An Information Based Approach to Decentralized Multi-platform Sensor Management

Christopher M. Kreucher[a], Keith D. Kastella[a], John W. Wegrzyn[a], and Brent L. Rickenbach[b]

[a]The General Dynamics Michigan Research and Development Center, Ypsilanti MI, USA
[b]General Dynamics Advanced Information Systems, Bloomington MN, USA

ABSTRACT

This paper describes a decentralized low communication approach to multi-platform sensor management. The method is based on a physicomimetic relaxation to a joint information theoretic optimization, which inherits the benefits of information theoretic scheduling while maintaining tractability. The method uses only limited message passing, only neighboring nodes communicate, and each node makes its own sensor management decisions.

We show by simulation that the method allows a network of sensor nodes to automatically self organize and perform a global task. In the model problem, a group of unmanned aerial vehicles (UAVs) hover above a ground surveillance region. An initially unknown number of moving ground targets inhabit the region. Each UAV is capable of making noisy measurements of the patch of ground directly below, which provide evidence as to the presence or absence of targets in that sub-region. The goal of the network is to determine the number of targets and their individual states (positions and velocities) in the entire surveillance region through repeated interrogation by the individual nodes. As the individual nodes can only see a small portion of the ground, they must move in a manner that is both responsive to measurements and coordinated with other nodes.

Keywords: distributed multi-sensor management, information theory, self organizing coherent systems

1. INTRODUCTION

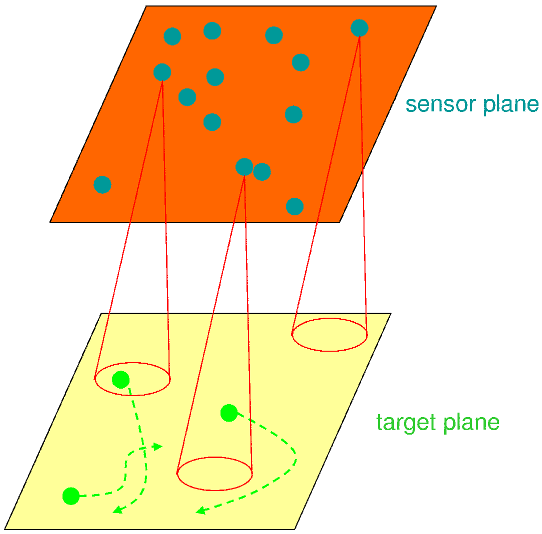
This paper presents a decentralized approach to multi-platform sensor management. Sensor management, as defined here, is the process whereby an agile sensor node autonomously determines what action it should take. The goal of a sensor management algorithm in a decentralized multi-platform environment is to control the behavior of the individual sensor nodes so that the network as a whole satisfies some global objective. The method we present is specifically designed for large networks of sensor nodes that require limited communication and a decentralized approach. As such, it relies only on communication between neighboring nodes and when communication is done, only a selected subset of a node's measurements are shared.

Our method is based on a physicomimetic relaxation of a constrained joint information theoretic optimization. The method builds on earlier work[1–3] which has shown that information theoretic sensor scheduling provides significant gains over other methods. The principle behind information theoretic methods is that nodes should take actions that are expected to gain information, where information may be measured by the Kullback-Leibler divergence,[1] relative entropy[3] or the Rényi Divergence.[2]

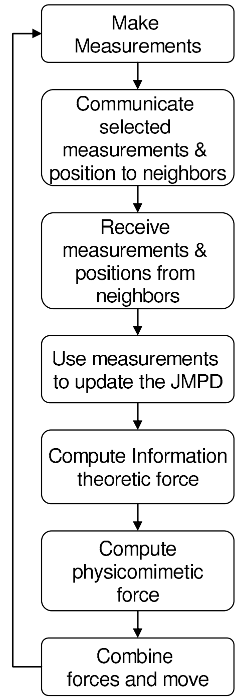
We illustrate our method by simulation on the following model problem. A collection of unmanned aerial vehicles (UAVs) hover above a ground surveillance region. There are an unknown number of moving ground targets in the region. Each UAV is capable of making measurements of the patch of ground directly below, and these measurements provide evidence as to the presence or absence of targets in that sub-region. The goal of the network is to determine the number of targets and their individual states (positions and velocities) through repeated interrogation of the ground. As individual nodes can only see a small portion of the ground, they need to move in response to the measurements to both completely survey the entire region and ensure that detected targets are accurately tracked. Thus, in this setting each node uses a sensor management algorithm to chose how to move. Fig. 1(a) provides a visual depiction of this model problem.

Send correspondence to Christopher.Kreucher@gd-ais.com

Defense Transformation and Network-Centric Systems, edited by Raja Suresh,
Proc. of SPIE Vol. 6249, 62490H, (2006) · 0277-786X/06/$15 · doi: 10.1117/12.665604



(a) The network is to determine the number and kinematic states of a group of moving ground targets. Each node stares directly down making measurements of the surveillance region. The sensor management algorithm described here provides a distributed, decentralized, low communication method for controlling the motion of nodes over time.



(b) Each node in the network repeatedly follows the procedure of generating measurements, transmitting them to neighbors, receiving measurements, updating its probability density, and finally computing the information theoretic and physicomimetic forces to decide where to move next.

Figure 1. The model problem. L: A visual depiction of the scenario. R: An outline of the algorithm used by each sensor.

Our application has several unique characteristics. First, the action space is continuous, which implies that simply enumerating all possible future actions and choosing the best is infeasible. Second, we want to jointly optimize the actions of N sensors which is a formidable computational task. Third, we have a constraint that sensors must not collide. Fourth, the method must be distributable and use limited communication. For these reasons, our approach is a "vector field"[4] method that addresses the constrained joint optimization problem by combining an information theoretic force with a collaboration force based on physicomimetics. This method allows tractability and low communication while still maintaining the benefits of information theoretic optimization.

The algorithm executed by each node in the network is illustrated in Fig. 1(b). As the figure shows, nodes communicate with their neighbors only. The combination of measurements made by a node and measurements received from others is used to update a probability density which describes the surveillance area. This probability density, combined with knowledge of nearby node positions, is used to select what actions the nodes should take next.

