

# An Information-Based Approach to Sensor Management in Large Dynamic Networks

*The problem addressed here is surveillance over a large area using remotely-controlled sensors that are managed so that the sensor nodes work together to meet multiple overall goals.*

By CHRISTOPHER M. KREUCHER, ALFRED O. HERO, III, *Fellow IEEE*,  
KEITH D. KASTELLA, AND MARK R. MORELANDE

**ABSTRACT** | This paper addresses the problem of sensor management for a large network of agile sensors. Sensor management, as defined here, is the process of dynamically retasking agile sensors in response to an evolving environment. Sensors may be agile in a variety of ways, e.g., the ability to reposition, point an antenna, choose sensing mode, or waveform. The goal of sensor management in a large network is to choose actions for individual sensors dynamically so as to maximize overall network utility. Sensor management in the multiplatform setting is a challenging problem for several reasons. First, the state space required to characterize an environment is typically of very high dimension and poorly represented by a parametric form. Second, the network must simultaneously address a number of competing goals. Third, the number of potential taskings grows exponentially with the number of sensors. Finally, in low-communication environments, decentralized methods are required. The approach we present in this paper addresses these challenges through a novel combination of particle filtering for nonparametric density estimation, information theory for comparing actions, and physicomimetics for computational tractability. The efficacy of the method is illustrated in a realistic surveillance

application by simulation, where an unknown number of ground targets are detected and tracked by a network of mobile sensors.

**KEYWORDS** | Information theory; joint multitarget probability density; multiplatform sensor management; multitarget tracking; particle filtering

## I. INTRODUCTION

Large networks of inexpensive sensors provide a means for performing persistent and ubiquitous surveillance over a wide region. Such networks have found use in diverse areas including habitat monitoring, the biomedical arena, industrial robotics, and defense. In this paper, we address the problem of managing the resources of a network consisting of a large number (tens to thousands) of agile sensors. Agility, as defined here, refers to any property of a sensor that can be dynamically tasked so that the network of sensors will be better able to perform surveillance on a region. In the general case, each sensor in the network is capable of a variety of actions, including where to move, which direction to emit energy, what mode to use, what waveform to transmit (if active), or which direction to listen (if passive). The goal of network sensor management is to develop a methodology where each node in the network adjusts its behavior dynamically so that the overall utility of the network is maximized.

Sensor management in large networks is challenging for a host of reasons. First, the state space required to characterize the region under surveillance is typically of extremely high dimension and is poorly represented by a

Manuscript received October 1, 2006; revised January 12, 2007.

**C. M. Kreucher** and **K. D. Kastella** are with General Dynamics Advanced Information Systems, Ypsilanti, MI 48197 USA (e-mail: Christopher.Kreucher@gd-ais.com; Keith.Kastella@gd-ais.com).

**A. O. Hero, III** is with the University of Michigan, Department of Electrical Engineering and Computer Science, Ann Arbor, MI 48109 USA (e-mail: hero@umich.edu).

**M. R. Morelande** is with the University of Melbourne, Department of Electrical and Electronic Engineering, Parkville VIC 3010, Australia (e-mail: m.morelande@ee.unimelb.edu.au).

Digital Object Identifier: 10.1109/JPROC.2007.893247

parametric form (e.g., a Gaussian or a sum of Gaussians). It is this state space that the network of nodes is to estimate, so proper mathematical formulation and efficient algorithmic implementation is key. Second, the sensor network must simultaneously address many competing goals (e.g., detection of new areas of interest and monitoring known areas of interest), so the scheduling metric must be suitably chosen to appropriately balance between these goals. Third, exact maximization of overall network utility is intractable as the number of actions available to the network at each decision epoch is exponential in the number of nodes and the number of actions each node can take. Therefore, a principled approximation to simultaneous multiplatform scheduling must be employed. This method must be robust and, while not solving the joint optimization problem exactly, encourage collaboration between sensor nodes in the manner that joint optimization would if it were practical to implement. Fourth, there must be information sharing between the individual sensor nodes (or the nodes and a central controller) so that the sensing workload is appropriately divided up amongst the collection of sensors. Information collected by the individual nodes must be fused (either centrally or at each node individually) to yield a single picture that characterizes the knowledge of the system under surveillance. This fused picture must then drive the actions of the sensors at the next decision epoch.

In this paper, we describe a method of scheduling the nodes in a large agile network that addresses each of the challenges outlined above. This method is a novel combination of adaptive importance density particle filtering for nonparametric density estimation, information theoretic measures for estimating the value of possible future actions, and physicomimetics for providing a tractable approximation to the joint optimization. An outline of the paper is as follows.

First, in Section II, we describe a mathematical formulation called the joint multitarget probability density (JMPD). This work has been reported previously [1], [2] and is reviewed briefly here as required background material for the following sections. The JMPD is used to capture the estimate of the state of the surveillance area, and is constructed on-line using statistical models of how the surveillance area evolves coupled with statistical models of how sensors work and actual measurements. This method is related to the approach of others, including Stone [3], Srivastava and Miller [4], and others [5]–[7] as discussed in [1] and elsewhere. Our model problem consists of a surveillance area encompassing a number of moving ground targets. The number of targets and their positions, velocities, and classes are unknown at startup and (potentially) time varying from then on. The JMPD is a single probabilistic entity that simultaneously describes uncertainty about the number of targets, as well as the positions, velocities, and identifications of those targets.

The JMPD is estimated on-line using a novel multitarget particle filtering [8] technique, which relies on an importance density specifically designed for this problem. Others have used particle filtering approaches for multitarget filtering, including Orton [9], Maskell [10], and others [11]–[14].

Second, in Section III, a method of using an information theoretic measure called the Rényi Divergence for sensor management is discussed. Portions of this work have been reported previously in [15]. The repetition here is minimal and serves to establish the required background and notation for the following sections. Specifically, the quality of a proposed action (be it moving the sensor to another location, or pointing an antenna in a particular direction) is measured by the amount of information that is expected to be gained by its execution. This approach is related to that of others, including Zhao [16], Hintz [17], Schmaedeke [18], and others [19]–[21] as discussed in [15] and elsewhere. At each epoch when a decision is to be made, the uncertainty about the surveillance region (as captured by the JMPD) is used to compute the value of each of the possible sensing actions using an information theoretic measure called the Rényi (alpha-) Divergence. Information theoretic metrics have the compelling property that different types of information (e.g., information about the presence or absence of targets, the position, velocity, and identification of targets) can all be compared on an equal footing. For example, by using an information-based approach, the value of an action that extracts information about the class of a firm target can be compared directly to the value of an action that is meant to search for new targets. We restrict our attention in this paper to single-stage (myopic) scheduling. Of course, the most general dynamic sensor scheduling problem is a partially observed stochastic control problem over a finite or infinite horizon. Such problems are formulated in terms of the information state and therefore exactly solving the resulting dynamic programming problem is computationally intractable in most cases [22]. Multistage extensions to our approach using a partially observable Markov decision process (POMDP) [23], [24] formulation and approximation techniques have been discussed elsewhere [25], [26]. Others have used POMDP approaches with other metrics and approximation methods for related problems, e.g., [27]–[30].

The method of multiplatform information-based sensor management that is the central contribution of this paper is given in Section IV. It is shown therein that the multiplatform optimization can be written as a sum of single platform optimizations and a correction term. The correction term can be explicitly written for a limiting case of the Rényi Divergence, but it can be qualitatively described in the general case. A physicomimetic term is used to approximate the correction term and properly enforce collaboration and cooperation between the large

number of sensor nodes. Physicomimetics (or “artificial physics”) [31] refers to a class of approximation methods motivated by natural physical forces, e.g., the intermolecular forces of liquids. Due to the exponential explosion in the number of possible actions the network can take at any decision epoch, it is impractical to enumerate all possible combinations of sensing actions for the nodes in the network and choose the best. The physicomimetic approach is a tractable and robust approximation that allows each sensor to be scheduled locally while providing an impetus for working together with the other sensors. While this does not precisely get at the globally optimum sensor management solution, it provides a tractable approximation with robust performance. Others have used physicomimetic approaches for multiplatform scheduling [32], but to our knowledge this is the first time this approach has been combined with information theory, and more importantly the first time this approach has been directly related to a constrained joint information theoretic optimization.

Fourth, we show that by having each sensor compute a local estimate of the JMPD, the method can be employed in a decentralized setting. Therefore, it is possible to implement this method with no centralized controller, where each sensor is responsible for making its own sensor management decisions. When bandwidth is limited, only a subset of measurements may be shared among sensors, leaving each local estimate of the JMPD sub-optimal. However, it is shown by simulation that adequate performance is still achieved as each sensor has a very good local estimate of the JMPD.

Finally, we give a series of simulations in Section V that show the performance of the method in detecting and tracking an unknown number of moving ground targets in a model problem. We consider large-scale problems involving tens to hundreds of platforms cooperating together to perform surveillance on a large region. The simulations illustrate several key features of the approach.

- (a) The Rényi Divergence metric combined with the JMPD estimate of uncertainty allows platforms to trade between the competing goals of detection and tracking, resulting in a system that performs well under both criteria.
- (b) As the amount of communication available in the system changes, different behavior patterns emerge from the collection of platforms—although the platforms are always controlled by maximization of information flow through the network.
- (c) The combination of a physicomimetic force and a (single-platform) information seeking force, as dictated by the joint information theoretic optimization, properly balances the exploitation and exploration goals in a manner that the individual forces themselves cannot.

## II. THE JOINT MULTITARGET PROBABILITY DENSITY (JMPD)

This section describes the JMPD and its particle filter (PF) implementation. Our model problem focuses on using a network of sensors to provide surveillance on a region to determine the number and states of a collection of moving targets in the region. The JMPD is a single probabilistic entity that captures all of the uncertainty about a surveillance region. This includes uncertainty about the number of targets present in the region, as well as the kinematic state, class, and mode of each. The JMPD is computed recursively by fusing measurements, target models, sensor models, and ancillary information such as roadway and terrain elevation maps. This nonlinear filtering approach is appropriate rather than other methods such as the MHT [33] because it captures all uncertainty (i.e., uncertainty about target number, kinematic state, and class) under one framework, and linear/Gaussian assumptions typically do not apply in this setting. As will be discussed in Sections III and IV, the method of multiplatform sensor management advocated here uses reduction in uncertainty as measured by the JMPD to drive future sensing actions. A high level overview of this process is illustrated in Fig. 1.

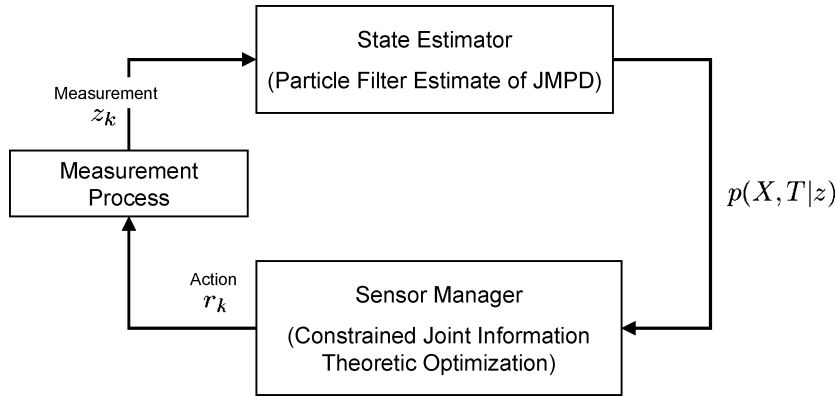
The material in this section is largely drawn from a series of previously published papers [1], [2], [34], [35]. More detail on the formulation and implementation can be found therein. The summary discussion here is provided as background material necessary before introducing the main topic of this paper, multiplatform sensor resource allocation via maximizing information flow. As discussed in [1] and elsewhere, the JMPD approach presented is related to the approach of others, e.g., [5]–[7], [36].

### A. Formulation of the JMPD

Recursive estimation of the JMPD provides a means for simultaneously estimating the number of targets and their kinematic states by fusing models and measurements. The joint multitarget conditional probability density

$$p(x_k^1, x_k^2, \dots, x_k^{T-1}, x_k^T, T_k | z_{0:k}) \\ = p(x_k^1, x_k^2, \dots, x_k^{T-1}, x_k^T | T_k, z_{0:k}) p(T_k | z_{0:k}) \quad (1)$$

is the probability density for exactly  $T$  targets with states  $x^1, x^2, \dots, x^{T-1}, x^T$  at time  $k$  based on a set of past observations  $z_{0:k}$ . We abuse terminology by calling the JMPD  $p(x_k^1, x_k^2, \dots, x_k^{T-1}, x_k^T, T_k | z_{0:k})$  a density since  $T_k$  is a discrete valued random variable. In fact, as (1) shows, the JMPD is a continuous discrete hybrid as it is a product of the probability mass function  $p(T_k | z_{0:k})$  and the probability density function  $p(x_k^1, x_k^2, \dots, x_k^{T-1}, x_k^T | T_k, z_{0:k})$ .



