \quad (15)$$

### 3.4 Resampling

After each measurement update,  $N_{part}$  particles are selected with replacement from the existing set of particles based on the weights  $w_p$ . We then have a collection of  $N_{part}$  particles with uniform weight that approximate the density  $p(\mathbf{X}|\mathbf{Z})$ . At this step, particles that do not correspond to measurements are eliminated – in particular, particles that have an  $n_p$  that is unsupported by measurements are not retained.

The particular resampling that we have implemented is called systematic resampling [6]. We like this scheme because it is easy to implement, runs in  $O(N)$ , is unbiased, and minimizes Monte Carlo variance. Many other resampling schemes and modifications are presented in the literature [7]. Of the methods we have investigated, we have found that adaptive resampling [8] and Markov Chain Monte Carlo (MCMC) moves using a Metropolis-Hasting scheme [7] lead to improved performance over straightforward resampling in our application.

### 3.5 Estimation

Estimates of various interesting quantities may be easily made using the particles. Estimation is best performed before resampling, as resampling has been shown to only increase the variance of the estimate.

To compute the probability that there are exactly  $n$  targets in the surveillance volume, first define the indicator variable

$$I_p = \begin{cases} 1 & \text{if } n_p = n \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Then the probability of  $n$  targets in the surveillance volume,  $p(n|\mathbf{Z})$ , is given by

$$p(n|\mathbf{Z}) = \sum_{p=1}^{N_{part}} I_p w_p \quad (17)$$

So to estimate the probability of  $n$  targets in the surveillance volume, we sum up the weights of the particles that have  $n$  partitions.

To compute the estimated state and covariance of target  $i$ , we first define a second indicator variable  $\tilde{I}_p$  that indicates if particle  $p$  has a partition corresponding to target  $i$ :

$$\tilde{I}_p = \begin{cases} 1 & \text{if } n_p \geq n \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Furthermore, we define the normalized weights to be

$$\hat{w}_p = \frac{w_p \tilde{I}_p}{\sum_{l=1}^{N_{part}} \tilde{I}_l w_l} \quad (19)$$

So  $\hat{w}_p$  is the relative weight of particle  $p$ , with respect to all particles tracking target  $i$ . Then the estimate of the state of target  $i$  is given by

$$\hat{\mathbf{X}}(i) = E[\mathbf{X}(i)] = \sum_{p=1}^{N_{part}} \tilde{I}_p \hat{w}_p \mathbf{X}_{p,i} \quad (20)$$

Which is simply the weighted summation of the position estimates from those particles that are tracking target  $i$ . The covariance is given by

$$\hat{\Lambda}(i) = \sum_{p=1}^{N_{part}} \tilde{I}_p \hat{w}_p (\mathbf{X}_{p,i} - \hat{\mathbf{X}}(i)) (\mathbf{X}_{p,i} - \hat{\mathbf{X}}(i))' \quad (21)$$

The summations in (20) and (21) are taken over only those particles that are tracking target  $i$ . The permutation symmetry issue mentioned in Section 3 comes to the forefront here. Notice that it is not necessarily true that partition  $i$  of particle  $j$  is tracking the same target that partition  $i$  of particle  $j+1$  is tracking. Therefore, before evaluation of or can be made, a sorting of the partitions in each particle must be accomplished. This is affected by storing information about the trajectories of each of the partitions and clustering based on the current and past states.

## 4 Rényi Information Divergence For Sensor Management

Our information based method for tasking the sensor is to choose the sensing action that maximizes the expected information gain. To that end, our algorithm proceeds by first enumerating all possible sensing actions. A sensing action may consist of choosing a particular mode (i.e. SAR mode versus GMTI mode), a particular dwell point, or a combination of the two. We next calculate the *expected* information gain in making each of the possible sensing actions, and select to take the action that yields the maximum expected information gain.