The paper proceeds as follows. In Section 2, we show how the joint multitarget probability density (JMPD) captures uncertainty about the surveillance region, and give a brief description of how it is estimated. Second, in Section 3, we show how the JMPD is used to compute the expected utility (as measured by information gain) of each possible future positioning of a node. The information theoretic method directly extends to the multisensor case. However, it becomes impractical computationally for more than a few nodes. Therefore, in Section 4, we give a method of approximate multisensor control based on physicomimetics. This method approximates the exponentially difficult problem of choosing the best combination of sensing actions by one that uses greedy (single-sensor) optimal actions combined with a secondary physicomimetic force that encourages cooperation. Section 5 gives simulation results for the model problem that illustrate the efficacy of the method. Finally, in Section 6, we give some remarks on the application of this method in a network centric warfare environment.

2. THE JMPD FOR CAPTURING NODE UNCERTAINTY

Each node in the network estimates the state of the surveillance region conditioned on the measurements it has access to. The estimate captures the uncertainty the node has about the contents of the surveillance region. As we will see in Section 3, we use reduction in uncertainty as a method for selecting future sensing actions.

Uncertainty comes in many forms. For example, a node may be uncertain about a portion of the region because it has never received measurements pertaining to that region. This is uncertainty in number of targets as well as the states of the targets (if there are any). Alternatively, a node may have surveyed a portion of the region well and be confident that there are some number of targets, but since targets are moving there is uncertainty about their kinematic state. Finally, the number and location of targets may be well known but other parameters – such as velocity, acceleration, identification, or mode – may by uncertain.

Estimation of the joint multitarget probability density (JMPD) is a Bayesian method of fusing models of target behavior, sensor capability, and actual measurements into a single picture which captures these and other uncertainties. Others have studied related methods based on Bayesian reasoning, e.g., the work in.[5–8]

Each node recursively estimates the JMPD as a means of describing the uncertainty about the surveillance region. If there were infinite communication ability, each node would have the same estimate and hence have the same JMPD. However, in our approach, each node only has access to measurements it has made and those it has received from neighbors, meaning that in general each node will have a different estimate of the JMPD.

2.1. The JMPD

The joint multitarget probability density (JMPD)

$$p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k|\mathbf{Z}^k) = p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k|T^k, \mathbf{Z}^k)p(T^k|\mathbf{Z}^k) \quad , \tag{1}$$

is the probability density for exactly T targets with states $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_{T-1}, \mathbf{x}_T$ at time k based on a set of past observations $\mathbf{Z}^k$. We abuse terminology by calling $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k|\mathbf{Z}^k)$ a density since T^k is a discrete valued random variable. In fact, the JMPD is a continuous discrete hybrid as it is a product of the probability mass function $p(T^k|\mathbf{Z}^k)$ and the probability density function $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k|T^k, \mathbf{Z}^k)$.

The number of targets at time k, T^k, is a variable estimated simultaneously with the states of the T^k targets. The JMPD is defined for all T^k, $T^k = 0 \cdots \infty$. The observation set $\mathbf{Z}^k$ refers to the collection of all measurements that the node has had access to over time, 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 from time i. Each of the $\mathbf{z}^i$ may be a different size depending on the number of measurements a node has received from others at time i.

Each $\mathbf{x}_t$ in $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k|\mathbf{Z}^k)$ is a vector quantity and may (for example) be of the form $[x, \dot{x}, y, \dot{y}]$. For convenience, the JMPD will be written more compactly in the traditional manner as $p(\mathbf{X}^k, T^k|\mathbf{Z}^k)$, which implies that the state-vector $\mathbf{X}^k$ represents a variable number of targets each possessing their own state vector. We will omit the time subscript k when convenient and no confusion will arise, and e.g., write simply $p(\mathbf{X}, T|\mathbf{Z})$.

The likelihood $p(\mathbf{z}|\mathbf{X}, T)$ and the JMPD $p(\mathbf{X}, T|\mathbf{Z})$ are conventional Bayesian objects manipulated by the usual rules of probability and statistics. Thus, a multitarget system has state $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_T)$ with probability distribution $p(\mathbf{x}_1, \cdots, \mathbf{x}_T, T|\mathbf{Z})$. This can be viewed as a hybrid stochastic system where the discrete random variable T governs the dimensionality of $\mathbf{X}$. Therefore the normalization condition the JMPD must satisfy is

$$\sum_{T=0}^{\infty} \int d\mathbf{X} p(\mathbf{X}, T|\mathbf{Z}) = 1 \quad , \tag{2}$$

where the single integral sign is used to denote the T integrations required.

The temporal update of the posterior likelihood proceeds according to the usual rules of Bayesian filtering. The model of how the JMPD evolves over time is given by $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$ and will be referred to as the kinematic prior (KP). The kinematic prior includes models of target motion, target birth and death, and any additional prior information that may exist such as terrain and roadway maps. In the case where target identification is part of the state being estimated, different kinematic models may be used for different target types. The time-updated (prediction) density is computed via the *model update* equation as

$$p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}) = \sum_{T^{k-1}=0}^{\infty} \int_{\mathbf{X}} d\mathbf{X}^{k-1} p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1}) p(\mathbf{X}^{k-1}, T^{k-1}|\mathbf{Z}^{k-1}) \tag{3}$$

The *measurement update* uses Bayes' rule to update the posterior density with a new measurement $\mathbf{z}^k$ as

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

2.1.1. Kinematic Modeling

The Bayesian framework outlined above requires a model of how the surveillance region evolves with time, $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$. This includes both how the number of targets changes with time (i.e. T^k versus T^{k-1}) and how individual targets that persist over time evolve (i.e. $\mathbf{x}^k$ versus $\mathbf{x}^{k-1}$). This model is chosen using the physics of the particular system under consideration. The simulation studies used herein come from a set of real ground targets recorded during a military exercise. Therefore, we specialize the models to this application.

