Approximate Dynamic Programming for Communication-Constrained Sensor Network Management

Jason L. Williams, *Student Member, IEEE*, John W. Fisher, III, *Member, IEEE*, and Alan S. Willsky, *Fellow, IEEE*

Abstract—Resource management in distributed sensor networks is a challenging problem. This can be attributed to the fundamental tradeoff between the value of information contained in a distributed set of measurements versus the energy costs of acquiring measurements, fusing them into the conditional probability density function (pdf) and transmitting the updated conditional pdf. Communications is commonly the highest contributor among these costs, typically by orders of magnitude. Failure to consider this tradeoff can significantly reduce the operational lifetime of a sensor network. While a variety of methods have been proposed that treat a subset of these issues, the approaches are indirect and usually consider at most a single time step. In the context of object tracking with a distributed sensor network, we propose an approximate dynamic programming approach that integrates the value of information and the cost of transmitting data over a rolling time horizon. We formulate this tradeoff as a dynamic program and use an approximation based on a linearization of the sensor model about a nominal trajectory to simultaneously find a tractable solution to the leader node selection problem and the sensor subset selection problem. Simulation results demonstrate that the resulting algorithm can provide similar estimation performance to that of the common most informative sensor selection method for a fraction of the communication cost.

Index Terms—Adaptive estimation, dynamic programming, resource management, tracking.

I. INTRODUCTION

NETWORKS of intelligent sensors have the potential to provide unique capabilities for monitoring wide geographic areas through the intelligent exploitation of local computation (so-called in-network computing) and the judicious use of intersensor communication. In many sensor networks, energy is a dear resource to be conserved so as to prolong the network's operational lifetime. In addition, it is typically the case that the energy cost of communications is orders of magnitude greater than the energy cost of local computation [1], [2].

J. L. Williams and A. S. Willsky are with the Laboratory for Information and Decision System, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jlwilliams@alum.mit.edu; willsky@mit.edu).

J. W. Fisher, III, is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: fisher@csail.mit.edu).

Tracking moving objects is a common application in which the quantities of interest (i.e., kinematic state) are inferred largely from sensor measurements that are in proximity to the object (e.g., [3]). Consequently, local fusion of sensor data is sufficient for computing an accurate model of the object state and associated uncertainty, as captured by the conditional probability density function (pdf). This property, combined with the need to conserve energy, has led to a variety of approaches [4], [5] that effectively designate the responsibility of computing the conditional pdf to one sensor node (referred to as the *leader node*) in the network. Over time, the leader node changes dynamically as function of the kinematic state of the object. This leads to an inevitable tradeoff between the accuracy of the model, the cost of acquiring measurements, and the cost of propagating the model through the network. In this paper, we examine this tradeoff in the context of object tracking in distributed sensor networks. In doing so, we consider the aggregate cost over a rolling time horizon using an approximate dynamic programming approach. Our results show that, as compared with pure information-driven approaches, comparable tracking performance can be obtained at a fraction of the communications cost.

We consider a sensor network consisting of N_s sensors, in which the sensing model is assumed to be such that the measurement provided by the sensor is highly informative in the region close to the node and uninformative in regions far from the node. For the purpose of addressing the primary issue, trading off energy consumption for accuracy, we restrict ourselves to sensor resource planning issues associated with tracking a single object. While additional complexities certainly arise in the multi-object case (e.g., data association), they do not change the basic problem formulation or conclusions.

If the energy consumed by sensing and communication were unconstrained, then the optimal solution would be to collect and fuse the measurements provided by *all* sensors in the network. We consider a scheme in which, at each time step, a subset of sensors is selected to take a measurement and transmit to a sensor referred to as the leader node [4], which fuses the measurements with the prior conditional pdf and tasks sensors at the next time step. The questions that must be answered by the controller are how to select the subset of sensors at each point in time and how to select the leader node at each point in time.

The approach developed in Section III allows for optimization of estimation performance subject to a constraint on expected communication cost or minimization of communication cost subject to a constraint on expected estimation performance.

The controller uses a dual problem formulation to adaptively utilize multiple sensors at each time step, incorporating a subgradient update step to adapt the dual variable (Section III-H), and introducing a heuristic cost-to-go in the terminal cost to avoid anomalous behavior (Section III-I). Our dual problem formulation is closely related to [6] and provides an approximation that extends the Lagrangian relaxation approach to problems involving sequential replanning. Other related work includes [7], which suggests incorporation of sensing costs and estimation performance into a unified objective without adopting the constrained optimization framework that we utilize, and [8], which adopts a constrained optimization framework without incorporating estimation performance and sensing cost into a unified objective, a structure that results in a major computational savings for our approach.

Preliminary versions of the work in Section III were presented in [9] and [10], while the discussion in Section IV is a generalization of [11].

II. PROBLEM FORMULATION

The tracking problem naturally fits into the Bayesian state estimation formulation such that the role of the sensor network is to maintain a representation of the conditional pdf of the object state (i.e., position, velocity, etc.) conditioned on the measurements.

A. Object Dynamics and Sensor Models

In order to be concrete, we now discuss specific object dynamics and sensor measurement models. However, we emphasize that the underlying principles have general applicability. Denoting $\boldsymbol{x}_k \in \mathcal{X}$ as the state of the object (or "object state") at time k, we assume that object dynamics evolve according to a linear Gaussian model, as follows:

$$\boldsymbol{x}_{k+1} = \mathbf{F}\boldsymbol{x}_k + \boldsymbol{w}_k \qquad (1)$$

where $\boldsymbol{w}_k \sim \mathcal{N}\{\boldsymbol{w}_k; \mathbf{0}, \mathbf{Q}\}$[1] is a white Gaussian noise process, and $\mathbf{F}$ and $\mathbf{Q}$ are known matrices. For the simulations in this paper, we track position and velocity in two dimensions ($\boldsymbol{x}_k = [p_x \ v_x \ p_y \ v_y]^T$), where velocity is modeled as a random walk with constant diffusion strength q (independently in each dimension), and position is the integral of velocity. Denoting the sampling interval as T, the corresponding discrete-time model is

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad \mathbf{Q} = q \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix}. \qquad (2)$$

Denoting the measurement taken by sensor $s \in \mathcal{S} = \{1 : N_s\}$ (where N_s is the number of sensors) at time k as $\boldsymbol{z}_k^s$, a nonlinear measurement model is assumed, as follows:

$$\boldsymbol{z}_k^s = \boldsymbol{h}(\boldsymbol{x}_k, s) + \boldsymbol{v}_k^s \qquad (3)$$

[1]We use the notation $\boldsymbol{w}_k \sim \mathcal{N}\{\boldsymbol{w}_k; \mathbf{0}, \mathbf{Q}\}$ as short-hand for $p(\boldsymbol{w}_k) = \mathcal{N}\{\boldsymbol{w}_k; \mathbf{0}, \mathbf{Q}\}$, where $\mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}, \mathbf{P}\} = |2\pi \mathbf{P}|^{-1/2} \exp\{-0.5(\boldsymbol{x} - \boldsymbol{\mu})^T \mathbf{P}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\}$.

where $\boldsymbol{v}_k^s \sim \mathcal{N}\{\boldsymbol{v}_k^s; \mathbf{0}, \mathbf{R}^s\}$ is a white Gaussian noise process, independent of $\boldsymbol{w}_k \forall k$ and of $\boldsymbol{v}_k^j, j \neq s \forall k$. $\mathbf{R}^s$ is a known matrix for each s, and $\boldsymbol{h}(\cdot, s)$ is a known, vector-valued function for each s. For the simulations in this paper, we set the measurement model to a quasi-range measurement

$$h(\boldsymbol{x}_k, s) = \frac{a}{\|\mathbf{L}\boldsymbol{x}_k - \boldsymbol{y}^s\|_2^2 + b} \qquad (4)$$

where $\mathbf{L}$ is the matrix that extracts the position of the object from the object state (such that $\mathbf{L}\boldsymbol{x}_k$ is the location of the object), and $\boldsymbol{y}^s$ is the location of the sth sensor (which is assumed to be known, e.g., through the calibration procedure as described in [12]). The constants a and b can be tuned to model the signal-to-noise ratio (SNR) of the sensor, and the rate at which the SNR decreases as distance increases. The measurement has additive Gaussian noise as per (3), with variance R. The information provided by the measurement reduces as the range increases due to the nonlinearity.