**Fig. 1.** An illustration of the recursive state estimation and sensor management process described in this paper. In general, one performs state estimation to capture the uncertainty about the surveillance region. Here, the state of the surveillance region is captured by the joint multitarget probability density, which is approximated using a particular method of state estimation based on particle filtering. This estimate is passed to a sensor management algorithm which decides what action(s) to take next. Our method of sensor management is based on a constrained joint information theoretic optimization. This action is then executed, resulting in a measurement of the environment which is used to update the state estimate.

The observation set  $z_{0:k}$  refers to the collection of measurements up to and including time  $k$ , i.e.,  $z_{0:k} \doteq \{z_0, z_1, \dots, z_k\}$ , where each of the  $z_i$  may be a single measurement or a collection of measurements made at time  $i$  (e.g., a vector, matrix, or cube of measurements from a single sensor or a concatenation of measurements from multiple sensors made at the same time). We will refer to measurements made at a specific time  $i$  as  $z_i$ , all measurements made from time 0 to time  $k$  as  $z_{0:k}$ , and a generic measurement set (either a collection of measurements or a measurement at a single time) as simply  $z$ , which will be clear by context. Furthermore, in future sections we will also find it necessary to explicitly include the sensing action  $r$  (e.g., the choice of sensor mode or sensor movement) that resulted in the measurement  $z$ . In this case, the JMPD will be more explicitly written as  $p(x_k^1, x_k^2, \dots, x_k^{T-1}, x_k^T, T_k | z_{0:k}, r_{0:k})$  and measurement likelihood will be written as  $p(z_k | x_k^1, x_k^2, \dots, x_k^{T-1}, x_k^T, T_k, r_k)$ . For simplicity, this extended notation is suppressed in the present discussion.

Each of the single target states  $x^i$  in the density  $p(x_k^1, x_k^2, \dots, x_k^{T-1}, x_k^T | T_k, z_{0:k})$  is a vector quantity. We will typically use the 2-D target state idealization  $[x, \dot{x}, y, \dot{y}]$  when providing concrete examples in this paper, although the notation will be kept general until examples are presented. In other problems where the mode is to be estimated [37], we have used  $[x, \dot{x}, y, \dot{y}, m]$  and when the class is to be estimated [38] we have used  $[x, \dot{x}, y, \dot{y}, c]$ .

For convenience, the JMPD will be written more compactly in the traditional manner as  $p(X_k, T_k | z_{0:k})$ , which implies that the system state-vector  $X_k$  represents a collection of  $T_k$  targets each possessing their own state vector. This can be viewed as a hybrid stochastic system

where the discrete random variable  $T_k$  governs the dimensionality of  $X_k$ .

The number of targets at time  $k$ ,  $T_k$ , is a variable to be estimated simultaneously with the states of the  $T_k$  targets. The JMPD is defined for all  $T_k$ ,  $T_k = 0 \dots \infty$ . Therefore the normalization condition that the JMPD must satisfy is

$$\sum_{T=0}^{\infty} \int dx^1 \dots dx^T p(x^1, \dots, x^T, T | z) = 1 \quad (2)$$

where the single integral sign is used to denote the  $T$  integrations required (note that we have dropped the time subscripts here to lighten the notation). This can alternatively be written in the shorthand notation

$$\sum_{T=0}^{\infty} \int dX p(X, T | z) = 1 \quad (3)$$

where it is understood again that  $T$  determines the dimensionality of  $X$  and the single integral sign represents the  $T$  integrations required.

The likelihood  $p(z|X, T)$  and the joint multitarget probability density  $p(X, T|z)$  are conventional Bayesian objects manipulated by the usual rules of probability and statistics. Specifically, the temporal update of the posterior density proceeds according to the usual rules of Bayesian filtering. The model of how target state and target number evolve over time is given by  $p(X_k, T_k | X_{k-1}, T_{k-1})$  and will be referred to as the kinematic prior. The kinematic prior

includes models of target motion, target birth and death, and any additional prior information on kinematics 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

$$p(X_k, T_k | z_{0:k-1}) = \sum_{T_{k-1}=0}^{\infty} \int dX_{k-1} p(X_k, T_k | X_{k-1}, T_{k-1}) \times p(X_{k-1}, T_{k-1} | z_{0:k-1}). \quad (4)$$

Note that the formulation of the time evolution of the JMPD given in (4) makes several assumptions. First, as is commonly done, we assume that state evolution is Markovian. Furthermore, we assume the action at time  $k-1$  does not influence state evolution, i.e., if the sensing action performed at time  $k-1$  is denoted  $r_{k-1}$  then by assumption  $p(X_k, T_k | X_{k-1}, T_{k-1}, r_{k-1}) = p(X_k, T_k | X_{k-1}, T_{k-1})$ . In some situations this assumption is not valid, including the “smart” target problem [39]. If either of these assumptions is invalid in a particular setting, (4) would be generalized appropriately.

The *measurement update* equation uses Bayes’ rule to update the posterior density with a new measurement  $z_k$  as

$$p(X_k, T_k | z_{0:k}) = \frac{p(z_k | X_k, T_k) p(X_k, T_k | z_{0:k-1})}{p(z_k | z_{0:k-1})}. \quad (5)$$

## B. The Particle Filter Implementation of the JMPD

The sample space of the JMPD is very large since it contains all possible configurations of state vectors  $X_k = \{x_k^1, \dots, x_k^{T_k}\}$  for all possible values of  $T_k$ . Thus, for computational tractability, a sophisticated approximation method is required. This section briefly describes our particle filter implementation with special attention given to the adaptive importance density that allows tracability. Measurements of the computational complexity of this estimation algorithm versus number of targets in the surveillance area on standard equipment are given in [1].

1) *Notation*: In particle filtering, the probability density of interest (here the JMPD) is represented by a set of  $N$  weighted samples (particles). Since a particle is a sample from the PDF of interest, here a particle is more than just the estimate of the state of a target; it is an estimate of the state of the surveillance region. In particular, it incorporates both an estimate of the states

of all of the targets as well as an estimate of the number of targets.

The multitarget state vector for  $T$  targets is simply the concatenation of  $T$  single target state vectors (again here the time subscript is dropped for notational simplicity)

$$X = [x^1, x^2, \dots, x^{T-1}, x^T]. \quad (6)$$

A particle  $i$  is similarly expressed as a concatenation of  $T^{(i)}$  state estimates as

$$X^{(i)} = [x^{(i)(1)}, x^{(i)(2)}, \dots, x^{(i)(T^{(i)-1})}, x^{(i)(T^{(i)})}] \quad (7)$$

which says particle  $i$  estimates there are  $T^{(i)}$  targets, where  $T^{(i)}$  can be any nonnegative integer, and in general is different for different particles.

To formalize, let  $\delta_D$  denote the ordinary Dirac delta, and define a delta function between the  $T$ -target state vector  $X$  and the  $T^{(i)}$ -target state vector  $X^{(i)}$  as

$$\delta(X - X^{(i)}) = \begin{cases} 0 & T \neq T^{(i)} \\ \delta_D(X - X^{(i)}) & \text{otherwise.} \end{cases} \quad (8)$$

Then the particle filter approximation to the JMPD is given by a set of particles  $X^{(i)}$  and corresponding weights  $w^{(i)}$  as

$$p(X, T | z) \approx \sum_{i=1}^N w^{(i)} \delta(X - X^{(i)}) \quad (9)$$

where  $\sum_{i=1}^N w^{(i)} = 1$ .

The JMPD is defined for all possible numbers of targets,  $T = 0, 1, 2, \dots$ . As each of the particles is a sample drawn from the JMPD, a particle may estimate 0, 1, 2,  $\dots$  targets. Here, different particles in the approximation may correspond to different estimates of the number of targets.

2) *Multitarget SIR*: With these definitions, the traditional sampling importance resampling (SIR) particle filter extends directly to filtering with the JMPD. The method is to simply propose new particles at time  $k$  from the particles at time  $k-1$  by projecting through the kinematic prior. This kinematic model includes both the dynamics of persistent targets (e.g., a nearly constant velocity model) and the model of how targets enter and exit the



surveillance region (e.g., a spatially uniform birth/death process). Specifically, targets entering or leaving the surveillance region are accounted for as the proposed particle  $X_k^{(i)}$  may have either fewer targets or more targets than  $X_{k-1}^{(i)}$  (i.e.,  $T_k^{(i)} = T_{k-1}^{(i)} - 1$  or  $T_k^{(i)} = T_{k-1}^{(i)} + 1$ ). The weight update when particles are proposed in this manner is simply

$$w_k^{(i)} = w_{k-1}^{(i)} p(z_k | X_k^{(i)}). \quad (10)$$

3) *The Inefficiency of the SIR Method:* The SIR particle filter has the benefit of being simple to describe and easy to implement. However, SIR is too numerically inefficient for multitarget problems.

Assume for discussion that the sensor is pixelated, returning energy in one of  $C$  sensor cells. Target birth may occur in any unoccupied cell at any time step. Target death may occur in any occupied cell at any time step. One method of handling this would be to have a very large number of particles, capable of encoding all possibilities of the next state, i.e., no new target, one new target (in each of the possible unoccupied cells), two new targets (in each possible pair of unoccupied cells), etc., and likewise with target removal. This must still retain the particle diversity required for efficient filtering of the persistent targets. This method requires an enormous number of particles for successful approximation.

Furthermore, even with no birth and death, target proposals using kinematics are too inefficient for multitarget problems. Consider the simple case where there are  $T$  targets in the surveillance region, and this is known to the filter. In order for a particle to be a “good” estimate of the multitarget state, all  $T$  targets must be proposed to “good” locations. Without knowledge of measurements, the probability an individual target is proposed to a “good” location is  $\alpha < 1$ . Therefore, as the number of targets grows, the probability of a “good” multitarget proposal becomes significantly less than one (goes as  $\alpha^T$ ). Hence, the number of particles required to perform adequate tracking with high probability grows exponentially.

Both of these problems are remedied via an importance density that more closely approximates the optimal importance density (OID) [8]. The OID is the density that minimizes the conditional variance of particle weights by most efficiently combining current measurements, the kinematic prior, and the prior to direct proposals to higher likelihood multitarget states. This is in contrast to simply sampling from the kinematic prior, which as illustrated above is in general very inefficient. Typically, the OID is prohibitive to sample from exactly, but in some applications an approximate OID can be constructed, drawn from, and any mismatch corrected precisely in the weight

update. In the following subsections, we briefly summarize the method of approximately sampling from the OID developed for efficient estimation in this setting. Additional detail can be found in [1], [2].

4) *Importance Density Design for Target Birth/Death:* In order to reach the efficiency required for tractable detection and tracking of multiple targets, we advocate a measurement directed sampling scheme for target birth and death. As described in detail in [2], the key idea in the development of a tractable method to handle target birth and death is an existence grid. The existence grid contains the probability that a single target is in cell  $i$  at time  $k$  given the measurements made up to and including time  $k$ . Qualitatively, the existence grid describes those regions of the measurement space that deserve attention. The existence grid cells are initialized with a prior probability which may be spatially varying. The probability of target existence in each cell is propagated forward via a simple addition/removal model, and updated with new measurements according to Bayes’ rule. In our application, we have chosen to use a spatially and temporally constant arrival and removal rate. These simplifications make the existence grid computationally simple to maintain.

To handle target birth, new targets are preferentially proposed in locations according to the rate dictated by the existence grid. This bias is removed during the weight update process so that the Bayesian recursions are still exactly implemented. This implementational technique allows particles to be used more efficiently as new targets are only proposed in highly probable areas. Target death is handled analogously.

5) *Importance Density Design for Persistent Targets:* The kinematic prior does not take advantage of the fact that the JMPD state vector is made up of individual target state vectors. In particular, targets that are far apart in measurement space are uncoupled and should be treated as such. Furthermore, similar to that of the uninformed birth/death proposal, the current measurements are not used when proposing new particles. These two considerations taken together result in an inefficient use of particles and therefore require a large number of particles to successfully track. Empirical results illustrating this assertion are given in [1].

To overcome these deficiencies, we use a technique which biases proposals towards measurements and allows for factorization of the multitarget state when permissible. These strategies propose each target (or cluster of coupled targets, as will be clarified later) in a particle separately, and form new particles as the combination of the proposed clusters. We describe the use of two sample-based methods here, the independent partitions (IP) method of [9] and the coupled partitions (CP) method. The basic idea of both CP and IP is to construct particle proposals at the target

(or group-of-targets) level, incorporating measurements to bias proposals toward the optimal importance density. This bias is removed in the weight update stage, and therefore the Bayes recursions are still exactly implemented. We advocate an adaptive partition (AP) method which performs a clustering on targets and automatically switches between the two methods as appropriate. Finally, we mention an improved method of target (or group-of-targets) proposal that is based on directly sampling from the optimal importance density. This method is applicable in some situations (as discussed in [2]) and has been shown to increase algorithm efficiency significantly in those cases. All of the methods are performed only on the persistent targets, and the algorithm is done in conjunction with the addition and removal of targets as described in the preceding section.

