Network Sensor Management for Tracking and Localization

Alfred O. Hero III
Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI, U.S.A.
Email: hero@umich.edu

Christopher M. Kreucher
Michigan Research and Development Center
General Dynamics
Ypsilanti, MI, U.S.A.
Email: Christopher.Kreucher@gd-ais.com

***Abstract*— This paper addresses the problem of sensor management for a large network of agile sensors. Sensor management refers to 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 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 to be detected and tracked by a network of mobile sensors.

Keywords: multiplatform sensor management, joint multitarget probability density

I. Introduction

Large networks of inexpensive sensors provide a means for performing persistent and ubiquitous surveillance over a wide region. In this paper, we address the problem of managing the resources of a network consisting of a large number (i.e., 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 sensor network adjusts its behavior dynamically so that the overall utility of the network is maximized.

Portions of this work were supported under the USAF contract F33615-02-C-1199 and by the DARPA MURI program under ARO contract DAAD19-02-1-0262.

Sensor management in large networks is challenging for a host of reasons. First, the state space required to characterize the surveillance region is typically of very high dimension and poorly represented by a parametric form. It is this state space that the network of nodes is to estimate, so proper mathematical formulation and efficient implementation is key. Second, the network must simultaneously address many competing goals (e.g., detection of new areas of interest while monitoring known areas of interest), and so the scheduling metric must be chosen to appropriately balance these goals. Third, exact maximization of overall network utility is intractable as the number of actions at each decision epoch is exponential in the number of nodes. Therefore, a principled approximation to simultaneous multiplatform scheduling must be employed. Fourth, there must be information sharing between the individual 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 (Section II, information theoretic measures for estimating the value of possible future actions (Section III), and physicomimetics for providing a tractable approximation to the joint optimization (Section IV. Section V provides simulation results illustrating the efficacy of the approach.

II. The Joint Multitarget Probability Density (JMPD)

This section reviews the Joint Multitarget Probability Density (JMPD) and its Particle Filter (PF) implementation [1]–[3]. 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.

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

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

is the density for T targets with states $x^1, x^2, ..., x^{T-1}, x^T$ at time k based on a set of past observations $z_{0:k}$.

The observation set $z_{0:k}$ is the collection of measurements up to and including time k, i.e., $z_{0:k} \doteq \{z_0, z_1, \cdots, z_k\}$, where each z_i may be a single measurement or a collection of measurements made at time i (e.g., a vector, matrix, or a concatenation of measurements from multiple sensors made at the same time). We will refer to measurements made at 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 $p(x_k^1, x_k^2, ... x_k^{T-1}, x_k^T, T_k | z_{0:k}, r_{0:k})$. For simplicity, this notation is suppressed in the present discussion.

Each of the single target state vectors x^i in the density $p(x_k^1, x_k^2, \cdots, x_k^{T-1}, x_k^T | T_k, z_{0:k})$ is a vector quantity. We will typically use the two-dimensional target state idealization $[x, \dot{x}, y, \dot{y}]$ when providing concrete examples.

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, $T_k = 0 \cdots \infty$. Therefore the normalization condition that the JMPD must satisfy is

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

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

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

where 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. The model of how the JMPD evolves 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 (KP). The KP includes models of target motion, birth and death, and any additional prior information on kinematics that may exist. In the case where identification is part of the state, different models may be used for different target types.

The time-updated density is computed via

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

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 as it contains all possible configurations of state vectors $X_k = \{x_k^1, \cdots, x_k^{T_k}\}$ for all possible values of T_k. Thus, for computational tractability, a sophisticated approximation method is required [2], [3].

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, it 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 :

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

Similarly, a particle is a concatenation of $T^{(i)}$ states,

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

which says particle i estimates there are $T^{(i)}$ targets, where $T^{(i)}$ can be any non-negative integer, and in general is different for different particles. To formalize, let δ_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, $T = 0, 1, 2, \cdots$. As each of the particles is a sample drawn from the JMPD, a particle may estimate $0, 1, 2, \cdots$ targets.

III. Information Theoretic Sensor Management

This section describes a method of sensor management based on maximizing information flow [4], [5]. 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.