The measurement function $h(\cdot, s)$ has sufficient smoothness that, in a small vicinity around a nominal point $\boldsymbol{x}^0$, it can be approximated as a first-order Taylor series truncation, as follows:

$$\boldsymbol{z}_k^s \approx \boldsymbol{h}(\boldsymbol{x}^0, s) + \mathbf{H}^s(\boldsymbol{x}^0)(\boldsymbol{x}_k - \boldsymbol{x}^0) + \boldsymbol{v}_k^s \qquad (5)$$

$$\mathbf{H}^s(\boldsymbol{x}^0) = \nabla_{\boldsymbol{x}} \boldsymbol{h}(\boldsymbol{x}, s)|_{\boldsymbol{x} = \boldsymbol{x}^0}. \qquad (6)$$

The linearization of the model in (4), which will be used in Section III to reduce the complexity of the planning process, is

$$\mathbf{H}^s(\boldsymbol{x}^0) = \frac{-2a}{(\|\mathbf{L}\boldsymbol{x}^0 - \boldsymbol{y}^s\|_2^2 + b)^2}(\mathbf{L}\boldsymbol{x}^0 - \boldsymbol{y}^s)^T \mathbf{L}. \qquad (7)$$

We denote by $\boldsymbol{z}_{0:k}$ the history of all measurements received up to an including time k, i.e., $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_k\}$.

This model will be utilized in the simulations in Section V. Although the following exposition is independent of the specific model, it has been specified now to aid the clarity and concreteness.

B. Estimation

The planning algorithm we present in Section III may be applied alongside a wide variety of estimation methods. Here, we discuss the estimator that we use, which is well suited to the challenging sensor models that arise in sensor networks. The motivation for sensor networks is to utilize many small sensors with limited local sensing capability to provide surveillance of a larger region. Because sensors focus on their local region, the nonlinearity in a measurement model such as the quasi-range measurement of (4) is significant, and substantial multimodality can result. Accordingly, we utilize a particle filter approximation for the dynamic state estimation, whereby the conditional pdf of object state $\boldsymbol{x}_k$ conditioned on measurements received up to and including time k, $\boldsymbol{z}_{0:k}$, $p(\boldsymbol{x}_k | \boldsymbol{z}_{0:k}) \in \mathcal{P}(\mathcal{X})$,[2] is approximated through a set of N_p weighted samples

$$p(\boldsymbol{x}_k | \boldsymbol{z}_{0:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_k^i\right). \qquad (8)$$

[2]We denote by $\mathcal{P}(\mathcal{X})$ the set of probability density functions on the set $\mathcal{X}$.

To calculate the same distribution at the next time step (for inference), $p(\boldsymbol{x}_{k+1}|\boldsymbol{z}_{0:k+1})$, we apply the commonly used approximate Sequential Importance Sampling (SIS) algorithm [13] with resampling at each step. Under this algorithm, for each previous sample $\boldsymbol{x}_k^i$, we draw a new sample at the next time step $\boldsymbol{x}_{k+1}$ from the distribution $q(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i, \boldsymbol{z}_{k+1})$, which results from the linearization of the measurement model for $\boldsymbol{z}_{k+1}$ (3) about the point $\mathbf{F}\boldsymbol{x}_k^i$, as described in (5). This distribution can be obtained using the extended Kalman filter equations: the Dirac delta function $\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i)$ at time k will diffuse to give

$$p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i\right) = \mathcal{N}\left(\boldsymbol{x}_{k+1}; \mathbf{F}\boldsymbol{x}_k^i; \mathbf{Q}\right) \qquad (9)$$

at time $(k+1)$. This distribution can be updated using the extended Kalman filter update equation [14] to obtain

$$q\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i, \boldsymbol{z}_{k+1}\right) = \mathcal{N}\left(\boldsymbol{x}_{k+1}; \hat{\boldsymbol{x}}_{k+1}^i, \mathbf{P}_{k+1}^i\right) \qquad (10)$$

where

$$\hat{\boldsymbol{x}}_{k+1}^i = \mathbf{F}\boldsymbol{x}_k^i + \mathbf{K}_{k+1}^i\left[\boldsymbol{z}_{k+1} - \boldsymbol{h}\left(\mathbf{F}\boldsymbol{x}_k^i, s\right)\right] \qquad (11)$$

$$\mathbf{P}_{k+1}^i = \mathbf{Q} - \mathbf{K}_{k+1}^i\mathbf{H}^s\left(\mathbf{F}\boldsymbol{x}_k^i\right)\mathbf{Q} \qquad (12)$$

$$\mathbf{K}_{k+1}^i = \mathbf{Q}\left\{\mathbf{H}^s\left(\mathbf{F}\boldsymbol{x}_k^i\right)\right\}^T$$
$$\cdot \left[\mathbf{H}^s\left(\mathbf{F}\boldsymbol{x}_k^i\right)\mathbf{Q}\left\{\mathbf{H}^s\left(\mathbf{F}\boldsymbol{x}_k^i\right)\right\}^T + \mathbf{R}^s\right]^{-1}. \qquad (13)$$

A new particle $\boldsymbol{x}_{k+1}^i$ is drawn from the distribution in (10), and weighted by w_{k+1}^i, calculated by

$$w_{k+1}^i = cw_k^i \frac{p\left(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}^i\right)p\left(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i\right)}{q\left(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i, \boldsymbol{z}_{k+1}\right)} \qquad (14)$$

where c is the normalization constant necessary to ensure that $\sum_{i=1}^{N_p} w_{k+1}^i = 1$, and

$$p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}^i) = \mathcal{N}\{\boldsymbol{z}_{k+1}; \boldsymbol{h}(\boldsymbol{x}_{k+1}^i, s), \mathbf{R}^s\}.$$

The resulting approximation for the distribution of $\boldsymbol{x}_{k+1}$ conditioned on the measurements $\boldsymbol{z}_{0:k+1}$ is

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{z}_{0:k+1}) \approx \sum_{i=1}^{N_p} w_{k+1}^i \delta\left(\boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^i\right). \qquad (15)$$

At any point in time, a Gaussian representation can be moment-matched to the particle distribution by calculating the mean and covariance

$$\boldsymbol{\mu}_k = \sum_{i=1}^{N_p} w_k^i \boldsymbol{x}_k^i, \mathbf{P}_k = \sum_{i=1}^{N_p} w_k^i\left(\boldsymbol{x}_k^i - \boldsymbol{\mu}_k\right)\left(\boldsymbol{x}_k^i - \boldsymbol{\mu}_k\right)^T. \qquad (16)$$

C. Communications

We assume that any sensor node can communicate with any other sensor node in the network and that the cost of these communications is known at every sensor node (although in practice, this will only be required within a small region around each

node). In our simulations, the cost (per bit) of direct communication between two nodes is modeled as being proportional to the square distance between the two sensors, as folows:

$$\tilde{C}_{ij} \propto \|\boldsymbol{y}^i - \boldsymbol{y}^j\|_2^2. \qquad (17)$$

Communications between distant nodes can be performed more efficiently using a multihop scheme, in which several sensors relay the message from source to destination. Hence, we model the cost of communicating between nodes i and j, C_{ij}, as the length of the shortest path between i and j, using the distances from (17) as arc lengths

$$C_{ij} = \sum_{k=1}^{n_{ij}} \tilde{C}_{i_{k-1}i_k} \qquad (18)$$

where $\{i_0, \ldots, i_{n_{ij}}\}$ is the shortest path from node $i = i_0$ to node $j = i_{n_{ij}}$. The shortest path distances can be calculated using any shortest path algorithm, such as deterministic dynamic programming or label correcting methods [15]. We assume that the complexity of the probabilistic model (i.e., the number of bits required for transmission) is fixed at B_p bits, such that the energy required to communicate the model from node i to node j is $B_p C_{ij}$. The content of these transmissions will depend on the estimation scheme used; for particle filters, one may employ an efficient compression scheme such as the one described in [16]. The number of bits in a measurement is denoted as B_m such that the energy required to transmit a measurement from node i to node j is $B_m C_{ij}$. These costs may be amended to incorporate the cost of activating the sensor, taking the measurement, the expected number of retransmissions required, etc., without changing the structure of the solution.

D. Estimation Performance Objective Functions

As discussed in Section I, the role of the sensor manager in a sensor network tracking problem is to trade off estimation performance against the energy consumed in obtaining that performance. The first task in optimizing the estimation performance of a system is to decide upon an objective function which measures that performance. Recent research [17], [18] has demonstrated the effectiveness of conditional entropy as an objective function for sensor management in object tracking applications. In the following sections, we briefly outline some of the desirable properties which entropy possesses, and its relationship to other estimation performance objectives.