We first define a set of spatially varying priors on target arrival and death. Let $\alpha^k(\mathbf{x})$ denote the *a priori* probability that a target will arrive at location $\mathbf{x}$ at time k. Similarly, let the *a priori* probability that a target in location $\mathbf{x}$ will die be denoted by $\beta^k(\mathbf{x})$. For the simulation studies, we assume temporally and spatially invariant (i.e. constant) rates for the birth and death and specify only α and β. The more general case is straightforward to implement, however it significantly complicates the notation. For more detail see.[9]

For persistent targets, we model the target motion as linear and independent for each target, i.e. if the state of a target is given by $\mathbf{x} = (x, \dot{x}, y, \dot{y})$ the model is

$$\mathbf{x}_i^k = \mathbf{F}\mathbf{x}_i^{k-1} + \mathbf{w}_i^k \quad , \tag{5}$$

where $\mathbf{w}_i^k$ is 0-mean Gaussian noise with covariance $\mathbf{Q}$, and

$$\mathbf{F} = \begin{pmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{Q} = q \begin{pmatrix} \tau^3/3 & \tau^2/2 & 0 & 0 \\ \tau^2/2 & \tau & 0 & 0 \\ 0 & 0 & \tau^3/3 & \tau^2/2 \\ 0 & 0 & \tau^2/2 & \tau \end{pmatrix} \quad . \tag{6}$$

We emphasize here that Linear/Gaussian models are not a requirement of the formulation, but are used as they have been found to perform well in simulation studies with the real data. More complicated models of target motion can be inserted where appropriate without directly effecting computations in the algorithm.

2.1.2. Sensor Modeling

The Bayesian framework further requires a model of how measurements are related to the state being tracked. The model problem considered in Section 5 consists of N sensors that make detections on a pixelated grid for the purposes of detecting and tracking a group of moving targets.

The measurement process is idealized as follows. The surveillance region is broken into $N_x \times N_y$ contiguous pixels. The x- and y- ground-plane projection of each pixel is Δ_x and Δ_y. The sensor response within pixel i is uniform for targets in pixel i and vanishes for targets outside pixel i. Define the occupation number $n_i(\mathbf{X})$ for pixel i as the number of targets in $\mathbf{X}$ that lie in i. The single target signal-noise-ratio (SNR), assumed constant across all targets, is denoted λ. We assume that when multiple targets lie within the same pixel their amplitudes add non-coherently (this will be accurate for unresolved optical targets and radar targets not moving as a rigid body). Then the effective SNR when there are n targets in a pixel is $\lambda_n = n\lambda$ and we use $p_n(z_i)$ to denote the pixel measurement distribution (note that the background distribution is obtained by setting $n = 0$).

A node hovers above the surveillance region and measures a set of P ($<< N_x \times N_y$) pixels directly below. Furthermore, some measurements may be received from neighboring sensors. We assume homogenous sensors and identical SNR targets for the purposes of this paper, but the method easily generalizes. Hence a measurement $\mathbf{z}$ consists of the pixel output vector $\mathbf{z} = [z_1, \ldots, z_M]$, where z_i is the output of pixel i and M is the total number of measurements available to the sensor (consisting of its own measurements and those received from neighbors).

We model both measurements of spatially separated pixels and multiple measurements of the same pixel at as conditionally independent given the state, i.e.,

$$p(\mathbf{z}|\mathbf{X}, T) = \prod_i p(z_i|\mathbf{X}, T) \ . \tag{7}$$

Let $\chi_i(\mathbf{x}_t)$ denote the indicator function for pixel i, defined as $\chi_i(\mathbf{x}_t) = 1$ when a (single) target in state $\mathbf{x}_t$ projects into sensor pixel i (i.e., couples to pixel i) and $\chi_i(\mathbf{x}_t) = 0$ when the target does not project into sensor pixel i. Observe a pixel can couple to multiple targets and single target can contribute to the output of multiple pixels, say, by coupling through side-lobe responses. The indicator function for the joint multitarget state is constructed as the logical disjunction

$$\chi_i(\mathbf{X}, T) = \bigvee_{t=1}^{T} \chi_i(\mathbf{x}_t) \ . \tag{8}$$

The set of pixels that couple to $\mathbf{X}$ is $i_{\mathbf{X}} = \{i|\chi_i(\mathbf{X}, T) = 1\}$. For the pixels that do not couple to $\mathbf{X}$, the measurements are characterized by the background distribution, $p_0(z_i)$. With this, (7) can be written as

$$p(\mathbf{z}|\mathbf{X}, T) \propto \prod_{i \in i_{\mathbf{X}}} \frac{p(z_i|\mathbf{X}, T)}{p_0(z_i)} \ . \tag{9}$$

With these modeling assumptions, the measurement distribution in pixel i depends only on its occupation number and (9) becomes

$$p(\mathbf{z}|\mathbf{X}, T) \propto \prod_{i \in i_{\mathbf{X}}} \frac{p_{n_i(\mathbf{X}), T}(z_i)}{p_0(z_i)} \ . \tag{10}$$

To complete the specification of the sensor model, we must give its dependence on SNR. Many models are plausible, depending on the detailed nature of the sensor physics. Here we have elected to use Rayleigh-distributed measurements. This distribution corresponds to envelope detected signals under a complex Gaussian radar return model, and has been used to model interfering targets in a monopulse radar system[10, 11] and to model clutter and target returns in turbulent environments. Rayleigh models are also often used for diffuse fading channels. We assume thresholded measurements and use a constant false-alarm rate (CFAR) model for the sensor. If the background false alarm rate is set at P_f, then the detection probability when there are n targets in a pixel is

$$P_{d,n} = P_f^{\frac{1}{1+n\lambda}} \ . \tag{11}$$

2.2. Estimation of the JMPD

Even for modest problems, the sample space of the JMPD is enormous as it contains all possible configurations of state vectors $\mathbf{x}_t$ for all possible values of T. For example, if the state of an individual target is given by the 4-tuple $\left(x, \dot{x}, y, \dot{y}\right)$, the sample space of the JMPD contains vectors of length $4N$ for all positive finite N. Thus, to estimate the JMPD in a computationally tractable manner, a sophisticated approximation method is required.