In general, the calculation of information gain between two densities  $f_1$  and  $f_0$  is done using the Rényi information divergence (22), also known as the  $\alpha$ -divergence:

$$D_\alpha(f_1||f_0) = \frac{1}{\alpha-1} \ln \int f_1^\alpha(x) f_0^{1-\alpha}(x) dx \quad (22)$$

The function  $D_\alpha$  given in equation (22) is a measure of the divergence between the two densities  $f_0$  and  $f_1$ . In our application, we are interested in computing the divergence between the predicted density  $p(\mathbf{X}|\mathbf{Z}^{k-1})$  and the updated density after a measurement is made,  $p(\mathbf{X}|\mathbf{Z}^k)$ . Therefore, we write

$$D_\alpha(p(\mathbf{X}|\mathbf{Z}^k)||p(\mathbf{X}|\mathbf{Z}^{k-1})) = \frac{1}{\alpha-1} \ln \sum_{\mathbf{X}} p(\mathbf{X}|\mathbf{Z}^k)^\alpha p(\mathbf{X}|\mathbf{Z}^{k-1})^{1-\alpha} \quad (23)$$

The integral in equation (22) reduces to a summation since any discrete approximation of the density  $p(\mathbf{X}|\mathbf{Z}^{k-1})$  only has nonzero probability at a finite number of target states. After some algebra and the

incorporation of Bayes rule (eq. 4), this quantity can be simplified to

$$D_\alpha (p(\mathbf{X}|\mathbf{Z}^k)||p(\mathbf{X}|\mathbf{Z}^{k-1})) = \frac{1}{\alpha-1} \ln \frac{1}{p(\mathbf{z}|\mathbf{Z}^{k-1})^\alpha} \sum_{\mathbf{X}} p(\mathbf{X}|\mathbf{Z}^{k-1}) p(\mathbf{z}|\mathbf{X})^\alpha \quad (24)$$

Our particle filter approximation of the density given by equation (6) reduces equation (24) to

$$D_\alpha (p(\mathbf{X}|\mathbf{Z}^k)||p(\mathbf{X}|\mathbf{Z}^{k-1})) = \frac{1}{\alpha-1} \ln \frac{1}{p(\mathbf{z})^\alpha} \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X}_p)^\alpha \quad (25)$$

where

$$p(\mathbf{z}) = \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X}_p) \quad (26)$$

We note in passing here that the sensor model  $p(\mathbf{z}|\mathbf{X}_p)$  is used to incorporate everything known about the sensor, including SNR, detection probabilities, and even whether the locations represented by  $\mathbf{X}_p$  are visible to the sensor.

We would like to choose to perform the measurement that makes the divergence between the current density and the density after a new measurement has been made as large as possible. This indicates that the sensing action has maximally increased the information content of the measurement updated density,  $p(\mathbf{X}|\mathbf{Z}^k)$ , with respect to the density before a measurement was made,  $p(\mathbf{X}|\mathbf{Z}^{k-1})$ .

We propose, then, as a method of sensor management calculating the expected value of equation (25) for each of the  $m$  ( $m = 1 \dots M$ ) possible sensing actions and then choosing the action that maximizes the expectation. In this notation  $m$  refers to any possible sensing action under consideration, including but not limited to sensor mode selection and sensor beam positioning. In this manner, we say that we are making the measurement that maximizes expected information gain.

The expected value of equation (25) may be written as an integral over all possible outcomes  $z_m$  when performing sensing action  $m$ :

$$\langle D_\alpha \rangle_m = \int dz_m p(z_m|\mathbf{Z}^{k-1}) D_\alpha (p(\mathbf{X}|\mathbf{Z}^k)||p(\mathbf{X}|\mathbf{Z}^{k-1})) \quad (27)$$

In the special case that we consider here where the measurements are thresholded and are therefore either

detections or no-detections (i.e.  $z = 0$  or  $z = 1$ ), this integral reduces to

$$\langle D_\alpha \rangle_m = p(z=0|\mathbf{Z}^{k-1}) D_\alpha|_{m,z=0} + p(z=1|\mathbf{Z}^{k-1}) D_\alpha|_{m,z=1} \quad (28)$$

Which, using equation (25) results in

$$\langle D_\alpha \rangle_m = \frac{1}{\alpha-1} \sum_{z=0}^1 p(z) \ln \frac{1}{p(z)^\alpha} \sum_{p=1}^{N_{part}} w_p p(z|\mathbf{X}_p)^\alpha \quad (29)$$

Implementationally, the value of equation (29) can be calculated for a host of possible actions using only a single loop through the particles. This results in a computationally efficient method for making sensor tasking decisions.

In summary, our sensor management algorithm is a recursive algorithm that proceeds as follows. At each occasion where a sensing action is to be made, we evaluate the expected information gain as given by equation (29) for each possible sensing action  $m$ . We then select and make the sensing action that gives maximal expected information gain. Notice that this is a myopic (greedy) scheme, which chooses to make the measurement that optimizes information gain only for the next time step.