Conditional entropy is defined as [19]

$$H\left(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^s\right) = -\int p\left(\boldsymbol{z}_k^s|\boldsymbol{z}_{0:k-1}\right)$$
$$\cdot \int p\left(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^s\right) \log p\left(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^s\right) d\boldsymbol{x}_k d\boldsymbol{z}_k^s. \qquad (19)$$

Although not explicit in our notation in (19), we condition on the *value* of the past measurements, $\boldsymbol{z}_{0:k-1}$ (which have already been realized), and on the *random variable* corresponding to the new measurement $\boldsymbol{z}_k^s$ (which has not yet been realized). When conditioning on a random variable, we must take an expectation over the possible values that the measurement may ultimately

assume. Throughout the following, we assume that the previously received measurements $z_{0:k-1}$ have been realized; hence, we condition on their value, whereas the candidate measurements $z_j, j \geq k$ have not; hence, we condition on the random variables.

1) Single Time Step, Single Sensor: If we choose to utilize the sensor $s_k \in \mathcal{S}$ whose measurement minimizes the expected posterior entropy, the resulting algorithm (e.g., [4]) would be

$$s_k = \arg\min_{s \in \mathcal{S}} H\left(\boldsymbol{x}_k | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^s\right). \qquad (20)$$

The mutual information between the object state $\boldsymbol{x}_k$ and measurement $\boldsymbol{z}_k^s$ conditioned on the previous measurement history is defined as the expected reduction in entropy of object state due to conditioning on the new measurement or, equivalently, the expected reduction in entropy of $\boldsymbol{z}_k^s$ due to conditioning on the object state, as follows [19]:

$$I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^s | \boldsymbol{z}_{0:k-1}\right) = H(\boldsymbol{x}_k | \boldsymbol{z}_{0:k-1}) - H\left(\boldsymbol{x}_k | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^s\right) \quad (21)$$
$$= H\left(\boldsymbol{z}_k^s | \boldsymbol{z}_{0:k-1}\right) - H\left(\boldsymbol{z}_k^s | \boldsymbol{x}_k\right). \quad (22)$$

The conditioning on previous measurements in the second term of (22) is discarded as we assume that the current measurement is independent of previous measurements conditioned on the current object state. Because the first term in (21) is independent of the sensor s, it is clear that selecting the sensor whose measurement minimizes the conditional entropy of the object state is equivalent to selecting the sensor whose measurement has the highest mutual information with the object state, i.e., (20) is equivalent to [20]

$$s_k = \arg\max_{s \in \mathcal{S}} I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^s | \boldsymbol{z}_{0:k-1}\right). \qquad (23)$$

The entropy of an n-dimensional multivariate Gaussian distribution with covariance $\mathbf{P}$ is equal to $(1/2)\log|2\pi e\mathbf{P}| = (n/2)\log 2\pi e + (1/2)\log|\mathbf{P}|$. Thus, under linear-Gaussian assumptions, minimizing conditional entropy is equivalent to minimizing the determinant of the posterior covariance, or the volume of the uncertainty hyper-ellipsoid.

2) Single Time Step, Multiple Sensors: Conceptually, the single-sensor selection methods discussed above can be extended to choosing a subset $\mathcal{S}_k \subseteq \mathcal{S}$ of n sensors at one time, e.g.,

$$\mathcal{S}_k = \arg\max_{\mathcal{S}_k \subseteq \mathcal{S}, |\mathcal{S}_k| = n} I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k} | \boldsymbol{z}_{0:k-1}\right) \qquad (24)$$

where $\boldsymbol{z}_k^{\mathcal{S}_k}$ denotes the collection of measurements corresponding to the sensors in the set $\mathcal{S}_k$. As in the single-sensor case above, this is equivalent to minimizing the posterior entropy of the object state conditioned on the new subset of measurements. Because of the additivity of mutual information, we can decompose the objective in (24) into the sum of the gains due to each measurement in the subset $\mathcal{S}_k$. Denoting by $\mathcal{S}_k^j$ the jth element of $\mathcal{S}_k$ (choosing an arbitrary ordering), and

by $\mathcal{S}_k^{1:j-1}$ the set containing the first $(j-1)$ elements of $\mathcal{S}_k$, the mutual information can be decomposed as

$$I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k} | \boldsymbol{z}_{0:k-1}\right) = \sum_{j=1}^{|\mathcal{S}_k|} I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k^j} | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^{\mathcal{S}_k^{1:j-1}}\right). \qquad (25)$$

The complexity of the optimization in (24) increases combinatorially with n, since one must consider all n-element subsets. Although the decomposition in (25) does not directly reduce this complexity (indeed the expression is equivalent), the additive form yields a convenient approximation, in which we select the jth sensor to maximize the jth term in the sum of (25). This form will be exploited extensively in Section III-F.

3) Multiple Time Steps, Multiple Sensors: Sensor management strategies that select the action that minimizes the conditional entropy or that maximizes the mutual information over the next time step are sometimes referred to as *greedy* or *myopic*. Situations can arise (e.g., [21]) in which alternative strategies have poorer performance in the next time step but better performance over several time steps. In practice, greedy schemes have been seen to provide good performance when estimation quality is the only objective. When energy is limited, it must also be incorporated into the objective, necessitating additional planning.

Dynamic sensor selection problems that span multiple time steps fit naturally into the framework of dynamic programming (DP) [15]. The DP methodology provides a conceptual mechanism for deriving a control policy that optimizes the system performance over a particular planning horizon, as quantified through an objective function. In order to be able to use efficient approximation methods, it is often desirable that the objective function value be expressed as the sum of costs or rewards resulting from each decision stage. By selecting as our objective (to be minimized) the joint conditional entropy over an N-step planning horizon, $H(\boldsymbol{x}_{k:k+N-1} | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^{\mathcal{S}_k}, \ldots, \boldsymbol{z}_{k+N-1}^{\mathcal{S}_{k+N-1}})$, one may apply the arguments above to find that this can indeed be decomposed into an additive form, since

$$H\left(\boldsymbol{x}_{k:k+N-1} | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^{\mathcal{S}_k}, \ldots, \boldsymbol{z}_{k+N-1}^{\mathcal{S}_{k+N-1}}\right)$$
$$= -\sum_{i=k}^{k+N-1} \sum_{j=1}^{|\mathcal{S}_i|} I\left(\boldsymbol{x}_i; \boldsymbol{z}_i^{\mathcal{S}_i^j} | \boldsymbol{z}_{0:i-1}, \boldsymbol{z}_i^{\mathcal{S}_i^{1:j-1}}\right)$$
$$+ H(\boldsymbol{x}_{k:k+N-1} | \boldsymbol{z}_{0:k-1}) \qquad (26)$$

and the final term in (26), which quantifies the uncertainty conditioned on measurements up to time $(k-1)$ (prior to current planning interval), is independent of the subsets of sensors selected (and can thus be excluded, as discussed above (23)). Hence, by selecting the expression in (25) as our per-stage reward, the resulting controller will minimize the joint conditional entropy over an N-step planning horizon.

III. CONSTRAINED DYNAMIC PROGRAMMING FORMULATION

The sensor network object tracking problem involves an inherent tradeoff between performance and energy expenditure. One way of incorporating both estimation performance and

communication cost into an optimization procedure is to optimize one of the quantities subject to a constraint on the other. In the development which follows, we provide a framework which can be used to either maximize the information obtained from the selected measurements subject to a constraint on the expected communication cost, or to minimize the communication cost subject to a constraint on the estimation quality. This can be formulated as a constrained Markov decision process (MDP) [6], [22]. Similarly to imperfect state information problems[3] [15], the *dynamic programming* state is the pdf of *object* state conditioned on previous controls and measurements. Throughout the following, we denote the conditional belief state as $\mathbb{X}_k \triangleq p(\boldsymbol{x}_k | \boldsymbol{z}_{0:k-1}) \in \mathcal{P}(\mathcal{X})$[4]; the decision state at time k will consist of $\mathbb{X}_k$, augmented with the leader node at the previous time step $l_{k-1} \in \mathcal{S}$. The control at each time is denoted as $u_k = (l_k, \mathcal{S}_k)$, where $l_k \in \mathcal{S}$ is the leader node at time k and $\mathcal{S}_k \subseteq \mathcal{S}$ is the subset of sensors activated at time k.