We use a particle filter to represent the JMPD, i.e.,

$$p(\mathbf{X}, T|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \tag{12}$$

where

$$\delta(\mathbf{X}-\mathbf{X}_p) = \begin{cases} 0 & T \neq T_p \\ \delta_D(\mathbf{X}-\mathbf{X}_p) & \text{otherwise} \end{cases} \quad . \tag{13}$$

and approximately implement the time and measurement update equations (3) and (4).

In earlier works,[9,12] we describe the novel importance density design that makes this tractable. The importance density is designed to recognize when it is permissible to factor the JMPD, and also addresses the issues of new targets arriving and exiting targets. For the purposes of this paper, we assume the JMPD has been efficiently estimated and is available for use by the sensor management algorithm.

3. INFORMATION THEORETIC SCHEDULING FOR A SINGLE SENSOR

In this section, we give an overview of information theoretic sensor scheduling in the single sensor setting. This method is extended to the multisensor case in the Section 4.

In our method of sensor management, actions are ranked based on the amount of information expected to be gained from their execution. In our model problem, nodes are mobile and so the management problem becomes one of choosing the next position for the node. In principle, we can compute the expected gain in information between the current JMPD and the JMPD that would result after moving to the new position $\mathbf{r} = (x, y, z)$ meters and making a measurement, for all $\mathbf{r}$. Then the sensor management decision is to select the best $\mathbf{r}$ based on expected information gain. Since in our application, the action $\mathbf{r}$ is to be chosen from the continuum, simple enumeration of possible actions is infeasible. Therefore, we instead sample a small number of possible $\mathbf{r}$ and use this sampling to derive a force to move the sensor. The rest of this section lays out the details of this approach.

3.1. The Rényi Divergence

The calculation of information gain between two densities p_1 and p_0 is done using the Rényi information divergence,[13] also known as the α-divergence:

$$D_\alpha(p_1||p_0) = \frac{1}{\alpha-1} \ln \int p_1^\alpha(x) p_0^{1-\alpha}(x) dx \tag{14}$$

The α parameter adjusts how heavily one emphasizes the tails of the two distributions p_1 and p_0. In the limiting case of $\alpha \to 1$ the Rényi divergence becomes the commonly utilized Kullback-Leibler (KL) discrimination

$$\lim_{\alpha\to 1} D_\alpha(p_1||p_0) = \int p_0(x) \ln \frac{p_0(x)}{p_1(x)} dx \quad . \tag{15}$$

If $\alpha = 0.5$, the Rényi information divergence becomes the Hellinger affinity $2 \ln \int \sqrt{p_1(x)p_0(x)}dx$, which is related to the Hellinger-Battacharya distance squared[13] via

$$D_{Hellinger}(p_1||p_0) = 2\Big(1 - \exp\Big(.5 D_{\frac{1}{2}}(p_1||p_0)\Big)\Big) \quad . \tag{16}$$

3.2. Rényi Divergence Between the Prior and Posterior JMPD

The function D_α in (14) is a measure of the divergence between the densities p_0 and p_1. In our application, we compute the divergence between the predicted density $p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1})$ and the updated density after a measurement $\mathbf{z}$ is made at new location $\mathbf{r}$, denoted $p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}, \mathbf{z}, \mathbf{r})$. Therefore, the relevant divergence is

$$D_\alpha\Big(p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}, \mathbf{r})||p(\cdot|\mathbf{Z}^{k-1})\Big) = \frac{1}{\alpha-1} \ln \sum_{T^k} \int p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}, \mathbf{z}, \mathbf{r})^\alpha p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1})^{1-\alpha} d\mathbf{X}^k \quad , \tag{17}$$

where the integral is interpreted as in (2). Using Bayes' formula applied to the JMPD (4) we obtain

$$D_\alpha\Big(p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}, \mathbf{r})||p(\cdot|\mathbf{Z}^{k-1})\Big) = \frac{1}{\alpha-1} \ln \frac{1}{p(\mathbf{z}|\mathbf{Z}^{k-1}, \mathbf{r})^\alpha} \sum_{T^k} \int p(\mathbf{z}|\mathbf{X}^k, T^k, \mathbf{r})^\alpha p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}) d\mathbf{X}^k \quad , \tag{18}$$

which shows that the ingredients to computing the divergence are the prior JMPD, the measurement likelihood $p(\mathbf{z}|\mathbf{X}^k, T^k, \mathbf{r})$ and the received measurements $\mathbf{z}$.

3.3. The Expected Rényi Divergence for a Sensing Action

To determine the best action, we must predict the value *before actually receiving* the measurement $\mathbf{z}$. Therefore, we calculate the expected value of the divergence for each possible action and use this to select the next action. The expectation may be written as an integral over all possible outcomes $\mathbf{z}$ when taking action $\mathbf{r}$ as

$$\mathbb{E}\left(D_\alpha\left(p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}, \mathbf{r})||p(\cdot|\mathbf{Z}^{k-1}) \right) \right) = \int d\mathbf{z} p(\mathbf{z}|\mathbf{Z}^{k-1}, \mathbf{r}) D_\alpha\left(p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}, \mathbf{r})||p(\cdot|\mathbf{Z}^{k-1}) \right) \ . \tag{19}$$

And then the method of scheduling is to choose

$$\hat{\mathbf{r}} = \arg\max_{\mathbf{r}} \mathbb{E}\left(D_\alpha\left(p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}, \mathbf{r})||p(\cdot|\mathbf{Z}^{k-1}) \right) \right) \ . \tag{20}$$

In practice, certain $\mathbf{r}$ are infeasible. There are *kinematic constraints* of the sensor which make certain locations unreachable in a single time step, including maximum sensor velocity and maximum sensor acceleration. Also there are *physical constraints* which prevent certain motions, including the topology of the surveillance region (i.e., a sensor should not collide with anything). Therefore, we actually need the constrained optimization

$$\hat{\mathbf{r}} = \arg\max_{\mathbf{r}\in\mathbb{C}} \mathbb{E}\left(D_\alpha\left(p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}, \mathbf{r})||p(\cdot|\mathbf{Z}^{k-1}) \right) \right) \ , \tag{21}$$

where $\mathbb{C}$ is the set of actions that meet both the kinematic and physical constraints. These constraints are handled in practice by simply removing those actions that violate the constraints from consideration.