*The IP Method:* The IP method given by Orton [9] is a convenient way to propose particles when part or all of the joint multitarget density factors. As employed here, the IP method proposes a new target as follows. For a target  $p$ , each particle at time  $k - 1$  has its  $p^{\text{th}}$  partition proposed via the kinematic prior and weighted by the measurements. From this set of  $N$  weighted estimates of the state of the  $p^{\text{th}}$  target, we select  $N$  samples with replacement to form the  $p^{\text{th}}$  partition of the particles at time  $k$ .

With well separated targets, this method allows many targets to be tracked with the same number of particles needed to track a single target (although the dimension of each particle is larger). Indeed, in the case of well separated targets, the multitarget tracking problem breaks down into many single-target problems. The IP method is useful for just this case, as it allows the targets to be treated independently when their relative spacing deems that appropriate. Note, however, that applying this method on a target by target basis is not appropriate when there is any measurement-to-target association ambiguity. Therefore, when targets are close together in sensor space, an alternative approach must be used.

*The CP Proposal Method:* When the posterior distributions on target position begin to overlap, we call the corresponding partitions coupled. In these instances, another method of particle proposal such as CP must be used. An alternative method would be to use the IP strategy on groups of partitions as alluded to in [9]. As discussed below, the CP method proposes multiple possible future realizations for each partition (as opposed to the IP method which proposes a single future realization for each particle). This additional sampling fidelity can be viewed as a better approximation to the optimal importance density than a method that simply proposes one possible realization for each particle. In practice, we find that the CP method provides a benefit by giving extra computation at those points where it is most necessary.

We apply the CP method as follows. To propose partitions  $p_1 \cdots p_M$  of particle  $i$ , CP proposes  $R$  possible realizations of the future state using the kinematic prior.

The  $R$  proposed futures are then given weights according to the current measurements and a single representative is selected. This process is repeated for each particle until the partitions for all particles has been formed. As in the IP method, the final particle weights are adjusted for this biased sampling.

*Adaptive Particle Proposal Method:* A more efficient method is to use a hybrid of the IP and CP method, called the AP method [1], [34]. The AP method again considers each target separately. Those targets sufficiently well separated from all other targets are treated as independent and proposed using the IP method. When targets are not sufficiently distant, the CP method is used on those groups (clusters) of targets that are coupled. To determine when targets are sufficiently separated, we use filter estimate of targets states and then perform a clustering procedure based on distance in sensor space between the estimated target states.

*An Improvement:* In certain circumstances, the optimal importance density can be more efficiently approximated than the sample based approach discussed here. In particular, if target dynamics are linear/Gaussian and measurements are made on a grid, the optimal proposal involves sampling from a mixture of truncated normals [2], [35]. In this case, a similar AP approach is used wherein partitions are first separated into groups that are uncoupled and then each group is treated by sampling from the optimal importance density. In the most general case, one does not have a convenient (semi-) closed form and instead relies on the purely sample driven methods of IP and CP as described above.

### III. INFORMATION THEORY FOR SINGLE-SENSOR MANAGEMENT

This section describes a method of sensor management based on maximizing information flow. We focus here on the single platform case and describe the multiplatform case in the following section. Sensor management, as defined here, refers to choosing the best action for an agile sensor to take. This may include where to point, what mode to use, or where to move. In this method of sensor management, actions are ranked based on the amount of information expected to be gained from their execution. In principle, this is accomplished by computing the expected gain in information between the current JMPD and the JMPD that would result after taking action  $r$  and making a measurement, for all feasible  $r$ . Then the sensor management decision is to select the best  $r$  using expected information gain as the metric. The method presented in this section is generic with respect to what  $r$  represents—i.e.,  $r$  may represent the choice of a waveform, the choice of a pointing direction, or the choice movement for the platform (or all three).

The material in the first half of this section is largely drawn from a series of previously published papers [15],

[38], [40]. It provides the background and notational conventions necessary before introducing the main topic of this paper, multiplatform sensor resource allocation via maximizing information flow. These references also include measurements of the computational complexity of the algorithms on standard equipment. The information-based approach presented here is related to the approach of others, e.g., [18], [19], [41] as discussed in [15] and elsewhere.

### A. The Rényi Divergence

In our approach, the calculation of information gain between two densities  $p_1$  and  $p_0$  is done using the Rényi information divergence [42], also known as the  $\alpha$ -divergence

$$D_\alpha(p_1||p_0) = \frac{1}{\alpha - 1} \ln \int p_1^\alpha(x) p_0^{1-\alpha}(x) dx. \quad (11)$$

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

$$\lim_{\alpha \rightarrow 1} D_\alpha(p_1||p_0) = \int p_0(x) \ln \frac{p_0(x)}{p_1(x)} dx. \quad (12)$$

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 [42] via

$$D_{\text{Hellinger}}(p_1||p_0) = 2 \left( 1 - \exp \left( .5 D_{\frac{1}{2}}(p_1||p_0) \right) \right). \quad (13)$$

### B. Rényi Divergence Between the Prior and Posterior JMPD

The function  $D_\alpha$  in (11) is a measure of the divergence between the densities  $p_0$  and  $p_1$ . In our application, we wish to compute the divergence between the prediction density  $p(X_k, T_k | z_{0:k-1}, r_{0:k-1})$  and the updated density after a measurement  $z_k$  when taking action  $r_k$ , denoted  $p(X_k, T_k | z_{0:k-1}, r_{0:k-1}, z_k, r_k)$ . Notice that we now include the action taken at time  $k$ ,  $r_k$ , and the history of actions  $r_{0:k-1}$  explicitly into the notation for clarity. This divergence measures the amount of information that the new measurement has provided and allows us to rank the utility of different actions according to the information

flow they produce. The relevant divergence for our setting is thus given by

$$\begin{aligned} D_\alpha(p(\cdot | z_{0:k-1}, r_{0:k-1}, z_k, r_k) || p(\cdot | z_{0:k-1}, r_{0:k-1})) \\ = \frac{1}{\alpha - 1} \ln \sum_{T_k} \int p^\alpha(X_k, T_k | z_{0:k-1}, r_{0:k-1}, z_k, r_k) \\ \times p^{1-\alpha}(X_k, T_k | z_{0:k-1}, r_{0:k-1}) dX_k \end{aligned} \quad (14)$$

where the integral is interpreted as in (3).

Using Bayes' formula applied to the JMPD (5), we obtain

$$\begin{aligned} D_\alpha(p(\cdot | z_{0:k-1}, r_{0:k-1}, z_k, r_k) || p(\cdot | z_{0:k-1}, r_{0:k-1})) \\ = \frac{1}{\alpha - 1} \ln \frac{1}{p^\alpha(z_k | z_{0:k-1}, r_{0:k-1}, r_k)} \\ \times \sum_{T_k} \int p^\alpha(z_k | X_k, T_k, r_k) p(X_k, T_k | z_{0:k-1}, r_{0:k-1}) dX_k \end{aligned} \quad (15)$$

which shows that the ingredients to computing the divergence are the prediction JMPD  $p(X_k, T_k | z_{0:k-1}, r_{0:k-1})$ , the measurement likelihood  $p(z_k | X_k, T_k, r_k)$  and the received measurements  $z_k$ .

### C. The Expected Rényi Divergence for a Sensing Action

To determine the best action to take next, we must in fact predict the value of taking action  $r_k$  *before actually receiving* the measurement  $z_k$ . 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  $z_k$  when taking action  $r_k$  as

$$\begin{aligned} E[D_\alpha(p(\cdot | z_{0:k-1}, r_{0:k-1}, z_k, r_k) || p(\cdot | z_{0:k-1}, r_{0:k-1})) \\ | z_{0:k-1}, r_{0:k-1}, r_k] \\ = \int dz_k p(z_k | z_{0:k-1}, r_{0:k-1}, r_k) \\ \times D_\alpha(p(\cdot | z_{0:k-1}, r_{0:k-1}, z_k, r_k) || p(\cdot | z_{0:k-1}, r_{0:k-1})). \end{aligned} \quad (16)$$

The expectation in (16) is across the measurement outcome  $z_k$  and is to be interpreted as a conditional expectation where the past sensor measurements  $z_{0:k-1}$ , past sensor actions  $r_{0:k-1}$ , and current sensing action  $r_k$  are known.

Then the method of scheduling we advocate is to choose the best action  $\hat{r}_k$  as the one that maximizes the



expected gain in information, i.e.,

$$\hat{r}_k = \arg \max_{r_k} \mathbb{E}[D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}, z_k, r_k)) \| p(\cdot|z_{0:k-1}, r_{0:k-1}) \| z_{0:k-1}, r_{0:k-1}, r_k]. \quad (17)$$

In practice, certain  $r_k$  are infeasible. There are *kinematic constraints* of the platform which make certain locations unreachable in a single time step, including maximum platform velocity and maximum platform 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

$$\hat{r}_k = \arg \max_{r_k \in \mathbb{C}} \mathbb{E}[D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}, z_k, r_k)) \| p(\cdot|z_{0:k-1}, r_{0:k-1}) \| z_{0:k-1}, r_{0:k-1}, r_k] \quad (18)$$

where  $\mathbb{C}$  is the set of actions that meet both the kinematic and physical constraints. For single-sensor scheduling, these constraints are handled in practice by simply removing those actions that violate the constraints from consideration.

#### D. Theoretical Motivation for the Information Gain Metric

Consider a situation where a target is to be detected, tracked, and identified using observations acquired sequentially according to a given sensor selection policy. In this situation one might look for a policy that is “universal” in the sense that the generated sensor sequence is optimal for all three tasks. A truly universal policy is not likely to exist since no single policy can be expected to simultaneously minimize target tracking MSE and target misclassification probability, for example. Remarkably, policies that optimize information gain are near universal: they perform nearly as well as task-specific optimal policies for a wide range of tasks. In this sense the information gain can be considered as a proxy for performance for any of these tasks. The fundamental role of information gain as a near universal proxy has been demonstrated both by simulation and by analysis in [38], [43]. The key result is a bound that shows any bounded risk function is sandwiched between two weighted alpha divergences. This inequality is a rigorous theoretical result that suggests that the expected information gain is a near universal proxy for arbitrary risk functions.

#### E. 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 (16). This computation is  $O(M)$  where  $M$  is the (small) number of (discrete) actions possible for the sensor to take. Of course, each of the  $M$  computations has complexity that scales with the number of particles used in the particle filter approximation to the JMPD ( $N$ ) and the number of targets predicted to be in the surveillance region ( $T$ ). Furthermore, when the measurement  $z$  is continuous (or multidimensional), advanced numerical techniques are required to evaluate the expectation.

However, when the action space is continuous, simple enumeration is not feasible. We now specialize to the case where the action  $r$  refers to a new positioning of the sensor (i.e., the platform is mobile and the sensor management problem is one of deciding where to move the platform) to illustrate control in the continuous action setting. The new position  $r$  of the sensor is in principle a 3-D vector from the continuum  $\mathbb{R}^3$  specifying the  $(x, y, z)$  coordinates of the next platform position. In this situation, we use ideas from earlier works that employ “virtual force” or “potential field” methods [32], [44], [45]. In the field approach, one computes a force that compels a sensor to move rather than explicitly calculating the value of all possible next positions and choosing the best.

In our method, the value of a potential next position is given by the expected information gain (16). Therefore, the force that drives platform action in the continuous action space case is the gradient of the information gain field at the current location, as given by

$$F_I(r_k) = -\beta \nabla_{r_k} \mathbb{E}[D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}, z_k, r_k)) \| p(\cdot|z_{0:k-1}, r_{0:k-1}) \| z_{0:k-1}, r_{0:k-1}, r_k] \quad (19)$$

where  $\beta$  is simply a scaling constant. This force then drives the sensor to move in the manner that maximally provides information flow (subject to the constraints discussed above).

## IV. MULTIPLATFORM INFORMATION BASED SENSOR MANAGEMENT

In this section, we present our method of information-based multiplatform sensor management. The method works by maximizing the expected information gain between the posterior JMPD and the JMPD after a new set of measurements are made by the  $P$  platforms. It builds on the ideas and notation developed in Section III for the single-sensor case but now has the additional constraints imposed by multiple sensors in a single surveillance area (i.e., the sensors should not collide and sensors should not

be redundantly tasked unless there is compelling reason to do so). Additionally, it is ultimately desired to employ the technique in a decentralized low-communication environment so the technique should lend itself to this setting. Some of this material has appeared in previous conference papers [45], [46]. As mentioned earlier, others have approached this problem from a similar viewpoint, e.g., [19], [47].