A. Constrained Dynamic Programming

We seek to minimize the per-stage cost over an N-step rolling horizon, i.e., at time k, we minimize the cost incurred in the planning horizon involving steps $(k : k + N - 1)$. Denoting by $\mu_k(\mathbb{X}_k, l_{k-1})$ the control policy for time k, and by $\pi_k = \{\mu_k, \ldots, \mu_{k+N-1}\}$ the set of policies for the next N time steps, we seek the policy corresponding to the optimal solution to the constrained minimization problem, as follows:

$$\min_{\pi} \mathrm{E}\left[\sum_{i=k}^{k+N-1} g\left(\mathbb{X}_i, l_{i-1}, \mu_i(\mathbb{X}_i, l_{i-1})\right) \right]$$
$$\text{s.t. } \mathrm{E}\left[\sum_{i=k}^{k+N-1} G\left(\mathbb{X}_i, l_{i-1}, \mu_i(\mathbb{X}_i, l_{i-1})\right) \right] \leq M \quad (27)$$

where $g(\mathbb{X}_k, l_{k-1}, u_k)$ is the per-stage cost and $G(\mathbb{X}_k, l_{k-1}, u_k)$ is the per-stage contribution to the additive constraint function. Both of these will be defined in Section III-B for the communication-constrained formulation, and in Section III-C for the entropy-constrained formulation. We address the constraint through a Lagrangian relaxation, a common approximation method for discrete optimization problems, by defining the Lagrangian function

$$L_k(\mathbb{X}_k, l_{k-1}, \pi_k, \lambda) = \mathrm{E}\left[\sum_{i=k}^{k+N-1} g\left(\mathbb{X}_i, l_{i-1}, \mu_i(\mathbb{X}_i, l_{i-1})\right) \right.$$
$$\left. + \lambda \left(\sum_{i=k}^{k+N-1} G\left(\mathbb{X}_i, l_{i-1}, \mu_i(\mathbb{X}_i, l_{i-1})\right) - M \right) \right]. \quad (28)$$

[3]Strictly speaking, the problem is not a partially observed Markov decision process (POMDP) in the traditional sense, as the per-stage cost will be defined directly as a function of the pdf ($p(\boldsymbol{x}_k | \boldsymbol{z}_{0:k-1})$), rather than indirectly as an expectation of a function of the underlying state ($\boldsymbol{x}_k$). Because the per-stage cost is a nonlinear function of the pdf, it cannot be written as an expectation of a function of the underlying state. Furthermore, since the resulting cost-to-go is neither concave nor convex (as a function of the pdf), traditional POMDP solution methods that rely on piecewise linear concavity are inapplicable, and it is necessary to develop new approximations, such as the one in this paper, that are tailored to the problem structure.

[4]Conditioning on previous control decisions is assumed throughout. In contrast to the convention of [15], the measurements at time k, $\boldsymbol{z}_k^{\mathcal{S}_k}$ are received after the control at time k, u_k has been applied, and the distribution $p(\boldsymbol{z}_k^{\mathcal{S}_k} | \boldsymbol{x}_k, u_k)$ depends upon the value of the control applied at time k.

The unconstrained optimization of the Lagrangian function over the policy π_k (which is the primal variable) yields the dual function[5]:

$$J_k^D(\mathbb{X}_k, l_{k-1}, \lambda) = \min_{\pi_k} L_k(\mathbb{X}_k, l_{k-1}, \pi_k, \lambda). \quad (29)$$

We then solve the dual optimization problem involving this function:

$$J_k^L(\mathbb{X}_k, l_{k-1}) = \max_{\lambda \geq 0} J_k^D(\mathbb{X}_k, l_{k-1}, \lambda). \quad (30)$$

Note that the optimization problem in the dual function $J_k^D(\mathbb{X}_k, l_{k-1}, \lambda)$ takes the form of an unconstrained dynamic program with a modified per-stage cost, as follows:

$$\bar{g}(\mathbb{X}_k, l_{k-1}, u_k, \lambda) = g(\mathbb{X}_k, l_{k-1}, u_k) + \lambda G(\mathbb{X}_k, l_{k-1}, u_k). \quad (31)$$

The optimization of the dual problem provides a lower bound to the minimum value of the original constrained problem; the presence of a duality gap is possible since the optimization space is discrete. The size of the duality gap is given by the expression $\lambda \mathrm{E}[\sum_i G(\mathbb{X}_i, l_{i-1}, \mu_i(\mathbb{X}_i, l_{i-1})) - M]$, where $\pi_k = \{\mu_i(\cdot, \cdot)\}_{i=k:k+N-1}$ is the policy attaining the minimum in (29) for the value of λ attaining the maximum in (30). If it happens that the optimal solution produced by the dual problem has no duality gap, then the resulting solution is also the optimal solution of the original constrained problem. This can occur in one of two ways: either the Lagrange multiplier λ is zero, such that the solution of the unconstrained problem satisfies the constraint, or the solution yields a result for which the constraint is tight. If a duality gap exists, a better solution may exist satisfying the constraint; however, the solution returned would have been optimal if the constraint level had been lower, such that the constraint was tight.

Conceptually, the dual problem in (30) can be solved using a subgradient method [23]. The following expression can be seen to be a supergradient[6] of the dual objective:

$$S(\mathbb{X}_k, l_{k-1}, \pi_k, \lambda) = \mathrm{E}\left[\sum_i G(\mathbb{X}_i, l_{i-1}, \mu_i(\mathbb{X}_i, l_{i-1})) - M \right]. \quad (32)$$

In other words, $S(\mathbb{X}_k, l_{k-1}, \pi_k, \lambda) \in \partial J_k^D(\mathbb{X}_k, l_{k-1}, \lambda)$, where ∂ denotes the superdifferential, i.e., the set of all supergradients. The subgradient method operates according to the same principle as a gradient search, iteratively stepping in the direction of a subgradient with a decreasing step size. The practical implementation of the method is discussed in Section III-H.

The use of the Lagrangian relaxation to solve the constrained dynamic program is similar to [6]. The practical implementation differs substantially since we utilize a rolling horizon rather than a fixed finite horizon, and since the dynamic program in the dual function cannot be solved exactly.

[5]Note that $\mathbb{X}_k$ and l_{k-1} define the initial state at time k: they are not variables subject to optimization.

[6]Since we are maximizing a nondifferentiable concave function rather than minimizing a nondifferentiable convex function, subgradients are replaced by supergradients.

B. Constrained Communication Formulation

Following the discussion in Section II-D, we utilize mutual information as our objective and define the per-stage cost, as follows:

$$g(\mathbb{X}_k, l_{k-1}, u_k) = -I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k} | \boldsymbol{z}_{0:k-1}\right) \qquad (33)$$

$$= -\sum_{j=1}^{|\mathcal{S}_k|} I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k^j} | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^{\mathcal{S}_k^{1:j-1}}\right) \quad (34)$$

where $\mathcal{S}_k^j$ is the jth element of $\mathcal{S}_k$ and $\mathcal{S}_k^{1:j-1}$ is the set containing the first $(j-1)$ elements of $\mathcal{S}_k$. We choose the per-stage constraint contribution $G(\mathbb{X}_k, l_{k-1}, u_k)$ to be such that the expected communication cost over the next N time steps is constrained:

$$G(\mathbb{X}_k, l_{k-1}, u_k) = \left[B_p C_{l_{k-1}l_k} + \sum_{j \in \mathcal{S}_k} B_m C_{l_k j} \right]. \qquad (35)$$

Substituting the per-stage cost and constraint function into (29), the unconstrained optimization in the dual function (for a particular value of the Lagrange multiplier λ) can be solved conceptually using the recursive dynamic programming equation:

$$J_i^D(\mathbb{X}_i, l_{i-1}, \lambda) = \min_{u_i} \left\{ \bar{g}(\mathbb{X}_i, l_{i-1}, u_i, \lambda) \right.$$
$$\left. + \underset{\mathbb{X}_{i+1}|\mathbb{X}_i, u_i}{\mathrm{E}} J_{i+1}^D(\mathbb{X}_{i+1}, l_i, \lambda) \right\} \quad (36)$$

for time indexes $i \in \{k : k + N - 1\}$, terminated by $J_{k+N}^D(\mathbb{X}_{k+N}, l_{k+N-1}, \lambda) = -\lambda M$. The belief state at the next time $\mathbb{X}_{i+1}$ is calculated using the recursive Bayes update described in Section II-B. The augmented per-stage cost combines the information gain and communication cost in a single quantity, using a Lagrange multiplier, as follows:

$$\bar{g}(\mathbb{X}_k, l_{k-1}, u_k, \lambda) = -\sum_{j=1}^{|\mathcal{S}_k|} I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k^j} | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^{\mathcal{S}_k^{1:j-1}}\right)$$
$$+ \lambda \left[B_p C_{l_{k-1}l_k} + \sum_{j \in \mathcal{S}_k} B_m C_{l_k j} \right]. \qquad (37)$$

This incorporation of the constraint terms into the per-stage cost is a key step, which allows the greedy approximation described in Sections III-F and III-G to capture the tradeoff between estimation quality and communication cost.