3.4. Computational Method

When there are only a small number of actions to choose from, application of this method is straightforward. For each possible action, we compute the expected gain in information as given by (19). This computation is $O(MN_{parts})$ where M is the number of actions and N_{parts} is the fidelity of the particle filter approximation to the JMPD (note that when $\mathbf{z}$ is continuous advanced numerical techniques are required to evaluate the integral).

However, when the number of actions is continuous, as in the case we study here, simple enumeration is not feasible. In this situation, we use ideas from earlier works that employ "virtual force" or "potential field" methods.[4, 14, 15] In the field approach, one computes a force that compels a sensor to move. In our method, we derive a force that compels a sensor to move in a manner that maximizes information gain.

This is done as follows. First, we compute the information gain field at a small number of discrete locations using (19). We then use this scalar field to derive a force by taking its gradient, i.e.,

$$F_I(\mathbf{r}) = -\beta \nabla_{\mathbf{r}} \langle D_\alpha \rangle_{\mathbf{r}} \ , \tag{22}$$

where β is a scaling constant. This force then drives the sensor motion (subject to the constraints listed above).

4. INFORMATION THEORETIC SCHEDULING FOR MULTIPLE SENSORS

In this section, we describe how the method of Section 3 is extended to the case of multiple collaborating sensors in a distributed environment.

4.1. Optimal Multisensor Information Theoretic Scheduling

By straightforward extension of the method given in Section 3, information theoretic scheduling for a collection of N sensors requires choosing the set of future locations of the N sensors $(\mathbf{r}_1, \cdots, \mathbf{r}_N)$ which satisfies

$$\begin{aligned}(\hat{\mathbf{r}}_1 \cdots \hat{\mathbf{r}}_N) &= \underset{(\mathbf{r}_1 \cdots \mathbf{r}_N) \in \mathbb{C}'}{\arg\max} \, \mathbb{E}\Big(D_\alpha \Big(p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}_1 \cdots \mathbf{z}_N, \mathbf{r}_1 \cdots \mathbf{r}_N) || p(\cdot|\mathbf{Z}^{k-1}) \Big) \Big) \\ &= \underset{(\mathbf{r}_1 \cdots \mathbf{r}_N) \in \mathbb{C}'}{\arg\max} \int \cdots \int d\mathbf{z}_1 \cdots d\mathbf{z}_N p(\mathbf{z}_1 \cdots \mathbf{z}_N | \mathbf{Z}^{k-1}, \mathbf{r}_1 \cdots \mathbf{r}_N) D_\alpha \Big(p(\cdot|\mathbf{Z}^{k-1}) || p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}_1 \cdots \mathbf{z}_N, \mathbf{r}_1 \cdots \mathbf{r}_N) \Big) \ .\end{aligned} \tag{23}$$

As in the single sensor case, this requires a constrained optimization. In the multisensor case $\mathbb{C}'$ includes both the original constraints of $\mathbb{C}$ and a new constraint that sensors do not collide with each other. That is

$$\mathbb{C}' = \mathbb{C} \cap \{ \|\mathbf{r}_i - \mathbf{r}_j\| > d \ \forall i, j \text{ where } i \neq j \ \} \ . \tag{24}$$

The joint optimization can be rewritten as a sum of single sensor optimizations plus a correction factor as

$$\underset{(\mathbf{r}_1 \cdots \mathbf{r}_N) \in \mathbb{C}'}{\arg\max} \left(\sum_{i=1}^{N} \int d\mathbf{z}_i D_\alpha \Big(p(\mathbf{X}^k, T^k | \mathbf{Z}^{k-1}) || p(\mathbf{X}^k, T^k | \mathbf{Z}^{k-1}, \mathbf{z}_i, \mathbf{r}_i) \Big) + \mathbb{E}\big[h(\mathbf{z}_1, \cdots, \mathbf{z}_N, \mathbf{r}_1 \cdots \mathbf{r}_N, \mathbf{Z}^{k-1}) \big] \right) \ . \tag{25}$$

The function h in this expression is an “information coupling” term which accounts for the fact (among other things) that the gain in information for two sensors taking the same action is not double the information gain for a single sensor taking the action. In the limiting case as $\alpha \to 1$, the correction term can be written explicitly and the simplification becomes

$$\begin{aligned}&\underset{(\mathbf{r}_1 \cdots \mathbf{r}_N) \in \mathbb{C}'}{\arg\max} \left(\sum_{i=1}^{N} \int d\mathbf{z}_i D_\alpha \Big(p(\mathbf{X}^k, T^k | \mathbf{Z}^{k-1}) || p(\mathbf{X}^k, T^k | \mathbf{Z}^{k-1}, \mathbf{z}_i, \mathbf{r}_i) \Big) + \right. \\ &\left. \qquad \mathbb{E}\Big(\ln \Big[\frac{p(\mathbf{z}_N | \mathbf{z}_{N-1} \cdots \mathbf{z}_1, \mathbf{r}_N \cdots \mathbf{r}_1, \mathbf{Z}^{k-1}) \cdots p(\mathbf{z}_2 | \mathbf{z}_1, \mathbf{r}_2, \mathbf{r}_1, \mathbf{Z}^{k-1}) p(\mathbf{z}_1 | \mathbf{r}_1, \mathbf{Z}^{k-1})}{p(\mathbf{z}_N | \mathbf{r}_N, \mathbf{Z}^{k-1}) p(\mathbf{z}_{N-1} | \mathbf{r}_{N-1}, \mathbf{Z}^{k-1}) \cdots p(\mathbf{z}_2 | \mathbf{r}_2, \mathbf{Z}^{k-1}) p(\mathbf{z}_1 | \mathbf{r}_1, \mathbf{Z}^{k-1})} \Big] \Big) \right) \ .\end{aligned} \tag{26}$$

The correction term is still $O(M^N)$ to compute, and therefore must be approximated.