This section proceeds by first giving the formulation of optimal multisensor information theoretic scheduling assuming the scheduler is centralized. This is seen to be a joint constrained information theoretic optimization by natural extension of the ideas in Section III, but the constraint set has changed. Furthermore, the optimization is now seen to be combinatoric in nature (i.e., the joint action space grows exponentially with the number of sensors) so relaxation is required. We next show that the joint constrained information theoretic optimization can be written as a sum of single-sensor optimizations and a correction term. The correction term can be explicitly written in a limiting case of the Rényi Divergence. The correction term is then approximated to produce a tractable method computationally. Finally, if we allow each sensor to compute a local estimate of the JMPD and use limited message passing between neighboring sensors, we show the entire procedure can be done approximately in a decentralized manner.

### A. Optimal Multisensor Information Theoretic Scheduling

Information theoretic scheduling for a collection of  $P$  platforms requires choosing the set of  $P$  next-actions for the  $P$  platforms. The formulation for the multiple platform case can be given as a direct extension of the single-sensor case. First, let  $r_k^i$  and  $z_k^i$  denote the sensing action and measurement received, respectively, for the  $i^{\text{th}}$  sensor at time  $k$ . Next, let  $\vec{r}_k$  and  $\vec{z}_k$  denote the sensing actions and measurements for the  $P$  platforms at time  $k$ , respectively. That is, let

$$\vec{r}_k = [r_k^1, r_k^2, \dots, r_k^{P-1}, r_k^P] \quad (20)$$

and

$$\vec{z}_k = [z_k^1, z_k^2, \dots, z_k^{P-1}, z_k^P]. \quad (21)$$

Once again, we highlight that at this point, we do not make assumptions on what the actions  $\vec{r}_k$  are. An action may be, for example, the choice of sensor position, the choice of waveform to emit, the choice of pointing angle, and so on.

By direct extension of the single-sensor approach, the multisensor information theoretic scheduling method is to find the sensor actions  $\hat{r}_k$  that maximize the expected gain in information, i.e., choose  $\hat{r}_k$  such that

$$\begin{aligned} \hat{r}_k &= \arg \max_{\vec{r}_k \in \mathcal{C}'} \mathbb{E}[D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}, \vec{z}_k, \vec{r}_k) \\ &\quad \|p(\cdot|z_{0:k-1}, r_{0:k-1}))|z_{0:k-1}, r_{0:k-1}, \vec{r}_k] \\ &= \arg \max_{\vec{r}_k \in \mathcal{C}'} \int d\vec{z}_k p(\vec{z}_k|z_{0:k-1}, r_{0:k-1}, \vec{r}_k) \\ &\quad \times D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}) \| p(\cdot|z_{0:k-1}, r_{0:k-1}, \vec{z}_k, \vec{r}_k)) \end{aligned} \quad (22)$$

where the integral is to be interpreted as performing the  $P$  integrations required.

Analogously to (16), the expectation in (22) is taken over the measurement outcomes  $\vec{z}_k$  and is conditioned on knowing the past measurements  $z_{0:k-1}$ , the past actions  $r_{0:k-1}$ , and the current action set  $\vec{r}_k$ .

Note that direct computation of this quantity requires comparison of  $M^P$  possible sensing actions (in the case where there are  $M$  discrete actions for each of the  $P$  platforms). This is clearly not tractable for large  $P$ , and therefore approximate techniques are required.

Note further that this is also a constrained optimization. In the multisensor case, the constraint set  $\mathcal{C}'$  is expanded beyond the single-sensor constraint set to now include both the original constraints of  $\mathcal{C}$  and a new constraint that sensors do not collide with each other. That is

$$\mathcal{C}' = \mathcal{C} \cap \{\|r^i - r^j\| > d \forall i, j \text{ where } i \neq j\}. \quad (23)$$

### B. Connection to Single-Sensor Optimization

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

$$\begin{aligned} \arg \max_{\vec{r}_k \in \mathcal{C}'} \sum_{i=1}^P \mathbb{E}[D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}, z_k^i, r_k^i) \\ \|p(\cdot|z_{0:k-1}, r_{0:k-1}))|z_{0:k-1}, r_{0:k-1}, \vec{r}_k] \\ + \mathbb{E}[h(\vec{z}_k, \vec{r}_k, z_{0:k-1}, r_{0:k-1})|z_{0:k-1}, r_{0:k-1}, \vec{r}_k]. \end{aligned} \quad (24)$$

where the function  $h$  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  $\alpha \rightarrow 1$ , the correction term can be written explicitly and the simplification becomes (25) (see bottom of the page), i.e., the multisensor optimization can be written explicitly as a sum of single-sensor optimizations and a correction term which is simply the expected value of the log of the joint measurement likelihood over the product of the individual measurement likelihoods. The proof of this statement is given in the Appendix.

The correction term has this intuitive form related to mutual information when the KL divergence ( $\alpha \rightarrow 1$ ) is used. It reflects the utility that other sensor measurements provide in predicting a sensors measurement. In the limiting case of independence, this term vanishes.

The correction term is still  $O(M^P)$  to compute, where  $M$  is the number of potential actions each platform could take and  $P$  is the number of platforms, and therefore must be approximated. Note also, that it is this correction term that hinders distributed implementation.

### C. 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 by defining the Lagrangian

$$\begin{aligned} L(\vec{r}_k) &= \mathbf{E} [D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}, \vec{z}_k, \vec{r}_k)) \| p(\cdot|z_{0:k-1}, r_{0:k-1}) \\ &\quad |z_{0:k-1}, r_{0:k-1}, \vec{r}_k] + \lambda f(\vec{r}_k) \\ &= \sum_{i=1}^P \mathbf{E} [D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}, z^i, r^i) \\ &\quad \| p(\cdot|z_{0:k-1}, r_{0:k-1})) |z_{0:k-1}, r_{0:k-1}, \vec{r}_k] \\ &\quad + \mathbf{E} [h(\vec{z}_k, \vec{r}_k, z_{0:k-1}, r_{0:k-1})] + \lambda f(\vec{r}_k) \end{aligned} \quad (26)$$

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

$$\hat{\vec{r}}_k = \arg \max_{\vec{r}_k} L(\vec{r}_k). \quad (27)$$

This optimization can be looked at as a sum of three terms: a collection of single-sensor optimizations, an

information coupling (or correction) term, and a collision avoidance term.

At this point, we again specialize to the situation where the sensor control action  $r$  refers to choosing the next position for the sensor. With this assumption, we choose to simultaneously approximate both the information coupling term involving the expectation of  $h$  and the collision prevention term  $f$  by introducing a single function which reduces the value of action sets that involve sensors moving close together. We have chosen to use a physi-comimetic force [32] 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. Different approximation methods may be more appropriate in other settings. For example, in cases where teams of sensors must work to interrogate a single target one may use a second-order expansion of the information gain and a third-order correction term. If there are additional obstacles in the region (e.g., buildings or no-fly zones) the collision avoidance term would be suitably modified.

We provide an empirical comparison between the correction term (exactly computed at a small number of points) and the Lennard–Jones force used as the approximation for the correction term and the relaxation term on a model problem in Section V-A2.

Since we remain in a continuous action space environment, we must cast this approximation 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 [48]. The Lennard–Jones force for a pair of platforms  $i, j$  separated by a distance  $d_{i,j}$  is radial with magnitude

$$F_{LJ}(d_{i,j}) = -\epsilon \left[ m \frac{\gamma^m}{d_{i,j}^{m+1}} - n \frac{\gamma^n}{d_{i,j}^{n+1}} \right]. \quad (28)$$

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  $d_{i,j}$  gets small. The terms  $\gamma$  and  $\epsilon$  are chosen based on platform kinematic properties. The total force platform  $i$  feels is simply the vector sum of the forces

---


$$\begin{aligned} \arg \max_{\vec{r}_k \in \mathcal{C}'} \sum_{i=1}^P \mathbf{E} [D_\alpha(p(\cdot|z_{0:k-1}, r_{0:k-1}, z_k^i, r_k^i) \| p(\cdot|z_{0:k-1}, r_{0:k-1})) |z_{0:k-1}, r_{0:k-1}, \vec{r}_k] \\ + \mathbf{E} \left[ \ln \left( \frac{p(z_k^1, \dots, z_k^P | r_k^1, \dots, r_k^P, z_{0:k-1}, r_{0:k-1})}{p(z_k^1 | r_k^1, z_{0:k-1}, r_{0:k-1}) \cdots p(z_k^P | r_k^P, z_{0:k-1}, r_{0:k-1})} \right) |z_{0:k-1}, r_{0:k-1}, \vec{r}_k \right] \end{aligned} \quad (25)$$

from all other platforms. To compute the total force, a platform 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 than  $\gamma$  have negligible effect on the computation. Therefore, for practical purposes, a node only needs to know the positions of nearby neighbors.

Denote by  $\mathbf{F}_{LJ}^{i,j}(r^i)$  the vector force node  $i$  feels from node  $j$  when positioned at  $r^i$  [which is radial in direction with magnitude given by (28)]. Then the total force node  $i$  feels from all other nodes when positioned at  $r^i$  is simply  $\mathbf{F}_{LJ}^i(r^i) = \sum_{j \neq i} \mathbf{F}_{LJ}^{i,j}(r^i)$ . Using this approximation approach to the joint constrained information theoretic optimization of (22) results in the final approximate multiplatform optimization

$$\hat{r}_k = \arg \max_{\vec{r}_k} \sum_{i=1}^P \left\{ \mathbf{E} \left[ D_\alpha \left( p(\cdot | z_{0:k-1}, r_{0:k-1}, z_k^i, r_k^i) \right) \right] \right\} \cdot \left\{ \left\| p(\cdot | z_{0:k-1}, r_{0:k-1}, \vec{r}_k) \right\| + \lambda \mathbf{F}_{LJ}^i(r_k^i) \right\}. \quad (29)$$

This approximation can be viewed as driving sensors to compute greedy actions (i.e., ignoring the actions of other sensors) and correcting overzealous information seeking behavior by compelling sensors to stay away from others. These two forces are balanced through the choice of  $\lambda$ , which when properly chosen, allows sensors to come near when the situation warrants (i.e., in cases where the maximal joint utility is gained from close positioning of sensors), while staying apart in general.

#### D. Distributed Implementation

Notice that the method of (29) allows each sensor to compute its next action in a completely distributed manner, assuming each sensor has: 1) knowledge of the other sensors' positions and 2) knowledge of the JMPD (or alternatively has access to all measurements the network has made). The first portion of (29) simply requires the expected information gain computed at each node without regard to the actions of other nodes. The second portion requires only that each node know of the position of the nearby nodes.

We are further interested here in a low-communication version of this optimization. Therefore, only selected measurements may be transmitted by the network. What results in this case is that each sensor in the network has an approximate JMPD, computed only using locally made measurements and measurements shared by nearby neighbors. There are many reasonable ways a node may decide what measurements should be transmitted to its neighbors and many reasonable ways to define a neighborhood. 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 (locally estimated) JMPD by marginalization. Furthermore, when a node transmits measurements, it also must also share its position so that the physicomimetic force may be computed by its neighbor. Our simulation studies assume a "radius of communication" which defines the neighborhood of a sensor. It is assumed all sensors within the communication radius can hear the transmission, and no sensor outside can. This results in a nice practical situation where no static interconnection of nodes is required. If a node does not hear from another, it knows the other is outside of range and therefore should have no bearing on current decisions.