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, also known as the α-divergence:

$$D_\alpha(p_1||p_0) = \frac{1}{\alpha-1}\ln\int p_1^\alpha(x)p_0^{1-\alpha}(x)dx \ . \tag{10}$$

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

$$\lim_{\alpha\to 1} D_\alpha(p_1||p_0) = \int p_0(x)\ln\frac{p_0(x)}{p_1(x)}dx \ . \tag{11}$$

B. Rényi Divergence Between the Prior and Posterior JMPD

The function D_α in eq. (10) 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

$$\begin{aligned} D_\alpha\Big(p(\cdot|z_{0:k-1}, r_{0:k-1}, z_k, r_k)||p(\cdot|z_{0:k-1}, r_{0:k-1})\Big) = \\ \frac{1}{\alpha-1}\ln\sum_{T_k}\int p^\alpha(X_k, T_k|z_{0:k-1}, r_{0:k-1}, z_k, r_k)\times \\ p^{1-\alpha}(X_k, T_k|z_{0:k-1}, r_{0:k-1})dX_k \ . \end{aligned} \tag{12}$$

Using Bayes' formula (eq. (5)), we obtain

$$\begin{aligned} D_\alpha\Big(p(\cdot|z_{0:k-1}, r_{0:k-1}, z_k, r_k)||p(\cdot|z_{0:k-1}, r_{0:k-1})\Big) = \\ \frac{1}{\alpha-1}\ln\frac{1}{p^\alpha(z_k|z_{0:k-1}, r_{0:k-1}, r_k)}\times \\ \sum_{T_k}\int p^\alpha(z_k|X_k, T_k, r_k)p(X_k, T_k|z_{0:k-1}, r_{0:k-1})dX_k \ . \end{aligned} \tag{13}$$

C. The Expected Rényi Divergence for a Sensing Action

To determine the best action to take next, we must 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

$$\begin{aligned} \mathbb{E}[D_\alpha] \doteq \int dz_k p(z_k|z_{0:k-1}, r_{0:k-1}, r_k)\times \\ D_\alpha\Big(p(\cdot|z_{0:k-1}, r_{0:k-1}, z_k, r_k)||p(\cdot|z_{0:k-1}, r_{0:k-1})\Big) \ . \end{aligned} \tag{14}$$

The expectation in eq. (14) 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 is to choose the action $\hat{r}_k$ as the one that maximizes the expected information gain, i.e.,

$$\hat{r}_k = \arg\max_{r_k} \mathbb{E}[D_\alpha] \ . \tag{15}$$

In practice, certain r_k are infeasible. There are *kinematic constraints* of the platform, including maximum platform velocity and 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 need the constrained optimization

$$\hat{r}_k = \arg\max_{r_k\in\mathbb{C}} \mathbb{E}[D_\alpha] \ . \tag{16}$$

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 miss-classification 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 [6]. 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 eq. (14). This computation is $O(M)$ where M is the (small) number of (discrete) actions possible.

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). The new position r of the sensor is in principle a 3 dimensional 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 [7]. 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 (eq. (14)). 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 $F_I(r_k) = -\beta \nabla_{r_k} \mathbb{E}[D_\alpha]$, where β is a scaling parameter. 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).

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 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^{th} sensor at time k. Next, let $\vec{r}_k$ and $\vec{z}_k$ denote the sensing actions (here the new positioning of the P platforms) and measurements for the P platforms at time k, respectively. That is, let $\vec{r}_k = [r_k^1, r_k^2, \cdots, r_k^{P-1}, r_k^P]$ and $\vec{z}_k = [z_k^1, z_k^2, \cdots, z_k^{P-1}, z_k^P]$. Then multisensor information theoretic scheduling seeks to find the best choice of sensor actions $\hat{\vec{r}}_k$ as given by eq. (17), where the integral is to be interpreted as performing the P integrations required.

$$\hat{\vec{r}}_k = \arg\max_{\vec{r}_k \in \mathbb{C}'} \mathbb{E}[D_\alpha^P] \doteq \int d\vec{z}_t p(\vec{z}_k | z_{0:k-1}, r_{0:k-1}, \vec{r}_k) \times$$
$$D_\alpha \Big(p(\cdot | z_{0:k-1}, r_{0:k-1}) || p(\cdot | z_{0:k-1}, r_{0:k-1}, \vec{z}_k, \vec{r}_k) \Big) \ . \quad (17)$$