C. Constrained Entropy Formulation

The formulation above provides a means of optimizing the information obtained subject to a constraint on the communication energy expended; there is also a closely related formulation that optimizes the communication energy subject to a constraint on the entropy of probabilistic model of object state. The cost per stage is set to the communication cost expended by the control decision, as follows:

$$g(\mathbb{X}_k, l_{k-1}, u_k) = B_p C_{l_{k-1}l_k} + \sum_{j \in \mathcal{S}_k} B_m C_{l_k j}. \qquad (38)$$

We commence by formulating a constraint function on the joint entropy of the state of the object over each time in the planning horizon, as follows:

$$\mathrm{E}\{H(\boldsymbol{x}_{k:k+N-1} | \boldsymbol{z}_{0:k+N-1})\} \leq H_{\max}. \qquad (39)$$

Manipulating this expression using (26), we obtain

$$-\mathrm{E}\left\{ \sum_{i=k}^{k+N-1} \sum_{j=1}^{|\mathcal{S}_i|} I\left(\boldsymbol{x}_i; \boldsymbol{z}_i^{\mathcal{S}_i^j} | \boldsymbol{z}_{0:i-1}, \boldsymbol{z}_i^{\mathcal{S}_i^{1:j-1}}\right) \right\}$$
$$\leq H_{\max} - H(\boldsymbol{x}_{k:k+N-1} | \boldsymbol{z}_{0:k-1}) \quad (40)$$

from which we set $M = H_{\max} - H(\boldsymbol{x}_{k:k+N-1} | \boldsymbol{z}_{0:k-1})$,[7] and

$$G(\mathbb{X}_k, l_{k-1}, u_k) = -\sum_{j=1}^{|\mathcal{S}_k|} I\left(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k^j} | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^{\mathcal{S}_k^{1:j-1}}\right). \quad (41)$$

Following the same procedure as described previously, the elements of the information constraint in (40) can be integrated into the per-stage cost, resulting in a formulation that is identical to (37), except that the Lagrange multiplier is on the mutual information term, rather than the communication cost terms.

D. Evaluation Through Monte Carlo Simulation

The constrained dynamic program described above has an infinite state space (the space of probability distributions over object state); hence, it cannot be evaluated exactly. The following sections describe a series of approximations that are applied to obtain a practical implementation.

Conceptually, the dynamic program of (36) could be approximated by simulating sequences of measurements for each possible sequence of controls. There are $N_s 2^{N_s}$ possible controls at each time step, corresponding all possible selections of leader node and subsets of sensors to activate. The complexity of the simulation process is formidable: to evaluate $\bar{J}_k^D(\mathbb{X}_k, l_{k-1}, \lambda)$ for a given DP state and control, we draw a set of N_p samples of the set of measurements $\boldsymbol{z}_k^{\mathcal{S}_k}$ from the distribution $p(\boldsymbol{z}_k^{\mathcal{S}_k} | \boldsymbol{z}_{0:k-1})$ derived from $\mathbb{X}_k$, and evaluate the cost-to-go one step later $\bar{J}_{k+1}^D(\mathbb{X}_{k+1}, l_k, \lambda)$ corresponding to the DP state resulting from each set of measurements. The evaluation of each cost-to-go one step later will yield the same branching. A tree structure develops, where for each previous leaf of the tree, $N_s 2^{N_s} N_p$ new leaves (samples) are drawn, such that the computational complexity increases as $O(N_s^N 2^{N_s N} N_p^N)$ as the tree depth N (i.e., the planning horizon) increases, as illustrated in Fig. 1. Such an approach quickly becomes intractable even for a small number of sensors (N_s) and simulated measurement samples (N_p); hence, we seek to exploit additional structure in the problem to find a computable approximate solution.

[7]In our implementation, we construct a new control policy at each time step by applying the approximate dynamic programming method described in the following section commencing from the current probabilistic model $\mathbb{X}_k$. At time step k, $H(\boldsymbol{x}_{k:k+N-1} | \boldsymbol{z}_{0:k-1})$ is a known constant; hence, the dependence on $\mathbb{X}_k$ is immaterial.
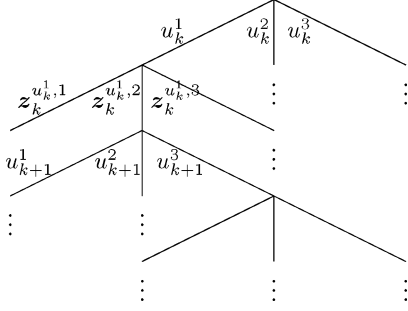
Fig. 1. Tree structure for evaluation of the dynamic program through simulation. At each stage, a tail subproblem is required to be evaluated for each new control and each simulated value of the resulting measurements.

E. Linearized Gaussian Approximation

If the dynamics and measurement models were linear and Gaussian, then the problem would be simplified substantially. The mutual information objective of a Gaussian pdf relates directly to its variance: if a linear measurement model holds

$$z_k^{\mathcal{S}_k} = \mathbf{H}_k^{\mathcal{S}_k} \boldsymbol{x}_k + \boldsymbol{v}_k^{\mathcal{S}_k} \tag{42}$$

and the *a priori* distribution of $\boldsymbol{x}_k$ is $\mathcal{N}\{\boldsymbol{x}_k; \boldsymbol{\mu}_k, \mathbf{P}_k\}$, then from (22)

$$I\left(\boldsymbol{x}_k; z_k^{\mathcal{S}_k} \mid \boldsymbol{z}_{0:k-1}\right) = H\left(z_k^{\mathcal{S}_k} \mid \boldsymbol{z}_{0:k-1}\right) - H\left(z_k^{\mathcal{S}_k} \mid \boldsymbol{x}_k\right).$$

Noting that $z_k^{\mathcal{S}_k} \mid \boldsymbol{x}_k \sim \mathcal{N}(z_k^{\mathcal{S}_k}; \mathbf{H}_k^{\mathcal{S}_k} \boldsymbol{x}_k, \mathbf{R}^{\mathcal{S}_k})$, we have

$$H\left(z_k^{\mathcal{S}_k} \mid \boldsymbol{x}_k\right) = \frac{1}{2} \log |2\pi e \mathbf{R}^{\mathcal{S}_k}|. \tag{43}$$

Similarly, with the linear measurement model $z_k^{\mathcal{S}_k} \mid \boldsymbol{z}_{0:k-1} \sim \mathcal{N}(z_k^{\mathcal{S}_k}; \mathbf{H}_k^{\mathcal{S}_k} \boldsymbol{\mu}_k, \mathbf{H}_k^{\mathcal{S}_k} \mathbf{P}_k \mathbf{H}_k^{\mathcal{S}_k\,T} + \mathbf{R}^{\mathcal{S}_k})$ [24], thus

$$H\left(z_k^{\mathcal{S}_k} \mid \boldsymbol{z}_{0:k-1}\right) = \frac{1}{2} \log \left| 2\pi e \left(\mathbf{H}_k^{\mathcal{S}_k} \mathbf{P}_k \mathbf{H}_k^{\mathcal{S}_k\,T} + \mathbf{R}^{\mathcal{S}_k}\right)\right|. \tag{44}$$

Collecting results, we obtain

$$I\left(\boldsymbol{x}_k; z_k^{\mathcal{S}_k} \mid \boldsymbol{z}_{0:k-1}\right) = \frac{1}{2} \log \left[\left| \mathbf{H}_k^{\mathcal{S}_k} \mathbf{P}_k \mathbf{H}_k^{\mathcal{S}_k\,T} + \mathbf{R}^{\mathcal{S}_k}\right| / |\mathbf{R}^{\mathcal{S}_k}|\right]. \tag{45}$$

Combining this with the result that the *a posteriori* covariance in a Kalman filter is independent of the measurement value, we see that future rewards depend only on the value of the control chosen (impacting $\mathbf{H}_k^{\mathcal{S}_k}$ and $\mathbf{R}^{\mathcal{S}_k}$, and hence the *a posteriori* covariance), and that they are invariant to the values of the measurements which result from applying the controls. Accordingly, the growth of the tree discussed in Section III-D is reduced to $O(N_s^N 2^{N_s N})$ with the horizon length N, rather than $O(N_s^N 2^{N_s N} N_p^N)$.