4.2. Computational Method

The new constraint that sensors cannot collide deals with action sets and not simply with individual actions and so it cannot be handled by simply censoring actions that violate the constraint. Therefore, we address this constraint in the standard way by defining the Lagrangian

$$\begin{aligned}L(\mathbf{r}_1 \cdots \mathbf{r}_N) &= \mathbb{E}\Big(D_\alpha \Big(p(\cdot|\mathbf{Z}^{k-1}, \mathbf{z}_1 \cdots \mathbf{z}_N, \mathbf{r}_1 \cdots \mathbf{r}_N) || p(\cdot|\mathbf{Z}^{k-1}) \Big) \Big) + \lambda f(\mathbf{r}_1 \cdots \mathbf{r}_N) \\ &= \sum_{i=1}^{N} \int d\mathbf{z}_i D_\alpha \Big(p(\mathbf{X}^k, T^k | \mathbf{Z}^{k-1}) || p(\mathbf{X}^k, T^k | \mathbf{Z}^{k-1}, \mathbf{z}_i, \mathbf{r}_i) \Big) + \mathbb{E}\big[h(\mathbf{z}_1, \cdots, \mathbf{z}_N, \mathbf{r}_1 \cdots \mathbf{r}_N, \mathbf{Z}^{k-1}) \big] + \lambda f(\mathbf{r}_1 \cdots \mathbf{r}_N) \ ,\end{aligned} \tag{27}$$

where the function f is a term that penalizes action sets that move the sensors too close together. The joint optimization then becomes an unconstrained optimization

$$(\hat{\mathbf{r}}_1 \cdots \hat{\mathbf{r}}_N) = \underset{(\mathbf{r}_1 \cdots \mathbf{r}_N)}{\arg\max} \, L(\mathbf{r}_1 \cdots \mathbf{r}_N) \ . \tag{28}$$

In our method, we simultaneously approximate both the information coupling term involving the expectation of h and the collision prevention term f by introducing a a function which reduces the value of action sets that involve sensors moving close together. We have chosen to use a physicomimetic force to provide this approximation, although other similar approximations are also valid. Evaluating this force has a very small computational burden, and requires only that a node know the positions of its neighbors.

Since we remain in a continuous action space environment, we must cast this term via a vector force as well. We use a generalization of the Lennard-Jones potential that serves as a zeroth order model of the intermolecular forces of liquids. The Lennard-Jones force for a pair of nodes i, j separated by a distance $r_{i,j}$ is radial with magnitude

$$F_{LJ}(r_{i,j}) = -\epsilon \left[m \frac{\gamma^m}{r_{i,j}^{m+1}} - n \frac{\gamma^n}{r_{i,j}^{n+1}} \right] \quad . \tag{29}$$

For the standard Lennard-Jones potential $m = 12$ and $n = 6$, and is referred to as the 6-12 potential. Observe that this is strongly repulsive as the radius between sensors $r_{i,j}$ gets small. The terms γ and ϵ are chosen based on sensor kinematic properties. The total force node i feels is simply the vector sum of the forces from all other nodes. To compute the total force, a node need only know the positions of the other nodes; in fact, since the force falls off so rapidly those sensors that are much more distant that γ have negligible effect on the computation. Therefore, for practical purposes, a node only needs to know the positions of nearby neighbors.

Denote by $\mathbf{F}_{LJ}^{i,j}(\mathbf{r}_i)$ the vector force node i feels from node j when positioned at $\mathbf{r}_i$. Then the total force node i feels from all other nodes when positioned at $\mathbf{r}_i$ is simply $\mathbf{F}_{LJ}^{i}(\mathbf{r}_i) = \sum_{j \neq i} \mathbf{F}_{LJ}^{i,j}(\mathbf{r}_i)$. This specification of relaxation term results in the final optimization

$$(\hat{\mathbf{r}}_1 \cdots \hat{\mathbf{r}}_N) = \arg\max_{(\mathbf{r}_1 \cdots \mathbf{r}_N)} \sum_{i=1}^{N} \left(\int d\mathbf{z}_i D_\alpha \Big(p(\mathbf{X}^k, T^k | \mathbf{Z}^{k-1}) || p(\mathbf{X}^k, T^k | \mathbf{Z}^{k-1}, \mathbf{z}_i, \mathbf{r}_i) \Big) + \lambda \mathbf{F}_{LJ}^{i}(\mathbf{r}_i) \right) \quad . \tag{30}$$

4.3. Distributed Implementation

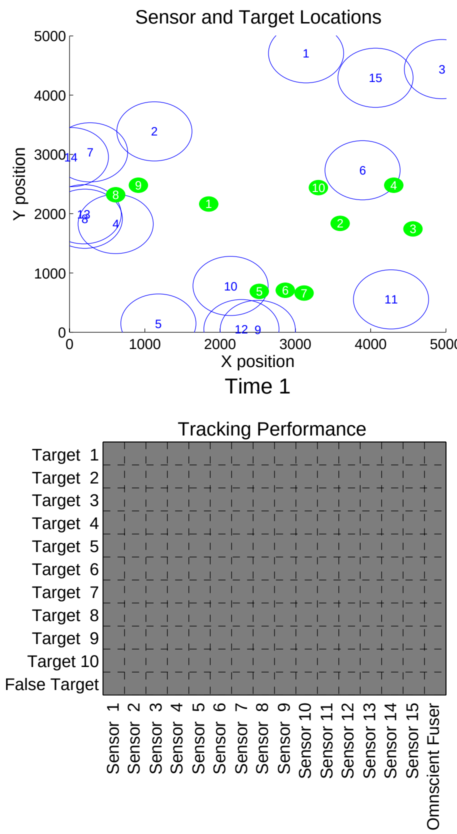
Notice that the method (30) allows each sensor to compute its next action in a completely distributed manner. By maximizing the term for each i, the sum is maximized. The first portion of the term in simply requires the exected information gain computed at each node without regard to the actions of other nodes. The second portion of the term requires only that each node know of the position of the nearby nodes.

There are many reasonable ways a node may decide what measurements should be transmitted to its neighbors. In this work, we employ a method where a node sends measurements based on the likelihood that they originate from a target. This information is directly calculable from the JMPD by integration over the measurement cell. Furthermore, when a node transmits measurements, it also shares its position so that the physicomimetic force may be computed by its neighbor. Our simulation studies assume a "radius of communication" which defines how far the distance that any transmission may be heard from a node.