Analogously to eq. (14), the expectation in eq. (17) 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 $\mathbb{C}'$ is expanded beyond the single sensor constraint set to now include both the original constraints of $\mathbb{C}$ and a new constraint that sensors do not collide with each other. That is

$$\mathbb{C}' = \mathbb{C} \cap \{ \|r^i - r^j\| > d \ \forall i, j \text{ where } i \neq j \ \} \ . \quad (18)$$

B. Connection to Single Sensor Optimization

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

$$\arg\max_{\vec{r}_k \in \mathbb{C}'} \sum_{i=1}^{P} \mathbb{E}[D_\alpha^{(i)}] + \mathbb{E}\Big[h(\vec{z}_k, \vec{r}_k, z_{0:k-1}, r_{0:k-1}) \Big] \ . \quad (19)$$

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 $\alpha \to 1$, the correction term can be written explicitly and the simplification becomes

$$\arg\max_{\vec{r}_k \in \mathbb{C}'} \sum_{i=1}^{P} \mathbb{E}[D_\alpha^{(i)}] + \quad (20)$$
$$\mathbb{E}\Big[\ln \Big(\frac{p(z_k^1, \cdots, z_k^P | r_k^1, \cdots, r_k^P, z_{0:k-1}, r_{0:k-1})}{p(z_k^1 | r_k^1, z_{0:k-1}, r_{0:k-1}) \cdots p(z_k^P | r_k^P, z_{0:k-1}, r_{0:k-1})} \Big) \Big]$$

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 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 independent actions, 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. It is not computationally feasible to enumerate and censor all action sets that violate the constraint. Therefore, we address this constraint by defining the Lagrangian

$$L(\vec{r}_k) = \mathbb{E}[D_\alpha^P] + \lambda f(\vec{r}_k)$$
$$= \sum_{i=1}^{P} \mathbb{E}[D_\alpha^{(i)}] + \mathbb{E}\big[h(\vec{z}_k, \vec{r}_k, z_{0:k-1}, r_{0:k-1})\big] + \lambda f(\vec{r}_k) \ ,$$

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

$$\hat{\vec{r}}_k = \arg\max_{\vec{r}_k} L(\vec{r}_k) \ . \tag{21}$$

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. In our method, we simultaneously approximate both the information coupling term involving the expectation of h and the collision prevention term f by introducing a function which reduces the value of action sets that involve sensors moving close together. We have chosen to use a physicomimetic force [7] to provide this approximation, although other similar approximations are also valid. Evaluating this force has a very small computational burden, and requires only that a node know the positions of its neighbors.

Since we remain in a continuous action space environment, we must cast this 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 (Other approximations with a similar attractive-repulsive character would also be viable). The Lennard-Jones force for a pair of platforms i, j separated by a distance $d_{i,j}$ is radial with magnitude

$$F_{LJ}(d_{i,j}) = -\epsilon \left[m\frac{\gamma^m}{d_{i,j}^{m+1}} - n\frac{\gamma^n}{d_{i,j}^{n+1}} \right] \ . \tag{22}$$

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 eq. (22)). 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 eq. (17) results in the final approximate multiplatform optimization

$$\hat{\vec{r}}_k = \arg\max_{\vec{r_k}} \mathbb{E}[D_\alpha^{(i)}] + \lambda \mathbf{F}_{LJ}^{i}(r_k^i) \ .$$

This approximation can be viewed as driving sensors to compute greedy actions (i.e., ignoring actions of other sensors) and correcting by compelling sensors to stay away from others. These two forces are balanced through λ, which when properly chosen, allows sensors to come near when the warranted (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 allows each sensor to compute its next action in a completely distributed manner, assuming each sensor has (a) knowledge of the other sensors positions, and (b) knowledge of the JMPD (or alternatively has access to all measurements the network has made). The first portion of the term in simply requires the expected information gain computed at each node without regard to the actions of other nodes. The second portion of the term requires only that each node know of the position of the nearby nodes.

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. 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 than 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

This section presents two simulation studies illustrating the efficacy of the sensor management method presented here.

The first case study uses a small number (15) of capable platforms for region surveillance. This simulation implements the decentralized version of the algorithm by (a) estimating the (local) JMPD at each platform from local measurements and measurements received from neighbors (if any), and (b) computing movements by combining locally computed information theoretic forces with locally computed physicomimetic forces.