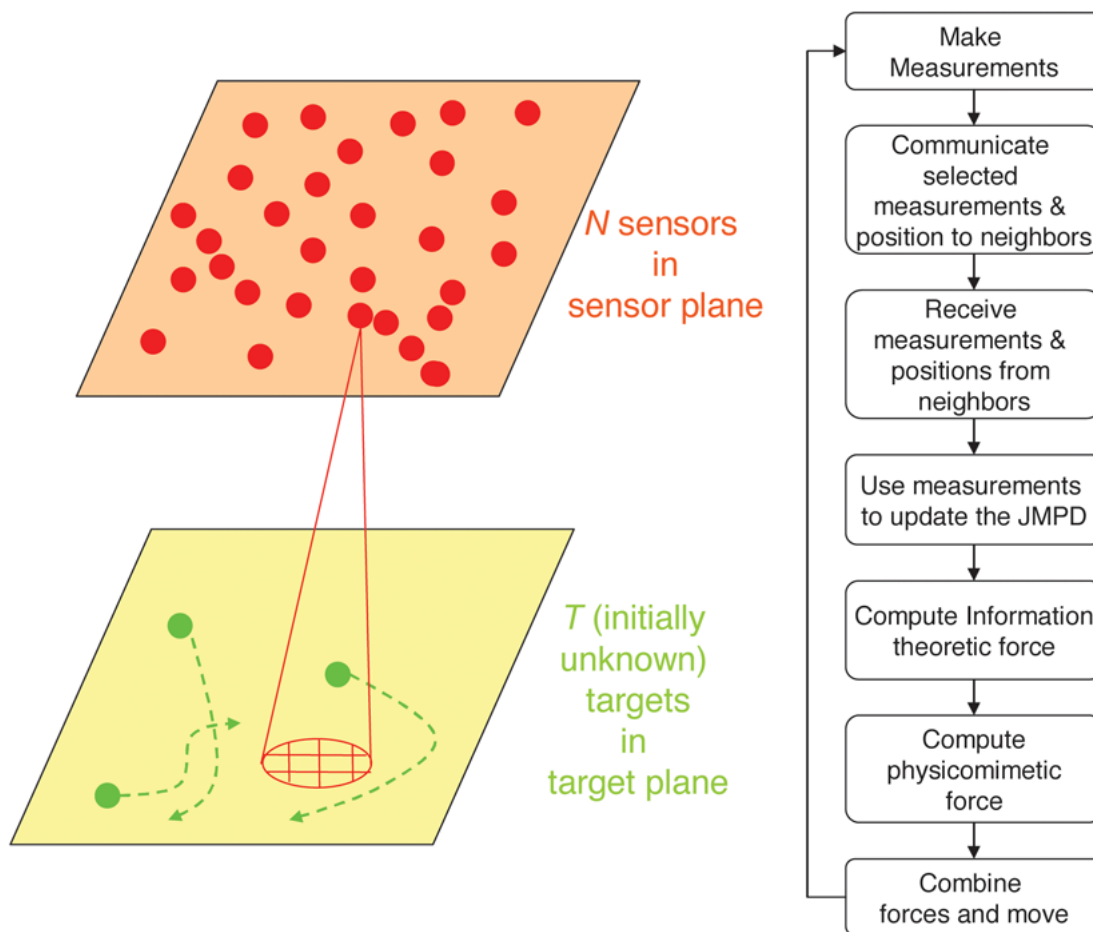
Therefore, in practice the distributed version of this optimization works as follows. Each sensor collects measurements at its current position. Selected measurements (based on the likelihood they originate from a target as determined by the local estimate of the JMPD) are broadcast along with an estimate of platform position. Those platforms within the communication radius receive this transmission, and likewise a platform receives the transmission from all other platforms for which it is in the communication radius. The locally made measurements and measurements received from neighbors are used to update the local JMPD as described in Section II. Each platform then computes the greedy (single-sensor) information-based utility for future positionings and corrects this impetus with the repulsive Lennard-Jones force. The platform then moves and the process starts anew.

## V. SIMULATION RESULTS

In this section, we present two simulation case studies that illustrate the efficacy of the sensor management method given in Section IV.

The first case study uses a small number (15) of very capable platforms to provide region surveillance. This simulation implements the decentralized version of the algorithm by: 1) estimating the (local) JMPD at each platform from local measurements and measurements received from neighbors (if any) and 2) computing platform movements by combining locally computed information theoretic forces with locally computed physicomimetic forces. The simulation analyzes performance in terms of detection and tracking capabilities as a function of communication radius.

The second case study focusses on a large number (as many as 500) of platforms with very limited sensing capabilities. For the purposes of simulation, the centralized version of the algorithm is used. Although simulation of the entire decentralized algorithm is near real-time on a per-platform basis (as would be required for implementation), simulation of 500 platforms requires significantly longer than real-time (500 times the single platform simulation time). The centralized algorithm is significantly cheaper computationally, owing to the fact that only one



**Fig. 2.** *Left: The model problem setup. 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 so as to best learn the contents of the surveillance region. Right: 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.*

JMPD must be estimated (rather than 500 separate JMPDs). The communication burden is significantly increased, however. This simulation illustrates surveillance performance in a similar model problem, and also compares the performance of the proposed algorithm with an algorithm that uses only the physicomimetic force and one that only uses the information gain force. It is shown that the proposed algorithm, which combines these two forces as motivated by the joint constrained information theoretic optimization approximation, significantly outperforms algorithms based on the constituent forces alone.

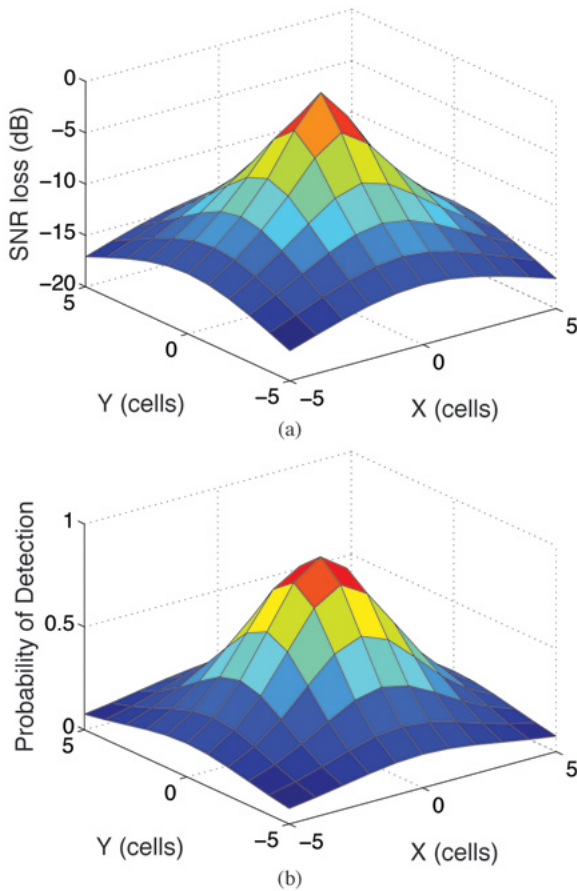
### A. A Simulation With a Small Number of Very Capable Platforms

1) *Description of the Model Problem:* The following simulation uses 15 platforms with decentralized control to

provide surveillance on a large region. The model problem uses a 5000 m × 5000 m surveillance area that contains ten moving ground targets (the number of targets and their positions and velocities are initially unknown). Each sensor has an imaging sensor with a wide field of view that provides evidence as to the presence or absence of targets in a subsection of the region at any time. The goal is for the network of sensors to collaborate together in a low-communication setting so that the number of targets and their individual states is learned as quickly and accurately as possible.

Target trajectories for the simulation come directly from a set of recorded data based on GPS measurements of vehicle positions over time collected as part of a battle training exercise at the Army’s National Training Center. Targets routinely come within sensor cell resolution (i.e., cross). Persistent targets are modeled in the JMPD time





**Fig. 3.** A description of the capability of the sensors used in this simulation. Each sensor has a footprint on the ground of radius 5 cells. The effective SNR is modeled to fall off as  $1/r^2$  from the field of view center. (a) SNR loss as a function of distance from the center of the field of view. (b)  $p_d$  as a function of distance from the center of the field of view for SNR = 16 dB with  $p_f = .01$  in the Rayleigh model.

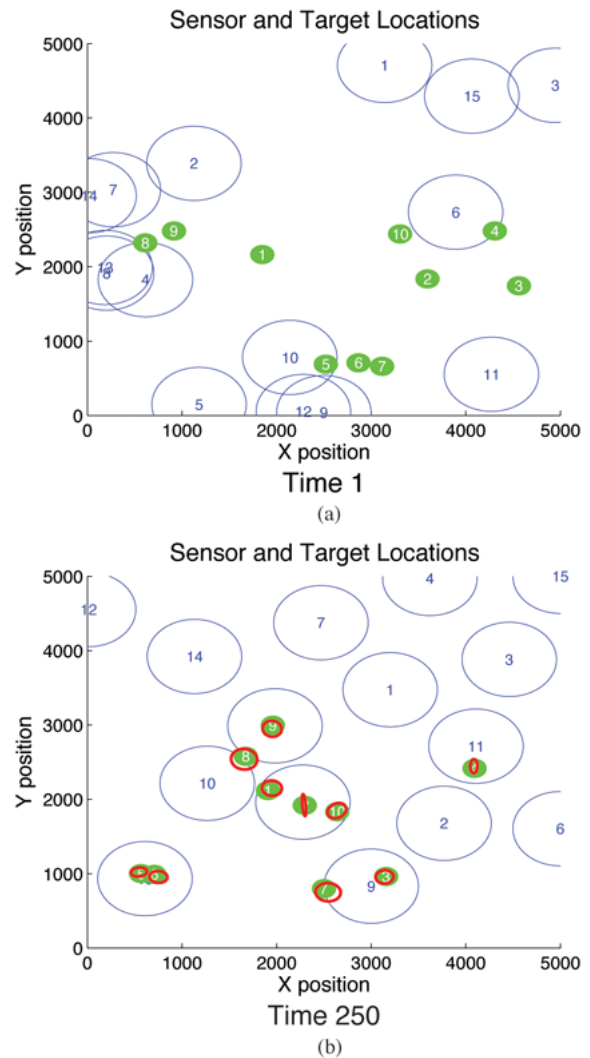
evolution using a simple nearly constant velocity approach, which is in fact mismatched to the actual targets as they routinely perform move-stop-move and other maneuvers. Target birth and death is modeled in the JMPD time evolution as spatially and temporally constant.

Each platform is idealized to hover above the surveillance region and has an imaging sensor that stares directly down. At each time step, the imager measures cells in the surveillance area by making measurements on a grid with  $100\text{ m} \times 100\text{ m}$  detection cell resolution. The model problem setup is illustrated in Fig. 2.

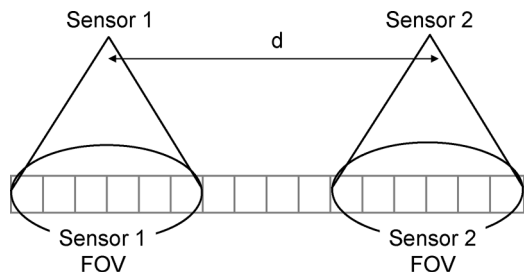
When measuring a cell, the imager returns either a 0 (no detection) or a 1 (detection) which is governed by a probability of detection ( $p_d$ ) and a per-cell false alarm rate ( $p_f$ ). The sensor is modeled to have a field of view with radius 5 cells from its center and hence measures a circular patch on the ground. The effective SNR is maximum at the center and falls off as  $r^2$  at the periphery. We fix  $\text{SNR}_{\max} = 16\text{ dB}$ ,  $p_f = 0.01$ , and use  $p_d =$

$p_f^{1/(1+\text{SNR})}$ , which is a standard model for thresholded detection of Rayleigh returns [49]. When there are  $T$  targets in the same cell, the detection probability increases according to  $p_d(T) = p_f^{1/(1+\text{SNR} \cdot T)}$ . Fig. 3 illustrates the SNR and  $p_d$  as a function of distance from field of view center.

Each platform computes a local estimate of the JMPD using measurements it has made and measurements



**Fig. 4.** The random positioning of the 15 platforms at initialization (top) and after some time (bottom). The platform position is given by the blue number and its field of view is described by the circle surrounding the number. The true position of each of the ten moving ground targets is shown by the green numbered circles. The estimate of the position for the targets (taken from the omniscient fuser) is given by the red covariance ellipses. Qualitatively, after some time, the platforms have preferentially aligned themselves over the targets while still allocating some network resources to look for incoming (new) targets. (a) The initial (randomly placed) deployment of 15 sensors in a surveillance region. (b) The configuration of the sensors after 250 time steps.



**Fig. 5.** Two platforms are a distance  $d$  apart. When  $d$  is large, their fields of view do not overlap, and the sum of individual information gains is close to the joint information gain. Conversely, when the platforms are close, the joint information gain differs significantly from the sum of individual information gains.

received from neighbors. Platforms then use the joint constrained information theoretic optimization approximation described in the previous section to compute next best movements.

Fig. 4 shows an initial (random) positioning of the 15 sensors and the position after some time. As can be seen from the figure, over time the sensors preferentially align themselves around the targets (which were discovered through repeated interrogation of the ground) while still allocating some resources to look for new targets.

2) *A Comparison of the Correction Term and the Physicomimetic Approximation:* As described in Section IV, the joint constrained information theoretic optimization is rewritten as a sum of single information theoretic optimizations, a correction term, and a relaxation term. These last two terms are approximated with a physicomimetic term resulting in a computationally tractable approach. In this section, we provide a comparison between the approximation term and the correction term in the model problem as motivation for its use.