While this is a useful result, its applicability to this problem is not immediately clear, as the measurement model of interest is nonlinear, as discussed in Section II-A. However, let us suppose that the measurement model can be approximated by linearizing about a nominal state trajectory. If the strength of the

dynamics noise is relatively low and the planning horizon length is relatively short (such that deviation from the nominal trajectory is small), then such a linearization approximation may provide adequate fidelity for *planning* of future actions (this approximation is *not* utilized for inference: the SIS algorithm of Section II-B is used with the nonlinear measurement function to maintain the probabilistic model). To obtain the linearization, we suppose that the *a priori* distribution of object state at time k is $\mathcal{N}(\boldsymbol{x}_k; \boldsymbol{\mu}_k, \mathbf{P}_k)$; in practice, we moment-match a Gaussian distribution to the current particle distribution through (16). We then calculate the nominal trajectory as the mean at each of the following N steps:

$$\boldsymbol{x}_k^0 = \boldsymbol{\mu}_k \tag{46}$$
$$\boldsymbol{x}_i^0 = \mathbf{F}\boldsymbol{x}_{i-1}^0, i \in \{k+1 : k+N-1\}. \tag{47}$$

Subsequently, the measurement model of (3) is approximated using (5), where the linearization point at time i is $\boldsymbol{x}_i^0$. This well-known approximation is referred to as the *linearized Kalman filter* [14]; it was previously applied to a sensor scheduling problem in [25]. The controller which results from this approximation has a structure similar to the open-loop feedback controller (OLFC) [15]: at each stage, a plan for the next N time steps is generated, the first step of the plan executed, and then a new plan for the following N steps is generated, having relinearized after incorporating the newly received measurements.

A significant horizon length is required in order to provide an effective tradeoff between communication cost and inference quality, since many time steps are required for the long-term communication cost saved and information gained from a leader node change to outweigh the immediate communication cost incurred. While the linear Gaussian approximation eliminates the $O(N_p^N)$ factor in the growth of computational complexity with planning horizon length, the complexity is still exponential in both time and the number of sensors, growing as $O(N_s^N 2^{N_s N})$. The following two sections describe two tree pruning approximations we introduce to obtain a tractable implementation.

F. Greedy Sensor Subset Selection

To avoid the combinatorial complexity associated with optimization over subsets of sensors, we break each decision stage into a number of substages, indexed by i'. Conditioned on a choice of leader node for the current stage, the control choices at each substage are to select another (previously unselected) sensor or to terminate with the current set of selections (similar to the generalized stopping problem [15]). For the communication constrained formulation, the DP recursion becomes

$$\bar{J}_i(\mathbb{X}_i, l_{i-1}, \lambda) = \min_{l_i} \left\{ \lambda B_p C_{l_{i-1} l_i} + \bar{J}_{i,0}\left(\mathbb{X}_i, l_i, \{\emptyset\}, \lambda\right)\right\} \tag{48}$$

for $i \in \{k : k+N-1\}$, terminated by setting $\bar{J}_{k+N}(\mathbb{X}_{k+N}, l_{k+N-1}, \lambda) = -\lambda M$. The function $\bar{J}_{i,i'}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, \lambda)$, defined below, forms part of the same recursion, and represents the cost to go from substage i' of stage i to the end of the problem, whereas $\bar{J}_i(\mathbb{X}_i, l_{i-1}, \lambda)$ represents the cost to go from the beginning of stage i to the
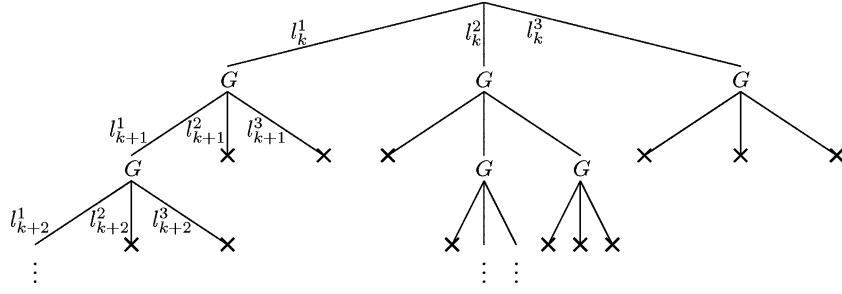
Fig. 2. Tree structure for n-scan pruning algorithm with $n = 1$. At each stage new leaves are generated extending each remaining sequence with using each new leader node. Subsequently, all but the best sequence ending with each leader node is discarded (marked with "$\times$"), and the remaining sequences are extended using greedy sensor subset selection (marked with "G").

end of the problem. $\mathcal{S}_{i,i'}$ is the set of sensors chosen in stage i prior to substage i'.

$$
\begin{aligned}
\bar{J}_{i,i'}&(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, \lambda) \\
&= \min \left\{ \underset{\mathbb{X}_{i+1}|\mathbb{X}_i, \mathcal{S}_{i,i'}}{\mathrm{E}} \bar{J}_{i+1}(\mathbb{X}_{i+1}, l_i, \lambda), \right. \\
&\qquad \underset{s_{i,i'}\in\mathcal{S}\setminus\mathcal{S}_{i,i'}}{\min} \left\{ \bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, s_{i,i'}, \lambda) \right. \\
&\qquad\qquad \left.\left. + \bar{J}_{i,i'+1}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}\cup s_{i,i'}, \lambda)\right\} \right\}
\end{aligned}
\tag{49}
$$

where the substage cost $\bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, s_{i,i'}, \lambda)$ is

$$
\bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, s_{i,i'}, \lambda) = \lambda B_m C_{l_i s_{i,i'}} - I\left(\boldsymbol{x}_i; \boldsymbol{z}_i^{s_{i,i'}} | \boldsymbol{z}_{0:i-1}, \boldsymbol{z}_i^{\mathcal{S}_{i,i'}}\right).
\tag{50}
$$

While this formulation is algebraically equivalent to the one described in Section III-A, it is in a form that is more suited to approximation. Namely, the substage optimization may be performed using a greedy method, in which, at each stage, if there is no sensor $s_{i,i'}$ for which the substage cost $\bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, s_{i,i'}, \lambda) \leq 0$ (i.e., for which the cost of transmitting the measurement is not outweighed by the expected information it will provide), then we progress to the next stage; otherwise the sensor $s_{i,i'}$ with the lowest substage cost is added. The fact that the constraint terms of the Lagrangian were distributed into the per-stage and per-substage cost allows the greedy approximation to be used in a way which trades off estimation quality and communication cost.

While worst-case complexity of this algorithm is $O(N_s^2)$, careful analysis of the sensor model can yield substantial practical reductions. One quite general simplification can be made: assuming that sensor measurements are independent conditioned on the state, one can show that, for the substage cost in (50)

$$
\bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, s, \lambda) \leq \bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i''}, s, \lambda) \quad \forall i' < i''.
\tag{51}
$$

This result exploits submodularity of mutual information, which is explored in detail in [26]. Using this result, if at any substage of stage i we find that the substage cost of adding a particular sensor is greater then zero, then that sensor will not be selected in any later substages of stage i; hence, it can be excluded from

consideration. In practice, this will limit the sensors requiring consideration to those in a small neighborhood around the current leader node and object, reducing computational complexity when dealing with large networks.

G. *n*-Scan Pruning

The algorithm described above is embedded within a slightly less coarse approximation for leader node selection, which incorporates costs over multiple time stages. This approximation operates similarly to the n-scan pruning algorithm, commonly used to control computational complexity in the multiple hypothesis tracker [27]. Setting $n = 1$, the algorithm is illustrated in Fig. 2. We commence by considering each possible choice of leader node[8] for the next time step and calculating the greedy sensor subset selection from Section III-F for each leader node choice. Then, for each leaf node, we consider the candidate leader nodes at the following time step. All sequences ending with the same candidate leader node are compared, the one with the lowest cost value is kept, and the other sequences are discarded. Thus, at each stage, we keep some approximation of the best control trajectory which ends with each sensor as leader node.