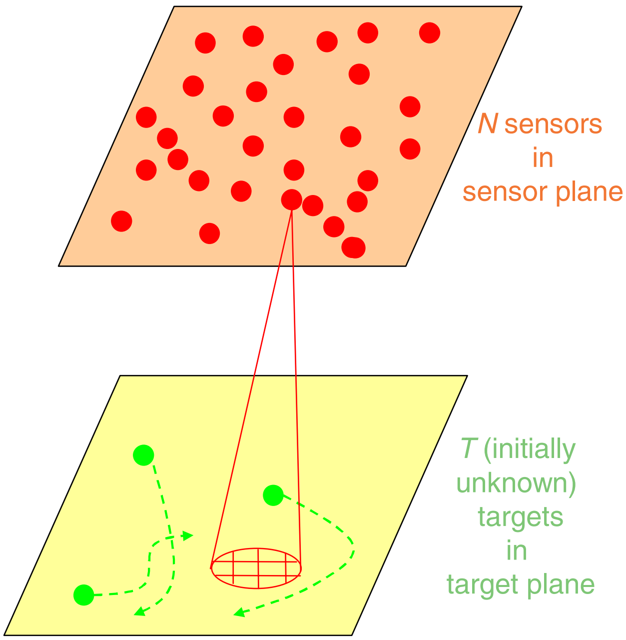


Fig. 1. The model problem. The network is to determine the number and state of a group of moving ground targets. Each node stares directly down and has its position controlled by the information theoretic method described here.

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 of platforms with 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, simulation of 500 platforms requires significantly longer than real-time (500 times longer). The centralized algorithm is significantly cheaper computationally, owing to the fact that only one JMPD must be estimated (rather than 500 local JMPDs). The communication burden is significantly increased, however. This simulation illustrates surveillance in a similar model problem, and also compares the performance of the algorithm with an algorithm using only the physicomimetic force and one with only the information gain force. Simulations show that the algorithm that combines forces significantly outperforms algorithms based on the constituent forces alone.

A. A Small Number of 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 $5000m \times 5000m$ surveillance area that contains 10 moving ground targets (the number of targets, 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

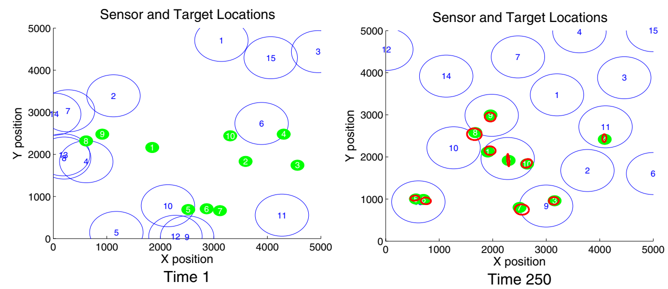


Fig. 2. The random positioning of the 15 platforms at initialization (l) and after some time (r). Platform position is given by the number and its field of view is described by the circle surrounding the number. The true position of each of the ten moving ground targets is shown by the numbered circles. Qualitatively, after some time, the platforms have preferentially aligned themselves over the targets while still allocating some network resources to look for new targets.

the number of targets and their individual states is learned as quickly and accurately as possible.

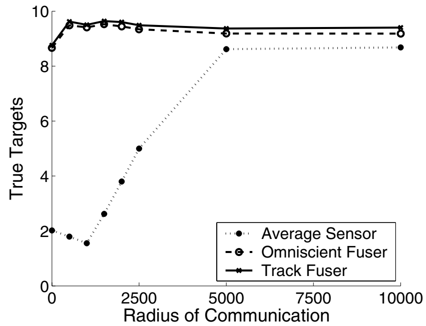
Target trajectories for the simulation come directly from a set of recorded data based on GPS measurements of vehicle positions over time collected as part of a battle training exercise at the Army's National Training Center. Targets routinely come within sensor cell resolution (i.e., cross). Persistent targets are modeled in the JMPD time 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. The model problem setup is illustrated in Figure 1.