We consider two sensors that are each able to measure cells in the surveillance region as described above. Of interest is the difference between the information gain for a pair of actions  $(r^1, r^2)$  when evaluated jointly as compared to the sum of individual information gains (i.e., the correction term). We can examine the discrepancy as a function of the distance  $d$  between the platforms. This is illustrated in Fig. 5.

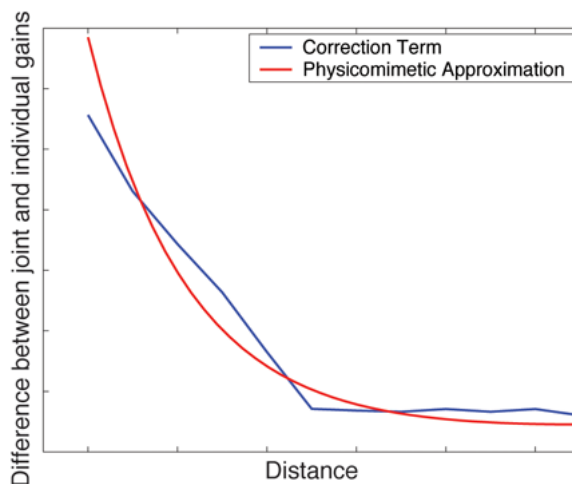
When the platforms are far apart, there is very little difference between the sum of individual platform information gains and the full joint information gain. As the platforms move closer, the sum of individual information gain terms overestimates the value actions by “double-counting” information (among other things). Fig. 6 illustrates the discrepancy in information gain estimation (i.e., the difference between the full joint optimization and the sum of individual optimizations) as a function of platform distance  $d$ . Additionally, the (scaled)

Lennard–Jones force is superimposed to provide motivation for its use.

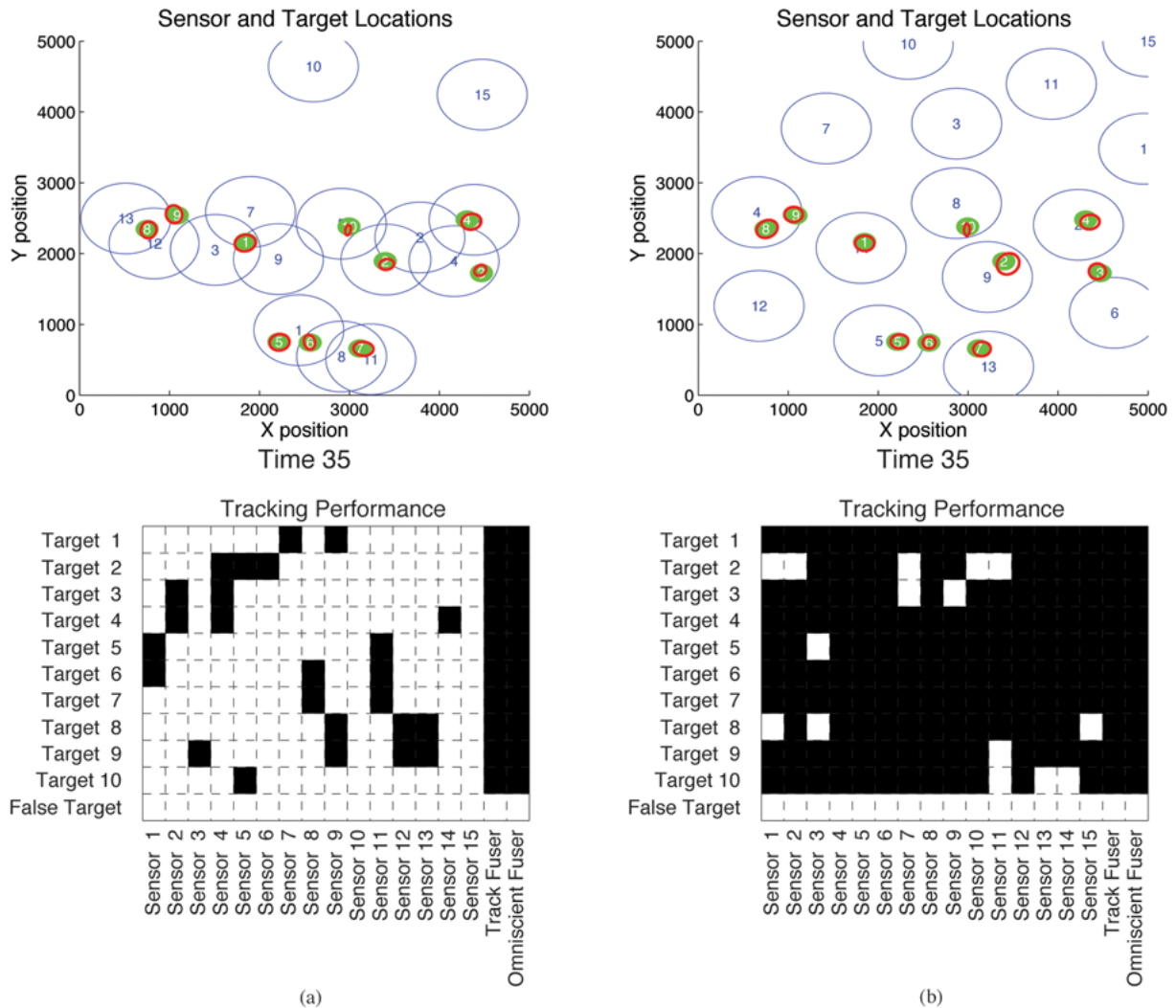
3) *Emergent Behavior as a Function of Communication Radius:* Fig. 7 illustrates the effect of communication radius on network behavior. When the communication radius is high, platforms spread out nearly evenly (while preferentially staying with targets) as each platform knows where (most of) the others are and that the existing targets are being covered. Conversely, in the low-communication radius setting, platforms tend to cluster near targets. This is because a platform does not know where other platforms are unless they are close (within the communication radius) and furthermore does not know if targets are being effectively maintained by other platforms until they are nearby. Despite this difference in behavior, in both cases the number and position of targets has been correctly learned by the network. However, in the high-communication radius case, each individual sensor knows much more. The net effect of this additional knowledge present in the high-communication setting is that if a platform were to fail, its duties would be picked up by another platform in the network much more quickly.

4) *Monte Carlo Simulation of Performance:* Fig. 8 presents the results of a Monte Carlo simulation of performance in this model problem. We illustrate the network knowledge in three ways.

- At the **Average Sensor:** Each sensor has a local estimate of the JMPD whose fidelity is governed by the communication radius. Therefore, at low-communication radius, each sensor only has knowledge only of the local area, and hence will



**Fig. 6.** A comparison of the correction term and the physicomimetic approximation as a function of distance between two platforms.



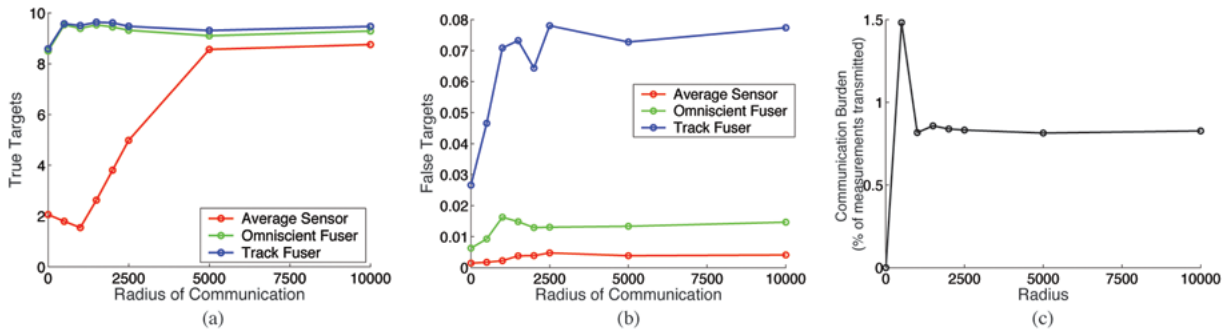
**Fig. 7.** The qualitative behavior of the platforms as a function of communication radius. In each graphic, the top plot shows the position of the platforms and targets in the surveillance region. The bottom plot shows which sensors (1..15) know about which of the ten targets (a black square indicates that a sensor knows about the target). The omniscient fuser and track fuser performance are included for reference. In both cases, all targets are successfully detected and tracked with no false targets at the displayed time. However, the behavior of the system as communication radius changes is markedly different. In the low-communication radius case (top-left), platforms tend to group heavily around existing targets, while in the high-communication radius case (top-right), platforms spread out more. Furthermore, as the bottom plots indicate, in the low-communication radius case (bot-left) platforms tend to only know about nearby targets, whereas in the high-communication radius case (bot-right), platforms have a very global picture. (a) Steady-state behavior when the communication radius is low ( $r = 500$  m). (b) Steady-state behavior when the communication radius is high ( $r = 5000$  m).

only provide estimates of nearby targets. As communication radius increases, sensors become more aware of the entire region.

- At the **Track Fuser**: Aperiodically, individual sensor estimates must be coalesced to provide a single picture of the surveillance region. We assume for bandwidth conservation purposes that sensors transmit estimates about confirmed targets only to a base station rather than the entire (local) JMPD estimate. The base station then fuses these

tracks to provide an estimate of the entire surveillance region.

- At the **(hypothetical) Omniscient Fuser**: To benchmark performance, we also include a (hypothetical) omniscient fuser that receives all measurements made by all nodes in the network and constructs the optimal JMPD estimate. Note: this entity is used only for constructing the figure and is not used in the simulation in any way. In particular, all sensor management decisions are



**Fig. 8. Monte Carlo performance results for the 15 sensor region surveillance application. (a) The average number of true targets correctly detected (ten is perfect) for the average sensor, the omniscient fuser, and the track fuser as a function of communication radius. (b) The average number of false targets incorrectly detected (zero is perfect) for the average sensor, the omniscient fuser, and the track fuser as a function of communication radius. (c) The average communication burden of the proposed decentralized approach as a function of communication radius.**

computed locally using the local estimate of the JMPD.

The performance of the network is measured in two ways.

- The number of **True Targets** detected and tracked. This measures the number, of ten possible, of actual targets that have been successfully detected and tracked (i.e., have position estimates that are within some allowable amount).
- The number of **False Targets** incorrectly thought to exist. This measures the number of targets that are thought to exist when in fact they do not. Sensors receive false alarms (detections when in fact no target exists) according to the false alarm rate  $p_f$ . When a number of false alarms occur in a row or when the sensor does not properly reinterrogate, a false target may be created.

Additionally, we look at the **Communication Requirements** of the method in terms of the percent of measurements that each node transmits. A node measures some number of cells at each time step. It then uses the (local) JMPD to compute the likelihood that each measurement originated from a target. Those measurements (along with the platform position) that have likely originated from a target are broadcast to be received by any neighbor within the broadcast radius. Since the target density in this experiment is low, the number of measurements truly originating from targets is also low. Therefore it is to be expected that the number of transmitted measurements will be small.

Each simulation runs 250 time steps. Fig. 8 presents the results of the average number of true targets, average number of false targets from time step 50 on (after the burn-in time where the initially ignorant network has been able to learn about the surveillance region), and communication burden for each of the three entities (the average sensor, the track fuser, and the omniscient fuser).

## B. A Simulation With a Large Number of Very Limited Capability Platforms

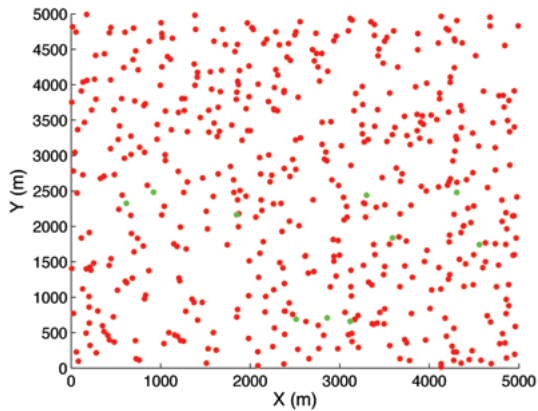
1) *Description of the Model Problem:* In this subsection, we turn our attention to a setting where surveillance is to be performed with a large number (hundreds or thousands) of inexpensive low-capability sensors. The simulation uses the same region size and target motion data as the previous simulation. Again, the platforms are idealized to hover above the surveillance region and stare directly down. However, in this simulation each sensor is capable of only measuring a single detection cell immediately below the platform and has degraded detection capabilities (SNR = 10 dB). Fig. 9 shows a typical (random) initial deployment of sensing assets.

2) *Emergent Behavior With Different Scheduling Methods:* In Section IV, we saw that the optimal multiplatform information theoretic scheduling criteria was in fact a joint constrained information theoretic optimization. Through algebraic manipulation, Lagrangian relaxation, and direct approximation we proposed a method of approximate scheduling that ultimately results in a sensor being compelled to move by two competing forces: one based on greedily maximizing information gain, and one based on physicomimetics that acts to keep sensors apart and promote region exploration in just the correct manner.