## 5 Simulation Results

We test the performance of the sensor management scheme proposed here by considering the following model problem. We have three targets moving on a  $12 \times 12$  sensor grid. Each target is modeled using the four-dimensional state vector  $[x, \dot{x}, y, \dot{y}]'$ . Target motion is simulated using a constant-velocity model with a (relatively) large diffusive component. The trajectories have been shifted and time delayed so that there are two times during the simulation where targets cross paths.

The target kinematics assumed by the filter (equation 3) are a constant velocity model with a slightly larger diffusive constant than that of the simulation. At each time step, a set of  $L$  (not necessarily distinct) cells are measured by the imager. The sensor is at a fixed location above the targets and all cells are visible to the sensor at all times in the simulation. When measuring a cell, the imager returns either a 0 (no detection) or a 1 (detection) governed by  $P_d$ ,  $P_f$ , and  $SNR$ . This model is known by the filter and used to evaluate equation (4).

We contrast in this section the performance of the tracker when the sensor uses a non-managed (periodic) scheme versus the performance when the sensor uses

the management scheme presented in Section 4. The periodic scheme behaves as follows. At time 1, cells  $1 \dots L$  are measured. At time 2, cells  $L + 1 \dots 2L$  are measured. This sequence repeats until all cells have been measured, at which time the scheme resets. The managed scheme uses the expected information divergence to calculate the best  $L$  cells to measure at each time. All simulations shown in this section are done with  $\alpha \rightarrow 1$  in equation (22), corresponding to the KL divergence.

In Fig. 1, we give a single-time snapshot, which graphically illustrates the difference between the two schemes. On the left, we show the managed scheme and on the right the periodic scheme. In both panes, the three targets are marked with an asterisk, the covariance ellipses of the estimated target position are shown, and we use grayscale to indicate the number of times each cell has been measured at this time step. In the managed case, sensor dwells are used in the areas that contain the targets which results in much tighter covariance ellipses.

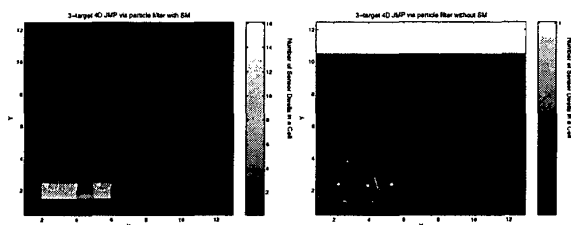


Figure 1: A Comparison of Non-Managed and Managed Tracking. (L) Using Sensor Management, (R) Using a Periodic Scheme.

Qualitatively, we see that in the managed scenario the measurements are focused in or near the cells that the targets are in. Furthermore, the covariance ellipses, which reflect the current state of knowledge of the tracker conditioned on all previous measurements, are much tighter in the managed scenario. In fact, the non-managed scenario has confusion about which tracks correspond to which target as the covariance ellipses overlap.

For a more detailed comparison, we provide the Monte Carlo simulation results shown in Figure 2 and Figure 3. The sensor management algorithm detailed earlier was run with  $L = 24$  (i.e. was able to scan 24 cells at each time step) and is compared to the non-managed scheme with 24 to 312 looks. Figures 2 and 3 show that the unmanaged scenario needs approximately 312 looks to equal the performance of the managed algorithm in terms of root mean square error (RMSE). We say that the sensor manager is approx-

imately 13 times as efficient as allocating the sensors without management.

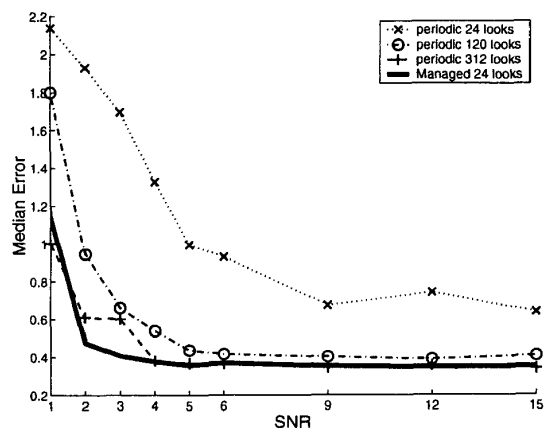


Figure 2: Median Error versus Signal To Noise Ratio (SNR). Managed Performance with 24 Looks is Similar to Unmanaged Performance with 312 Looks.