Using such an algorithm, the tree width is constrained to the number of sensors, and the overall worst case computational complexity is $O(NN_s^3)$ (in practice, at each stage we only consider candidate sensors in some neighborhood of the estimated object location, and the complexity will be substantially lower). This compares to the simulation-based evaluation of the full dynamic programming recursion which, as discussed in Section III-D, has a computation complexity of the order $O(N_s^N 2^{N_s N} N_p^N)$. The difference in complexity is striking: even for a problem with $N_s = 20$ sensors, a planning horizon of $N = 10$ and simulating $N_p = 50$ values of measurements at each stage, the complexity is reduced from 1.6×10^{90} to (at worst case) 8×10^5.

Because the communication cost structure is Markovian with respect to the leader node (i.e., the communication cost of a particular future control trajectory is unaffected by the control history given the current leader node), it is captured perfectly by this model. The information reward structure, which is not Markovian with respect to the leader node, is approximated using the greedy method.

---

[8]The set of candidate leader nodes would, in practice, be limited to sensors close to the object, similar to the sensor subset selection.

H. Sequential Subgradient Update

The previous two sections provide an efficient algorithm for generating a plan for the next N steps given a particular value of the dual variable λ. Substituting the resulting plan into (32) yields a subgradient that can be used to update the dual variables (under the linear Gaussian approximation, feedback policies correspond to open loop plans, hence the argument of the expectation of $\mathrm{E}[\sum_i G(\mathbb{X}_i, l_{i-1}, \mu_i(\mathbb{X}_i, l_{i-1})) - M]$ is deterministic). A full subgradient implementation would require evaluation for many different values of the dual variable at each time step, which is undesirable since each evaluation incurs a substantial computational cost. Since the planning is over many time steps, in practice the level of the constraint (i.e., the value of $\mathrm{E}[\sum_i G(\mathbb{X}_i, l_{i-1}, \mu_i(\mathbb{X}_i, l_{i-1})) - M]$) will vary little between time steps, hence the slow adaptation of the dual variable provided by a single subgradient step in each iteration may provide an adequate approximation.

In the experiments that follow, at each time step, we plan using a single value of the dual variable and then update it for the next time step utilizing either an additive update

$$
\lambda_{k+1} = \begin{cases} \min\{\lambda_k + \gamma^+, \lambda^{\max}\}, & \mathrm{E}\left[\sum_i G(\mathbb{X}_i, l_{i-1}, \right. \\ & \left. \mu_i(\mathbb{X}_i, l_{i-1}))\right] > M \\ \max\{\lambda_k - \gamma^-, \lambda^{\min}\}, & \mathrm{E}\left[\sum_i G(\mathbb{X}_i, l_{i-1}, \right. \\ & \left. \mu_i(\mathbb{X}_i, l_{i-1}))\right] \le M \end{cases}
$$

(52)

or a multiplicative update

$$
\lambda_{k+1} = \begin{cases} \min\{\lambda_k \beta^+, \lambda^{\max}\}, & \mathrm{E}\left[\sum_i G(\mathbb{X}_i, l_{i-1}, \right. \\ & \left. \mu_i(\mathbb{X}_i, l_{i-1}))\right] > M \\ \max\{\lambda_k / \beta^-, \lambda^{\min}\}, & \mathrm{E}\left[\sum_i G(\mathbb{X}_i, l_{i-1}, \right. \\ & \left. \mu_i(\mathbb{X}_i, l_{i-1}))\right] \le M \end{cases}
$$

(53)

where γ^+ and γ^- are the increment and decrement sizes, β^+ and β^- are the increment and decrement factors, and $\lambda^{\max}$ and $\lambda^{\min}$ are the maximum and minimum values of the dual variable. It is necessary to limit the values of the dual variable since the constrained problem may not be feasible. If the variable is not constrained, undesirable behavior can result, such as utilizing every sensor in a sensor network in order to meet an information constraint which cannot be met in any case, or because the dual variable in the communication constraint was adapted such that it became too low, effectively implying that communications are cost free.

I. Roll-Out

If the horizon length is set to be too small in the communications constrained formulation, then the resulting solution will be to hold the leader node fixed and take progressively fewer measurements. To prevent this degenerate behavior, we use a roll-out approach (a commonly used suboptimal control methodology), in which we add to the terminal cost in the DP recursion (36) the cost of transmitting the probabilistic model to the sensor with the smallest expected distance to the object at the final stage. Denoting by $\tilde{\mu}(\cdot) : \mathcal{P}(\mathcal{X}) \to \mathcal{S}$ the policy that selects as leader

node the sensor with the smallest expected distance to the object, the terminal cost is

$$
J_{k+N}(\mathbb{X}_{k+N}, l_{k+N-1}) = \lambda B_p C_{l_{k+N-1}\tilde{\mu}(\mathbb{X}_{k+N})}
$$

(54)

where the Lagrange multiplier λ is included only in the communication-constrained case. This effectively acts as the cost of the base policy in a roll-out [15]. The resulting algorithm constructs a plan which assumes that, at the final stage, the leader node will have to be transferred to the closest sensor; hence, there is no benefit in holding it at its existing location indefinitely. In the communication-constrained case, this modification will often make the problem infeasible for short planning horizons, but the limiting of the dual variables discussed in Section III-H can avoid anomalous behavior.

J. Surrogate Constraints

A form of information constraint that is often more desirable is one that captures the notion that it is acceptable for the uncertainty in object state to increase for short periods of time if informative measurements are likely to become available later. The minimum entropy constraint is such an example, as follows:

$$
\mathrm{E}\left\{\min_{i \in \{k:k+N-1\}} H(\boldsymbol{x}_i | \boldsymbol{z}_{0:i-1}) - H_{\max}\right\} \le 0.
$$

(55)

The constraint in (55) does not have an additive decomposition [cf. (40)], as required by the simplifications in Sections III-F and III-G. However, we can use the constraint in (40) to generate plans for a given value of the dual variable λ using the simplifications and then perform the dual variable update of Section III-H using the desired constraint (55). This simple approximation effectively uses the additive constraint in (40) as a surrogate for the desired constraint in (55), allowing us to use the computationally convenient method described previously with a more meaningful criterion.

IV. DECOUPLED LEADER NODE SELECTION

Most of the sensor management strategies proposed for object localization in existing literature seek to optimize the estimation performance of the system, incorporating communication cost indirectly, such as by limiting the maximum number of sensors utilized. These methods typically do not consider the leader node selection problem directly, although the communication cost consumed in implementing them will vary depending on the leader node since communications costs are dependent on the transmission distance. In order to compare the performance of the algorithm developed in Section III with these methods, we develop an approach which, conditioned on a particular sensor management strategy (that is insensitive to the choice of leader node), seeks to dynamically select the leader node to minimize the communications energy consumed due to activation, deactivation and querying of sensors by the leader node, and transmission of measurements from sensors to the leader node. This involves a tradeoff between two different forms of communication: the large, infrequent step increments produced when the probability distribution is transferred from sensor to sensor during leader node hand-off, and the small, frequent increments produced by activating, deactivating, and

querying sensors. The approach is fundamentally different from that in Section III as we are optimizing the leader node selection conditioned on a fixed sensor management strategy, rather than jointly optimizing sensor management and leader node selection.

A. Formulation

The objective that we seek to minimize is the expected communications cost over an N-step rolling horizon. We require the sensor management algorithm to provide predictions of the communications performed by each sensor at each time in the future. As in Section III, the problem corresponds to a DP in which the DP state at time k is the combination of the conditional pdf of object state $\mathbb{X}_k \triangleq p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1})$ and the previous leader node l_{k-1}. The control that we may choose is the leader node at each time $u_k = l_k \in \mathcal{S}$. Denoting the expected cost of communications expended by the sensor management algorithm (due to sensor activation and deactivation, querying, and transmission of measurements) at time k if the leader node is l_k as $g_c(\mathbb{X}_k, l_k)$, the DP for selecting the leader node at time k can be written as the following recursive equation:

$$J_i(\mathbb{X}_i, l_{i-1}) = \min_{l_i} \left\{ g_c(\mathbb{X}_i, l_i) + B_p C_{l_{i-1} l_i} \right.$$
$$\left. + \operatorname*{E}_{\mathbb{X}_{i+1}|\mathbb{X}_i, l_i} J_{i+1}(\mathbb{X}_{i+1}, l_i) \right\} \quad (56)$$

for $i \in \{k : k + N - 1\}$. In the same way as discussed in Section III-I, we set the terminal cost to the cost of transmitting the probabilistic model from the current leader node to the node with the smallest expected distance to the object $\tilde{\mu}(\mathbb{X}_{k+N})$:

$$J_{k+N}(\mathbb{X}_{k+N}, l_{k+N-1}) = B_p C_{l_{k+N-1} \tilde{\mu}(\mathbb{X}_{k+N})}. \quad (57)$$

In Section V, we apply this method using a single look-ahead step $(N = 1)$ with a greedy sensor management strategy selecting, first the most informative measurement and then, second, the two most informative measurements.