In this section, we illustrate how the combination of these two forces promotes just the correct platform behavior and that the individual forces themselves are not sufficient. Specifically, we compare both qualitatively and quantitatively the surveillance performance of a network of sensors with three different scheduling algorithms:

- the proposed **Combination of Information Theoretic Forces and Physicomimetic Forces**,



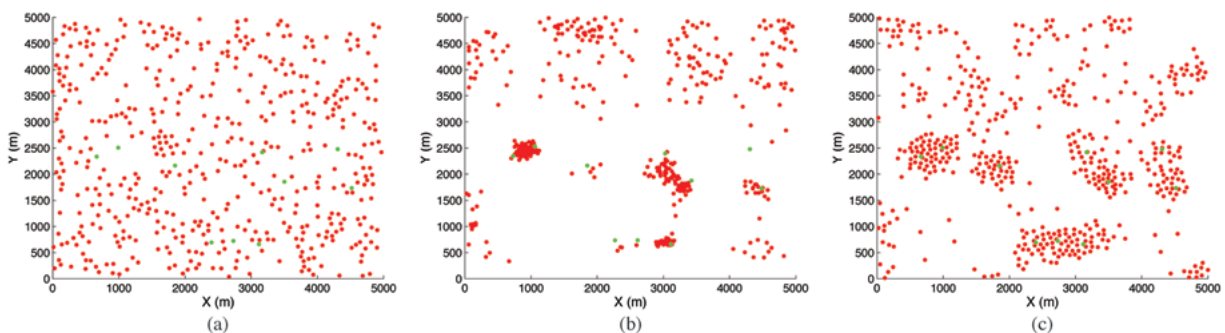


**Fig. 9.** The (random) initial deployment of 500 platforms in a surveillance region. The position of each of the platforms is shown by a red dot. The true location of the ground targets is shown by the green dots (of course, both the number of targets and their kinematic states are unknown at initialization). Each platform has a low-capability sensor that merely measures a single  $100\text{ m} \times 100\text{ m}$  pixel immediately below for the presence or absence of targets.

which provides a balance between information seeking behavior and explorative behavior and is connected directly with the optimal multiplatform scheduling method;

- a purely **Information Theoretic Method**, which tasks sensors to take actions that maximize information gain (only);
- a purely **Physicomimetic Method**, which maintains separation between sensors using the repulsive force (only).

Fig. 10 shows the steady-state platform positioning of 500 platforms under each of the three methods.



**Fig. 10.** The combination of information theoretic forces and physicomimetic forces drives the sensors to behave in a manner that combines the explorative nature of physicomimetics and the exploitative nature of information theoretic optimization. (a) Steady-state positioning of platforms controlled by the physicomimetic force only. Notice that the platforms simply spread out in the region to avoid collision. (b) Steady-state positioning of platforms controlled by the information theoretic force only. Notice that the platforms overcluster near the true target positions and have large regions that are not explored. (c) Steady-state positioning of platforms controlled by the combination physicomimetic and information theoretic forces. Here, platforms both explore the entire region and preferentially cluster near real targets.

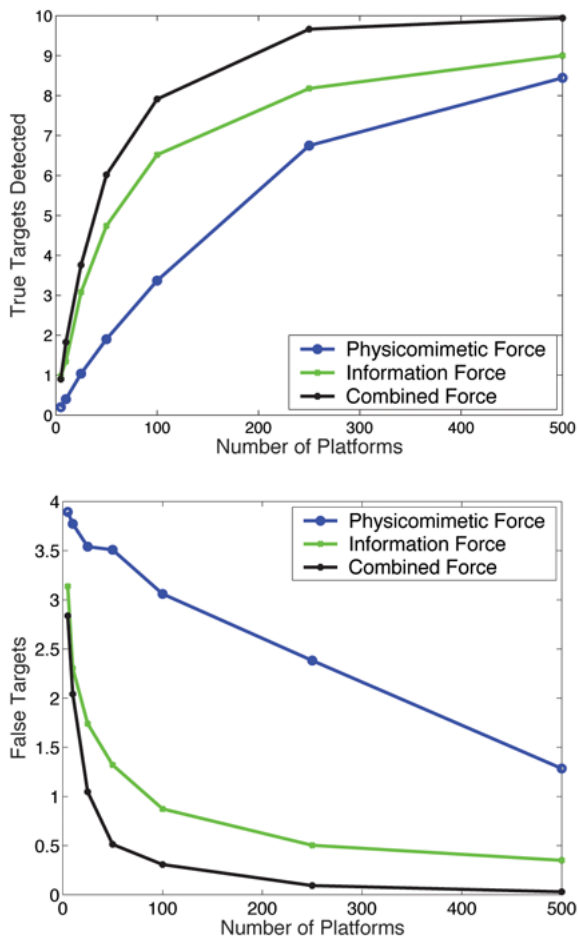
3) *Monte Carlo Simulation of Performance:* We again display the performance of the scheduling algorithm based on: 1) the number of true targets detected and 2) the number of false targets reported. Fig. 11 shows the performance of the proposed scheduling algorithm versus the number of platforms in comparison to the behavior of the two constituent components alone.

This figure shows that the proposed method effectively combines the strengths of the constituent methods. The physicomimetic method enforces collaboration and explorative behavior by encouraging platforms to maintain spatial separation. When used alone, this results in good detection capability but poor tracking capability, as once a target is found there is no impetus to continue to follow its motion. Furthermore, spurious detections that are the result of the false alarm process are not tracked down through reinterrogation, resulting in more false targets. Conversely, the information theoretic method encourages exploitative behavior. When used alone, this results in poor detection capability but good tracking capability. Platforms tend to cluster around known targets and track them very well but do not have the impetus to look for new targets in unsurveyed regions. False targets are minimized but real targets are less likely to be found. The proposed method, which combines these two forces, as motivated by the approximation to the joint constrained information theoretic optimization, manages to use the strengths of both of the constituent methods by both exploring and exploiting in just the right ratio.

## VI. CONCLUSION

This paper has addressed the problem of sensor management for a large network of dynamic sensors. The method presented is a novel combination of particle filtering for





**Fig. 11. Performance of the proposed method versus number of platforms in terms of true targets detected and false targets reported. For comparison purposes, the performance of each of the constituent forces (the physicomimetic force and the information theoretic force) are included. As can be seen in the figures, the combined force method significantly outperforms each of the constituent methods. In fact, the performance of the constituent methods at 500 platforms is similar to the combined method with 50-100 platforms.**

nonparametric density estimation, information theoretic measures for comparing possible action sequences, and artificial physics for providing approximate cooperation between sensor nodes.

This paper has provided three main contributions. First, it has described a mathematical formulation for estimation of the state of the surveillance region based on recursive estimation of the joint multitarget probability density. Numerical estimation of this high dimensional nonparametric density is done online via a novel multitarget particle filter. Second, this paper has presented a new method of sensor management for large dynamic networks that combines information theory and physicomimetics. Use of information theory allows this method to have the property that potential actions which

provide different types of information can be compared on a common footing, that of information gain. Use of physicomimetics provides a tractable and robust approximation to the joint optimization problem. As the number of possible network actions grows exponentially with the number of sensors and number of actions each sensor can take, finding the globally optimum action set is not tractable. Finally, this paper has shown that the method can be decentralized wherein each sensor generates a picture of the surveillance region based on its own measurements and measurements received from neighboring nodes. This local picture then drives the actions of each sensor at the next decision epoch, and also drives which measurements are sent to other sensors.

Future work in this area includes the extension of the methods to long-term (nonmyopic) scheduling. In a manner analogous to multisensor scheduling, (naive) multistep scheduling results in an exponential explosion of potential actions. Therefore, principled approximation methods (perhaps domain-specific) must be developed for tractable implementation. As alluded to earlier, some work has been done in extending the information theoretic scheduling metrics discussed here to the multistep setting, but has focussed mainly on the single platform setting. ■

**APPENDIX**

In this appendix, we show how the multisensor divergence can be written as sum of single-sensor divergences and an explicit correction term. As in the text, we specialize to the case of the Rényi Divergence where  $\alpha \rightarrow 1$  which becomes the Kullback-Leibler Divergence [42].

The KL divergence between two densities  $p_0(x)$  and  $p_1(x)$  is defined as

$$KLD(p_1||p_0) = \int p_0(x) \ln \frac{p_0(x)}{p_1(x)} dx. \tag{30}$$

In the JMPD setting, the divergence between the prediction density and the updated density after all  $P$  sensors have made measurements  $z^1, \dots, z^P$  is from the definition

$$\begin{aligned} &KLD(p(\cdot|z_{0:k-1}, z^1, \dots, z^P)||p(\cdot|z_{0:k-1})) \\ &= \sum_{T_k} \int p(X_k, T_k|z_{0:k-1}) \\ &\quad \times \ln \frac{p(X_k, T_k|z_{0:k-1})}{p(X_k, T_k|z_{0:k-1}, z^1, \dots, z^P)} dX_k. \end{aligned} \tag{31}$$

(Note that we omit from the notation conditioning on past measurements  $r_{0:k-1}$  and current action  $r^1, \dots, r^P$  for notational simplicity).

Using Bayes rule (5) on the denominator of the log term, this can be simplified to

$$\sum_{T_k} \int p(X_k, T_k | z_{0:k-1}) \ln \frac{p(z^1, \dots, z^P | z_{0:k-1})}{p(z^1, \dots, z^P | X_k, T_k)} dX_k. \quad (32)$$

Further simplifying algebraically on the log term, we have

$$\ln p(z^1, \dots, z^P | z_{0:k-1}) - \sum_{T_k} \int p(X_k, T_k | z_{0:k-1}) \times \ln p(z^1, \dots, z^P | X_k, T_k) dX_k. \quad (33)$$

Both log terms can be expanded giving

$$\begin{aligned} & \ln \left( p(z^1 | z_{0:k-1}) \cdots p(z^P | z_{0:k-1}) \frac{p(z^1, \dots, z^P | z_{0:k-1})}{p(z^1 | z_{0:k-1}) \cdots p(z^P | z_{0:k-1})} \right) \\ & - \sum_{T_k} \int p(X_k, T_k | z_{0:k-1}) \\ & \times \ln \left( p(z^1 | X_k, T_k) \cdots p(z^P | X_k, T_k) \right. \\ & \left. \times \frac{p(z^1, \dots, z^P | X_k, T_k)}{p(z^1 | X_k, T_k) \cdots p(z^P | X_k, T_k)} \right) dX_k. \quad (34) \end{aligned}$$

And distributing the logs, the multisensor KLD becomes

$$\begin{aligned} & \ln p(z^1 | z_{0:k-1}) + \cdots + \ln p(z^P | z_{0:k-1}) \\ & + \ln \frac{p(z^1, \dots, z^P | z_{0:k-1})}{p(z^1 | z_{0:k-1}) \cdots p(z^P | z_{0:k-1})} \\ & - \sum_{T_k} \int p(X_k, T_k | z_{0:k-1}) \ln p(z^1 | X_k, T_k) dX_k - \cdots \\ & - \sum_{T_k} \int p(X_k, T_k | z_{0:k-1}) \ln p(z^P | X_k, T_k) dX_k \\ & - \sum_{T_k} \int p(X_k, T_k | z_{0:k-1}) \ln \frac{p(z^1, \dots, z^P | X_k, T_k)}{p(z^1 | X_k, T_k) \cdots p(z^P | X_k, T_k)}. \quad (35) \end{aligned}$$

Recognizing the components as the individual sensor divergences by comparison to (33), we have

$$\begin{aligned} & \text{KLD}(p(\cdot | z_{0:k-1}, z^1) || p(\cdot | z_{0:k-1})) + \cdots \\ & + \text{KLD}(p(\cdot | z_{0:k-1}, z^P) || p(\cdot | z_{0:k-1})) \\ & + \ln \frac{p(z^1, \dots, z^P | z_{0:k-1})}{p(z^1 | z_{0:k-1}) \cdots p(z^P | z_{0:k-1})} \\ & - \sum_{T_k} \int p(X_k, T_k | z_{0:k-1}) \ln \frac{p(z^1, \dots, z^P | X_k, T_k)}{p(z^1 | X_k, T_k) \cdots p(z^P | X_k, T_k)}. \quad (36) \end{aligned}$$

The integral term is 0 since

$$\begin{aligned} & \frac{p(z^1, \dots, z^P | X_k, T_k)}{p(z^1 | X_k, T_k) \cdots p(z^P | X_k, T_k)} \\ & = \frac{p(z^1 | X_k, T_k) p(z^2 | X_k, T_k, z^1) \cdots p(z^P | X_k, T_k, z^{P-1}, \dots, z^1)}{p(z^1 | X_k, T_k) \cdots p(z^P | X_k, T_k)} \\ & = \frac{p(z^1 | X_k, T_k) \cdots p(z^P | X_k, T_k)}{p(z^1 | X_k, T_k) \cdots p(z^P | X_k, T_k)} \\ & = 1 \quad (37) \end{aligned}$$

i.e., as the likelihood in the numerator is conditioned on the truth at the current time  $(X_k, T_k)$ , the additional measurements from other platforms add no information. Note the subtlety that this is *not* the case with the likelihood conditioned on past measurements. Here knowing other sensor measurements does add additional information.