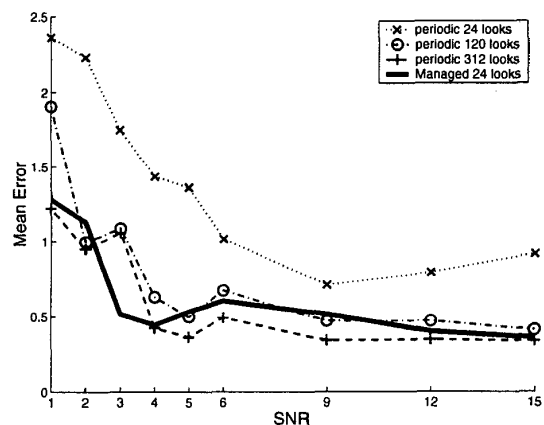


Figure 3: Mean Error versus Signal To Noise Ratio (SNR). Managed Performance with 24 Looks is Similar to Unmanaged Performance with 312 Looks.

## 6 Conclusion

We have presented in this paper a sensor management scheme based on computing the expected information gain for each sensor tasking under consideration. Simulation results show that the performance of our particle filter based multitarget tracker using sensor management with 24 looks is similar to a non-managed filter using 312 looks. This means that the

same tracking capability is achievable with sensor management using (approximately) a factor of 13 fewer sensor dwells.

There are two interesting directions in which we see this work evolving. First, this method is amenable to incorporating auxiliary information such as terrain constraints, sensor trajectories, and sensor visibility maps. For example, if the appropriate auxiliary information were incorporated, this method would clearly never choose to make a measurement in a region that was not visible to the sensor due to hill regions between the sensor and the desired look location. Visibility information, a property of the sensor enters directly into the filter through the sensor model, equation (5). Second, the current algorithm is a myopic algorithm, choosing to make the measurement that is best at the current time step. We are currently investigating Markov decision process strategies to allow non-myopic sensor management.

## 7 Acknowledgements

This work was supported under the United States Air Force contract F33615-02-C-119, Air Force Research Laboratory 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.

## References

- [1] D. Sinno and D. Kreithen, *A Constrained Joint Optimization Approach to Dynamic Sensor Configuration*, Thirty Six Asilomar Conference on Signals, Systems, and Computers, November 2002.
- [2] S. S. Blackman, *Multiple-Target Tracking with Radar Applications*, Norwood, MA. Artech House, 1986.
- [3] A. Rényi, *On measures of entropy and information*, in Proc. 4th Berkeley Symp. Math. Stat. and Prob., volume 1, pp. 547-561, 1961.
- [4] K. Kastella, *Joint multitarget probabilities for detection and tracking*, SPIE Proceedings, Acquisition, Tracking and Pointing XI, 21-25 April, 1997, Orlando, FL.
- [5] A. O. Hero III, B. Ma, O. J.J. Michel, and J. Gorman, *Applications of Entropic Spanning Graphs*, IEEE Signal Processing Magazine, September 2002, pp. 85-95.
- [6] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*, IEEE Transactions on Signal Processing, February 2002.
- [7] A. Doucet, N. de Freitas, and Neil Gordon, *Sequential Monte Carlo Methods in Practice*, Springer Publishing, New York, 2001.
- [8] J. Liu, and R. Chen, *Sequential Monte Carlo Methods for Dynamic Systems*, Journal of the American Statistical Association, September 1998.
- [9] M. Orton and W. Fitzgerald, *A Bayesian Approach to Tracking Multiple Targets Using Sensor Arrays and Particle Filters*, IEEE Transactions on Signal Processing, vol. 50, no.2, February 2002, pp. 216-223.
- [10] K. Kastella, *Discrimination gain to optimize detection and classification*, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 27, pp. 122-116, January 1997.
- [11] C. Hue, J-P. Le Cadre, P. Perez, *Tracking Multiple Objects with Particle Filtering*, IEEE Transactions on Aerospace and Electronic Systems, Vol 38. No. 3, pp. 791-802, July 2002.
- [12] F. Zhao, J. Shin, and J. Reich, *Information-Driven Dynamic Sensor Collaboration*, IEEE Signal Processing Magazine, March 2002, pp. 61-72.