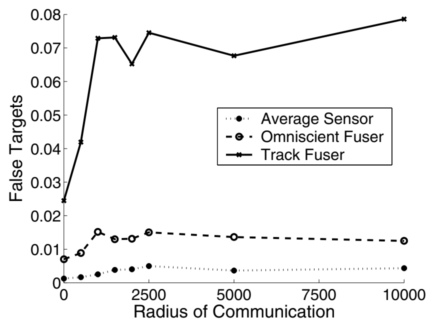
When measuring a cell, the imager returns either a 0 (no detection) or a 1 (detection) which is governed by a probability of detection (p_d) and a per-cell false alarm rate (p_f). Both are assumed to be temporally and spatially constant . The signal to noise ratio (SNR) links these values together. The 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 $SNR_{max} = 16dB$, $p_f = 0.01$, and use $p_d = p_f^{\frac{1}{1+SNR}}$ returns. When there are T targets in the same cell, the detection probability is $p_d(T) = p_f^{\frac{1}{1+SNR*T}}$.

Each platform computes a local estimate of the JMPD using measurements it has made and measurements received from neighbors. Platforms then use the joint constrained information theoretic optimization approximation described in the previous section to compute next best movements.

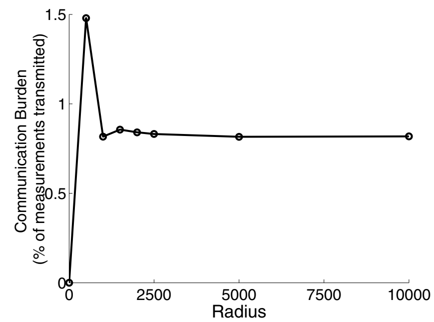
Figure 2 shows an initial (random) positioning of the 15 sensors and the position after some time. As can be seen from the figure, over time the sensors preferentially align themselves around the targets (which were discovered through repeated interrogation of the ground) while still allocating



(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.

Fig. 3. Monte Carlo performance results for the 15 sensor region surveillance application.

some resources to look for new targets.

2) Monte Carlo Simulation of Performance: Figure 3 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**, at the **Track Fuser**, and at the **(hypothetical) Omniscient Fuser**. The performance of the network is measured by the number of **True Targets** correctly found and **False Targets** incorrectly thought to exist. Additionally, we look at the **Communication Requirements** of the method in terms of the percent of measurements that each node transmits.

B. A Large Number of Low 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 ($SNR = 10dB$).

2) Emergent Behavior With Different Scheduling Methods: In Section IV, we cast the multiplatform information theoretic scheduling criteria as 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.

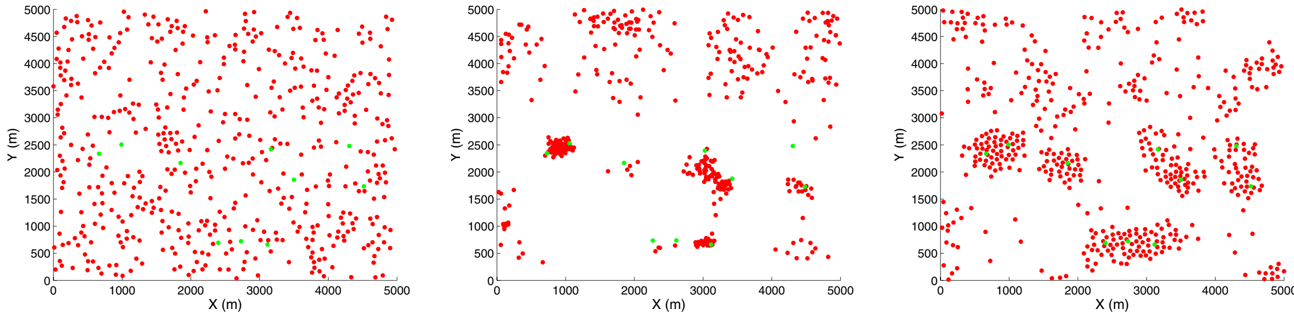
In this section, we illustrate how the combination of these forces promotes the correct platform behavior and that the individual forces themselves are not sufficient. Specifically, we compare both qualitatively and quantitatively the surveillance performance of a network of sensors with three different scheduling algorithms: (a) The proposed **Combination of Information Theoretic Forces and Physicomimetic Forces**, which provides a balance between information seeking behavior and explorative behavior and is connected directly with the optimal multiplatform scheduling method, (b) A purely **Information Theoretic Method**, which takes actions that maximize information gain (only), and (c) A purely **Physicomimetic Method**, which maintains separation between sensors using the repulsive force (only). Figure 4 shows steady-state platform positioning of 500 platforms under each method.