Therefore, the result is that the Kullback–Leibler Divergence between the prediction JMPD and the JMPD after  $P$  sensors have made measurements  $z^1 \cdots z^P$  is simply the sum of the  $P$  single-sensor divergences and a correction term, given explicitly by the log ratio of “informed” likelihoods to “uninformed” likelihoods, i.e.,

$$\begin{aligned} & \text{KLD}(p(\cdot | z_{0:k-1}, z^1, \dots, z^P) || p(\cdot | z_{0:k-1})) \\ & = \sum_{i=1}^P \text{KLD}(p(\cdot | z_{0:k-1}, z^i) || p(\cdot | z_{0:k-1})) \\ & + \ln \frac{p(z^1, \dots, z^P | z_{0:k-1})}{p(z^1 | z_{0:k-1}) \cdots p(z^P | z_{0:k-1})}. \quad (38) \end{aligned}$$

By taking the expected value of both sides and recognizing that the Rényi Divergence becomes the Kullback–Leibler Divergence as  $\alpha \rightarrow 1$ , we have the desired result.

## REFERENCES

- [1] C. Kreucher, K. Kastella, and A. Hero, "Multitarget tracking using the joint multitarget probability density," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.
- [2] M. Morelande, C. Kreucher, and K. Kastella, "A Bayesian approach to multiple target detection and tracking," *IEEE Trans. Signal Process.*, to be published.
- [3] L. D. Stone, T. L. Corwin, and C. A. Barlow, *Bayesian Multiple Target Tracking*. Norwood, MA: Artech House, 1999.
- [4] M. I. Miller, A. Srivastava, and U. Grenander, "Conditional mean estimation via jump-diffusion processes in multiple target tracking/recognition," *IEEE Trans. Signal Process.*, vol. 43, no. 11, pp. 2678–2690, Nov. 1995.
- [5] I. Goodman, R. Mahler, and H. Nguyen, *Mathematics of Data Fusion*. Norwell, MA: Kluwer, 1997.
- [6] S. Mori, C. Y. Shong, E. Tse, and R. P. Wishner, "Tracking and classifying multiple targets without a prior identification," *IEEE Trans. Autom. Control*, vol. AC-31, no. 5, pp. 401–409, May 1986.
- [7] E. W. Kamen, "Multiple target tracking based on symmetric measurement functions," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 371–374, Mar. 1992.
- [8] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.
- [9] M. Orton and W. Fitzgerald, "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 216–223, Feb. 2002.
- [10] S. Maskell, M. Rollason, N. Gordon, and D. Salmond, "Efficient particle filtering for multiple target tracking with application to tracking in structured images," in *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, 2002, vol. 4728, pp. 251–262.
- [11] M. Isard and J. MacCormick, "BraMBLE: A Bayesian multiple-blob tracker," in *Proc. 8th Int. Conf. Computer Vision*, 2001, vol. 2, pp. 24–31.
- [12] H. Sidenbladh, "Multi-target particle filtering for the probability hypothesis density," in *Proc. Int. Conf. Information Fusion*, 2003, pp. 800–806.
- [13] C. Hue, J.-P. Le Cadre, and P. Perez, "Tracking multiple objects with particle filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 3, pp. 791–812, Jul. 2002.
- [14] A. Doucet, B.-N. Vo, C. Andrieu, and M. Davy, "Particle filtering for multi-target tracking and sensor management," in *Proc. 5th Int. Conf. Information Fusion*, 2002, vol. 1, pp. 474–481.
- [15] C. Kreucher, K. Kastella, and A. Hero, "Sensor management using an active sensing approach," *Signal Process.*, vol. 85, no. 3, pp. 607–624, Mar. 2005.
- [16] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Process. Mag.*, vol. 19, pp. 61–72, Mar. 2002.
- [17] K. J. Hintz and E. S. McVey, "Multi-process constrained estimation," *IEEE Trans. Man, Syst. Cybern.*, vol. 21, no. 1, pp. 434–442, Jan./Feb. 1991.
- [18] W. Schmaedeke and K. Kastella, "Event-averaged maximum likelihood estimation and information-based sensor management," *Proc. SPIE*, vol. 2232, pp. 91–96, Jun. 1994.
- [19] R. Mahler, "Global optimal sensor allocation," in *Proc. 9th Nat. Symp. Sensor Fusion*, 1996, vol. 1, pp. 167–172.
- [20] J. Denzler and C. M. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 145–157, Feb. 2002.
- [21] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. New York: Ellis Horwood, 1994.
- [22] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995, vol. 1–2.
- [23] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Oper. Res.*, vol. 21, pp. 1071–1088, 1973.
- [24] L. Meier, J. Perschon, and R. M. Dressler, "Optimal control of measurement systems," *IEEE Trans. Autom. Control*, vol. AC-12, no. 5, pp. 528–536, Oct. 1967.
- [25] C. Kreucher, A. Hero, K. Kastella, and D. Chang, "Efficient methods of non-myopic sensor management for multitarget tracking," in *Proc. 43rd IEEE Conf. Decision and Control (CDC)*, 2004, vol. 1, pp. 722–727.
- [26] E. K. P. Chong, C. Kreucher, and A. Hero, "Pomdp approximation methods based on heuristics and simulation," in *Foundations and Applications of Sensor Management*, K. Kastella, D. Cochran, A. Hero, and D. Castañon, Eds. New York: Springer.
- [27] V. Krishnamurthy, "Emission management for low probability intercept sensors in network centric warfare," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 133–152, Jan. 2005.
- [28] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1382–1397, Jun. 2002.
- [29] Y. He and E. K. P. Chong, "Sensor scheduling for target tracking in sensor networks," in *Proc. 2004 IEEE Conf. Decision and Control*, pp. 734–748.
- [30] D. Castañon, "Approximate dynamic programming for sensor management," in *Proc. 1997 IEEE Conf. Decision and Control*, vol. 2, pp. 1202–1207.
- [31] W. Spears, R. Heil, D. Spears, and D. Zarzhitsky, "Physicomimetics for mobile robot formations," in *Proc. 3rd Int. Joint Conf. Autonomous Agents and Multi Agent Systems (AAMAS-04)*, vol. 3, pp. 1528–1529.
- [32] W. D. Spears, J. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," *Auton. Robots*, vol. 17, no. 2–3, 2004.
- [33] Y. Bar-Shalom, *Multitarget Multisensor Tracking: Advanced Applications*. Norwood, MA: Artech House, 1990.
- [34] C. Kreucher, K. Kastella, and A. Hero, "Tracking multiple targets using a particle filter representation of the joint multitarget probability density," in *Proc. SPIE: Signal and Data Processing of Small Targets*, 2003, O. E. Drummond, Ed., vol. 5204, pp. 258–269.
- [35] M. Morelande, C. Kreucher, and K. Kastella, "A study of factors in multiple target tracking with a pixelized sensor," in *Proc. SPIE: Signal and Data Processing of Small Targets 2005*, O. E. Drummond, Ed., vol. 5913, pp. 155–167.
- [36] K. Kastella, "Event averaged maximum likelihood estimation and mean-field theory in multitarget tracking," *IEEE Trans. Autom. Control*, vol. 50, no. 6, pp. 1070–1073, Jun. 1995.
- [37] C. Kreucher, A. Hero, K. Kastella, and B. Shapo, "Information-based sensor management for simultaneous multitarget tracking and identification," presented at the 13th Annu. Conf. Adaptive Sensor Array Processing (ASAP), Lexington, MA, 2005.
- [38] C. Kreucher, A. Hero, and K. Kastella, "A comparison of task driven and information driven sensor management for target tracking," in *Proc. 44th IEEE Conf. Decision and Control*, 2005, pp. 4004–4009.
- [39] C. Kreucher, D. Blatt, A. Hero, and K. Kastella, "Adaptive multi-modality sensor scheduling for detection and tracking of smart targets," *Dig. Signal Process.*, to be published.
- [40] C. Kreucher, K. Kastella, and A. Hero, "Information-based sensor management for multitarget tracking," in *Proc. SPIE: Signal and Data Processing of Small Targets 2003*, O. E. Drummond, Ed., vol. 5204, pp. 480–489.
- [41] K. J. Hintz, "A measure of the information gain attributable to cueing," *IEEE Trans. Syst., Man Cybern.*, vol. 21, no. 2, pp. 237–244, 1991.
- [42] A. O. Hero, III, B. Ma, O. Michel, and J. D. Gorman, "Alpha divergence for classification, indexing and retrieval," *Comm. and Sig. Proc. Lab., Dept. EECS, Univ. Michigan*, Tech. Rep. 328, 2001.
- [43] A. Hero, C. Kreucher, and D. Blatt, "Information theoretic approaches to sensor management," in *Foundations and Applications of Sensor Management*, K. Kastella, D. Cochran, A. Hero, and D. Castañon, Eds. New York: Springer.
- [44] J. Borenstein, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 1179–1187, May 1989.
- [45] C. Kreucher, K. Kastella, and A. Hero, "Multi-platform information-based sensor management," in *Proc. SPIE: Defense Transformation and Network-Centric Systems*, R. Suresh, Ed., 2005, vol. 5820, pp. 141–151.
- [46] C. Kreucher, K. Kastella, J. Wegryzn, and B. Rickenbach, "An information based approach to decentralized multi-platform sensor management," in *Proc. SPIE: Defense Transformation and Network-Centric Systems*, R. Suresh, Ed., 2006, vol. 6249, pp. 134–145.
- [47] B. Grocholsky, "Information-theoretic control of multiple sensor platforms," Ph.D. dissertation, Univ. Sydney, Sydney, Australia, 2002.
- [48] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. Oxford, U.K.: Oxford Univ. Press, 1989.
- [49] S. Blackman, *Multiple-Target Tracking With Radar Applications*. Norwood, MA: Artech House, 1986.

ABOUT THE AUTHORS

**Christopher M. Kreucher** received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Michigan, in 1997, 1998, and 2005, respectively.

Since 1998, he has been a Staff Scientist at General Dynamics Advanced Information Systems' Michigan Research and Development Facility (formerly ERIM), near Ann Arbor. His current research interests include nonlinear filtering (specifically particle filtering), Bayesian methods of multitarget tracking, self-localization, information theoretic sensor management, and distributed swarm management.



**Keith D. Kastella** received the B. A. degree in physics from Reed College, Portland OR, in 1980 and the M.S. and Ph.D. degrees in physics from the State University of New York, Stony Brook, in 1985 and 1988, respectively.

He is a Chief Scientist at General Dynamics Advanced Information Systems, Ypsilanti, MI. His current research interests are in data fusion, signal processing, and novel applications of quantum entanglement in sensing and signal processing.



**Alfred O. Hero, III** (Fellow, IEEE) received the B.S. degree (*summa cum laude*) from Boston University, Boston, MA, in 1980 and the Ph.D degree from Princeton University, Princeton, NJ, in 1984, both in electrical engineering.

Since 1984 he has been with the University of Michigan, Ann Arbor, where he is a Professor in the Department of Electrical Engineering and Computer Science and, by courtesy, in the Department of Biomedical Engineering and the Department of Statistics. His recent research interests have been in areas including: inference in sensor networks, adaptive sensing, bioinformatics, inverse problems, and statistical signal and image processing.

Prof. Hero has received a IEEE Signal Processing Society Meritorious Service Award (1998), a IEEE Signal Processing Society Best Paper Award (1998), and the IEEE Third Millennium Medal (2000). He is currently President of the IEEE Signal Processing Society (2006-2008).



**Mark R. Morelande** received the B.Eng. degree in aerospace avionics from Queensland University of Technology, Brisbane, Australia, in 1997 and the Ph.D. degree in electrical engineering from Curtin University of Technology, Perth, Australia, in 2001.

From November 2000 to January 2002, he was a Postdoctoral Fellow at the Centre for Eye Research, Queensland University of Technology. From January 2002 to July 2005 he was a Research Fellow at the Cooperative Research Centre for Sensor, Signal and Information Processing, University of Melbourne, Parkville, Australia. He is now a Senior Research Fellow in the Melbourne Systems Laboratory, also at the University of Melbourne. His research interests include nonstationary signal analysis and target tracking with particular emphasis on multiple target tracking and sequential Monte Carlo methods.