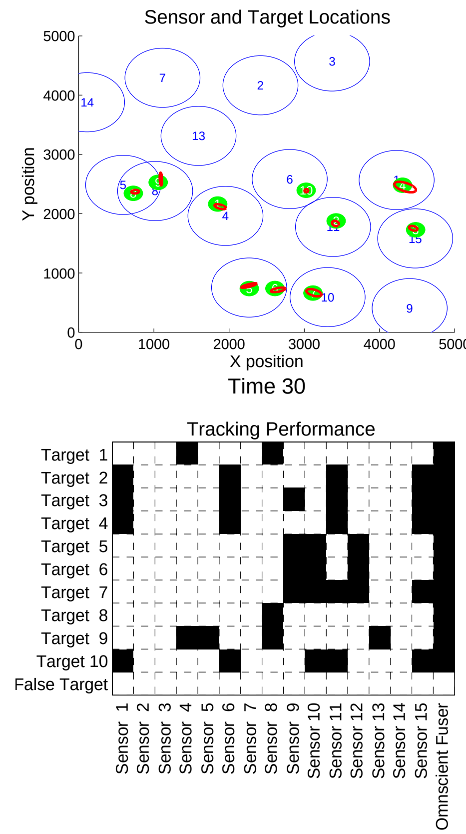
5. SIMULATION RESULT

In this section, we present simulation results to demonstrate the utility of the method. In the simulation, there are 15 nodes that hover above a surveillance region and make measurements as to the presence or absence of targets in the region as depicted in Fig. 1a. At initialization, none of the nodes has any information about the surveillance region, i.e., there is complete uncertainty as to the number of targets and the states of each. Furthermore, at initialization the nodes are randomly distributed above the surveillance region. An example of a set of initial positions is given in Fig. 2a.

Over time, the nodes execute the algorithm outlined in Section 4, by following the 7-step procedure in Fig. 1b. The resulting behavior of the nodes is one where they first collectively survey the entire surveillance region. Some nodes detect targets (or hear about them from neighbors) resulting in certain areas drawing the attention of nodes disproportionately due to the information theoretic force. However, the nodes as a collection still spread out through the entire area due to the physicomimetic force. After the transient time, all of the targets are detected and they are all monitored by some nodes in the region. Other nodes that are not actively involved in monitoring targets continue to sweep the region looking for new arrivals. The snapshot at time 30, which is shown in Fig. 2b, illustrates steady-state behavior.



(a) Top: Node locations at $t = 0$. The fifteen nodes hovering above the region are randomly placed. Each node measures only the ground directly below as indicated by the circle. The true location of the ten numbered ground targets is shown with a solid circle. Bottom: Each node in the network will know about a subset of the targets as it acquires or receives measurements over time. At initialization none of the targets are known to any node, as illustrated by the empty matrix.



(b) Top: Node locations at $t = 30$. The fifteen nodes hovering above the region have organized to survey the entire region and follow the detected targets. The true location of the ten numbered ground targets is shown with a solid circle, and the filter estimate given by the ellipse. Bottom: Each node in the network knows about a subset of targets as it has acquired or received measurements from only a portion of the region. This matrix indicates which sensors know about which targets.

Figure 2. Two snapshots of the system. Left: At initialization. Right: After 30 time steps

Extensive Monte Carlo tests (Fig. 3) where the radius of communication is varied from $r = 0m$ (i.e., no communication) to $r = 10000m$ (i.e., infinite communication radius) show the utility of the method. The quality of a nodes understanding is judged by two measures: (a) the number of true targets (of 10) the node knows about, and (b) the number of false targets that the node believes exist (ideally 0). Since we simulate a low communication distributed system, in general each node knows only about a subset of the targets in the area. This phenomenon is illustrated in Fig. 3 (left), which shows that for low communication radii (e.g., $r = 0 \cdots 1000m$) each node only knows about on the order of 2 targets. However, as illustrated in Fig. 3 (right), the network as a whole (as judged by a hypothetical omniscient fuser that gets to see all measurements made by all nodes) is able to correctly track all of the targets with a very low false target rate. Therefore, the network itself is performing the desired task – monitoring the entire surveillance region – by distributing work between the 15 nodes. The communication burden is very low. For $r = 1000m$, on average there is 1 measurement transmission for every 100 made. For $r = 2500m$, on average there is 1 measurement transmission for every 5 made.

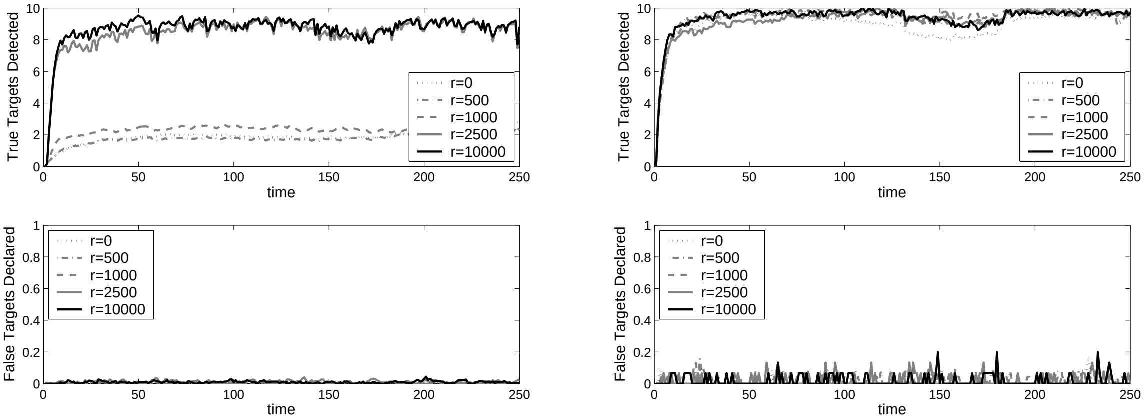


Figure 3. Performance of the algorithm as a function of communications radius (in meters). Left: The number of true targets correctly detected (of 10) and false targets incorrectly declared (ideally 0) versus time at *each node*. Right: The same statistics at a *hypothetical* omniscient fuser, which represents the performance of the network as a whole.