3) Monte Carlo Simulation of Performance: We again display the performance of the scheduling algorithm based on (a) the number of true targets detected, and (b) the number of false targets reported. Figure 5 shows the performance 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 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



(a) Steady-state positioning of platforms controlled by the physicomimetic force only.

(b) Steady-state positioning of platforms controlled by the information theoretic force only.

(c) Steady-state positioning of platforms controlled by the combination physicomimetic and information theoretic forces.

Fig. 4. 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.

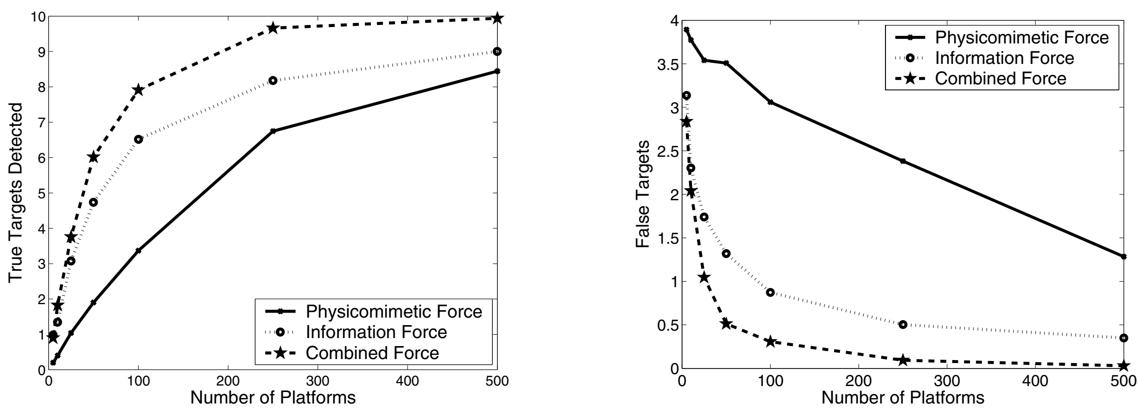


Fig. 5. Performance of the proposed method versus number of platforms in terms of true targets detected and false targets reported. For comparison purposes, the performance of each of the constituent forces (the physicomimetic force and the information theoretic force) are included. As can be seen in 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.

is a novel combination of particle filtering for nonparametric density estimation, information theoretic measures for comparing possible action sequences, and artificial physics for providing approximate cooperation between sensor nodes.

Future work in this area includes the extension of the methods to long-term (non-myopic) scheduling. In a manner analogous to multisensor scheduling, (naive) multi-step 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 to the multi-step setting, but has focussed mainly on the single platform setting.

References

[1] C. Kreucher, K. Kastella, and A. Hero, "A bayesian method for integrated multitarget tracking and sensor management," in *The Proceedings of The 6th International Conference on Information Fusion*, vol. 1, July 8 - 11 2003, pp. 704 – 711.

[2] ——, "Multitarget tracking using the joint multitarget probability density," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1396–1414, October 2005.

[3] M. Morelande, C. Kreucher, and K. Kastella, "A bayesian approach to multiple target detection and tracking," *to Appear in IEEE Transactions on Signal Processing*.

[4] C. Kreucher, K. Kastella, and A. Hero, "Information-based sensor management for multitarget tracking," in *Signal and Data Processing of Small Targets 2003*, ser. Proc. SPIE, O. E. Drummond, Ed., vol. 5204, August 3 - 8 2003, pp. 480 – 489.

[5] ——, "Sensor management using an active sensing approach," *Signal Processing*, vol. 85, no. 3, pp. 607 – 624, March 2005.

[6] C. Kreucher, A. Hero, and K. Kastella, "A comparison of task driven and information driven sensor management for target tracking," in *The 44th IEEE Conf. on Decision and Control*, December 12-15 2005, pp. 4004–4009.

[7] W. D. Spears, J. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," *Autonomous Robots*, vol. 17, no. 2-3, 2004.