6. NETWORK CENTRIC WARFARE (NCW)

One possible strategy for taking this approach to the NCW battlefield would be to use a service-oriented architecture (SOA). A SOA is a loosely coupled, distributed, composable, O/S & platform independent collection of components. Components can be composed at runtime to meet real-time objectives. The name service-oriented comes from the shift of platform based computing where platforms independently solve tasks to a service-based approach where platforms advertise services that they can perform. In a SOA, service providers and service consumers have no *a priori* knowledge of each other; instead, providers and consumers use a broker service to facilitate finding other services.

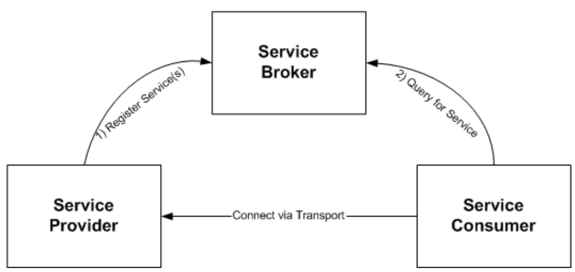


Figure 4. Simplified SOA Broker Interaction. 1) Provider registers its capabilities, transport information, and other metadata with the broker. 2) A consumer queries the broker for services of interest. 3) Using the query results, the consumer contacts the provider directly.

Figure 4 depicts a SOA. This simplified example shows a service provider registering its capabilities, metadata, and connection information with a broker. Any component may query the broker and learn of providers that match the query. Embedded in the service description is the transport or contact information that allows the consumer to communicate directly with the provider of interest after it has been discovered via the broker query.

A first step would be to advertise the homogenous sensor collection as a single service. This would allow consumers to subscribe to track information produced by the sensor network via a gateway node. The gateway could collect and report on the targets as well as sensor status (e.g. fuel level, payload configuration, position) and relay the information to consumers on the network that have subscribed to the sensor gateway service. One additional enhancement to this gateway approach would be to allow a remote operator to make adjustments to the algorithm changing the sensing area or other algorithm parameters, such as, the collision protection zone.

We have successfully demonstrated this approach in our lab showing the sensor field cross-cueing a simulated effects platform in a machine-to machine time critical target scenario.

A second step would be to use the SOA to aid in the introduction of heterogeneous sensors into the decentralized multi-platform sensor management system. This approach would require each sensor to register with a broker (which could be distributed across the network) and describe its capabilities. Nodes could query the broker for neighbors and obtain their position, target lists, and perhaps a history of places that their neighbors have visited. Heterogeneous nodes could collaborate using a standard interface for sharing target data and node status. This would allow nodes that can only do target detection to upgrade a targets information by collaborating with a platform that might provide target classification or identification. A tactical, peer-to-peer SOA would need to be used to accomplish this second step with a focus on the limited bandwidth of wireless networks.

7. CONCLUSION

This paper has presented a decentralized, low communication sensor management algorithm which effectively distributes the work load of a network among a large number of sensor nodes. The method is based a physicomimetic relaxation of the joint information theoretic optimization, which provides tractability and inherits the benefits of information theory. Via simulation we have shown that the network is capable of performing a global task (detect and track all targets in the surveillance region) via only neighbor to neighbor message passing.

REFERENCES

1. F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine* **19**, pp. 61–72, March 2002.
2. C. Kreucher, K. Kastella, and A. Hero, "Sensor management using an active sensing approach," *Signal Processing* **85**, pp. 607–624, March 2005.
3. K. J. Hintz and E. S. McVey, "Multi-process constrained estimation," *IEEE Transactions on Man, Systems, and Cybernetics* **21**, pp. 434–442, January/February 1991.
4. J. Borenstein, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics* (5), pp. 1179–1187, 1989.
5. L. D. Stone, T. L. Corwin, and C. A. Barlow, *Bayesian Multiple Target Tracking*, Artech House, 1999.
6. M. I. Miller, A. Srivastava, and U. Grenander, "Conditional mean estimation via jump-diffusion processes in multiple target tracking/recognition," *IEEE Trans. on Signal Processing* **43**(11), pp. 2678–2690, 1995.
7. R. E. Bethel and G. J. Paras, "A PDF multisensor multitarget tracker," *IEEE Transactions on Aerospace and Electronic Systems* **34**, pp. 153–168, January 1998.
8. I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, Kluwer Academic Publishers, 1997.
9. C. Kreucher, M. Morelande, K. Kastella, and A. Hero, "Particle filtering for multitarget detection and tracking (invited paper)," in *The Proceedings of The Twenty Sixth Annual IEEE Aerospace Conference*, March 5 - 12 2005.
10. Y. Bar-Shalom and W. D. Blair, *Multitarget-Multisensor Tracking: Applications and Advances, Volume III*, Artech House, 2000.
11. B. E. Tullsson, "Monopulse tracking of rayleigh targets: A simple approach," *IEEE Transactions on Aerospace and Electronic Systems* **27**(3), pp. 520–531, 1991.
12. C. Kreucher, K. Kastella, and A. Hero, "Multitarget tracking using the joint multitarget probability density," *IEEE Transactions on Aerospace and Electronic Systems* **41**, pp. 1396–1414, October 2005.
13. A. O. Hero III, B. Ma, O. Michel, and J. D. Gorman, "Alpha divergence for classification, indexing and retrieval," *Technical Report 328, Comm. and Sig. Proc. Lab., Dept. EECS, Univ. Michigan* , 2001.
14. W. Spears, R. Heil, D. Spears, and D. Zarzhitsky, "Physicomimetics for mobile robot formations," *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-04)* **3**, pp. 1528–1529, 2004.
15. C. Kreucher, K. Kastella, and A. Hero, "Multiplatform information-based sensor management," in *The Proceedings of the SPIE Defense Transformation and Network-Centric Systems Symposium*, **5820**, pp. 141–151, March 28 - April 1 2005.