V. SIMULATION RESULTS

The model presented in Section II-A was simulated for 100 Monte Carlo trials using 20 sensors positioned randomly according to a uniform distribution inside a 100×100-unit region; each trial used a different sensor layout and object trajectory. The initial position of the object is in one corner of the region, and the initial velocity is two units per second in each dimension, moving into the region. The simulation ends when the object leaves the 100×100 region or after 200 time steps, which ever occurs sooner (the average length is around 180 steps). The sample time was $T = 0.25$ s, diffusion strength was $q = 10^{-2}$, and the measurement model parameters were $a = 2000$, $b = 100$, and $R = 1$. The communication costs were $B_p = 64$ and $B_m = 1$, so that the cost of transmitting the probabilistic model is $64\times$ the cost of transmitting a measurement. For the communication-constrained problem, a multiplicative update was used for the subgradient method, with $\beta^+ = \beta^- = 1.2$, $\lambda^{\min} = 10^{-5}$, $\lambda^{\max} = 5 \times 10^{-3}$, and $C_{\max} = 10N$, where N is the planning horizon length. For
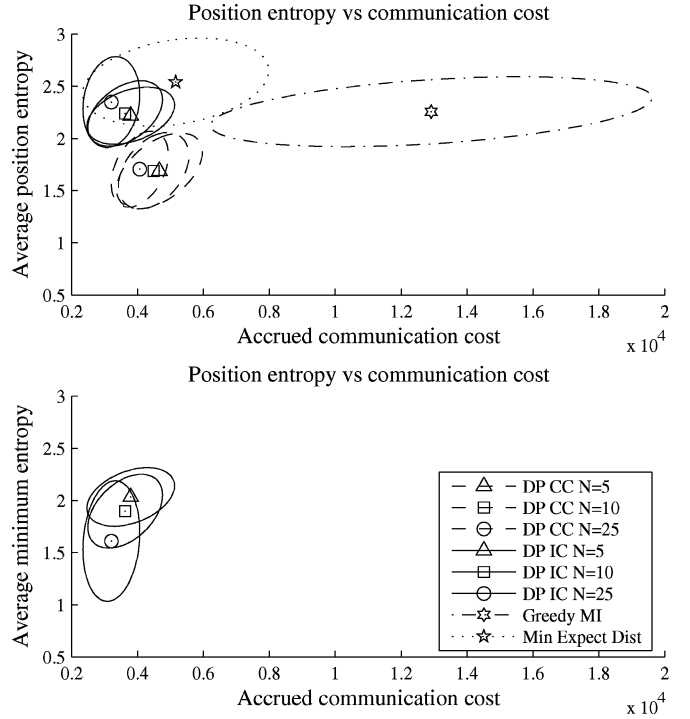

Fig. 3. Position entropy and communication cost for dynamic programming method with communication constraint (DP CC) and information constraint (DP IC) with different planning horizon lengths (N), compared to the methods selecting as leader node and activating the sensor with the largest mutual information (greedy MI), and the sensor with the smallest expected square distance to the object (min expect dist). Ellipse centers show the mean in each axis over 100 Monte Carlo runs; ellipses illustrate covariance, providing an indication of the variability across simulations. Upper figure compares average position entropy to communication cost, while lower figure compares average of the minimum entropy over blocks of the same length as the planning horizon (i.e., the quantity to which the constraint is applied) to communication cost.

the information-constrained problem, an additive update was used for the subgradient method, with $\gamma^+ = 50$, $\gamma^- = 250$, $\lambda^{\min} = 10^{-8}$, $\lambda^{\max} = 500$, and $H_{\max} = 2$ (these parameters were determined experimentally).

The simulation results are summarized in Fig. 3. The top diagram demonstrates that the communication-constrained formulation provides a way of controlling sensor selection and leader node which reduces the communication cost and improves estimation performance substantially over the myopic single-sensor methods, which, at each time, activate and select as leader node the sensor with the measurement producing the largest expected reduction in entropy. The information-constrained formulation allows for an additional saving in communication cost while meeting an estimation criterion wherever possible.

The top diagram in Fig. 3 also illustrates the improvement which results from utilizing a longer planning horizon. The constraint level in the communication-constrained case is 10 cost units per time step; since the average simulation length is 180 steps, the average communication cost if the constraint were always met with equality would be 1800. However, because this cost tends to occur in bursts (due to the irregular hand-off of leader node from sensor to sensor as the object moves), the practical behavior of the system is to reduce the dual variable when there is no hand-off in the planning horizon (allowing more
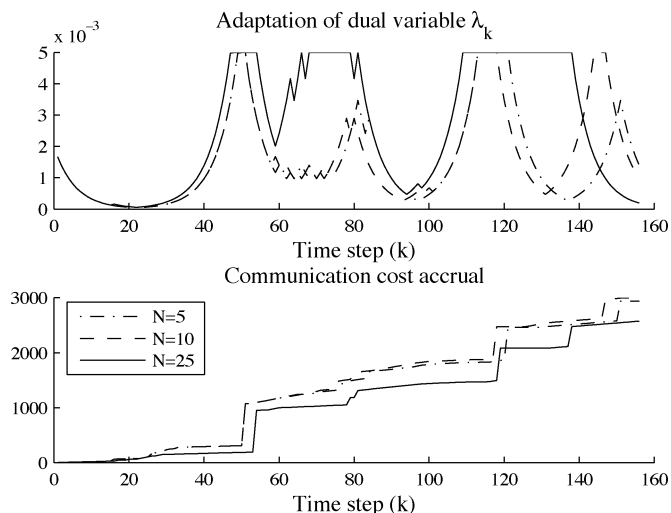
Fig. 4. Adaptation of communication constraint dual variable λ_k for different horizon lengths for a single Monte Carlo run, and corresponding cumulative communication costs.
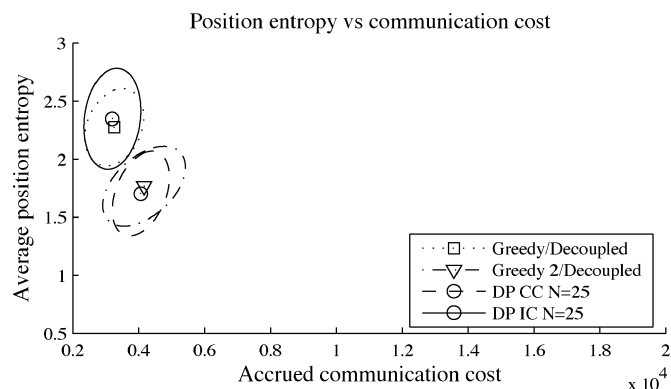

Fig. 5. Position entropy and communication cost for dynamic programming method with communication constraint (DP CC) and information constraint (DP IC), compared to the method which dynamically selects the leader node to minimize the expected communication cost consumed in implementing a fixed sensor management scheme. The fixed sensor management scheme activates the sensor ("greedy") or two sensors ("greedy 2") with the measurement or measurements producing the largest expected reduction in entropy. Ellipse centers show the mean in each axis over 100 Monte Carlo runs; ellipses illustrate covariance, providing an indication of the variability across simulations.

sensor measurements to be utilized) and increase it when there is a hand-off in the planning horizon (to come closer to meeting the constraint). A longer planning horizon reduces this undesirable behavior by anticipating upcoming leader node hand-off events earlier and tempering spending of communication resources sooner. This is demonstrated in Fig. 4, which shows the adaptation of the dual variable for a single Monte Carlo run.

In the information-constrained case, increasing the planning horizon relaxes the constraint, since it requires the *minimum* entropy within the planning horizon to be less than a given value. Accordingly, using a longer planning horizon, the average minimum entropy is reduced, and additional communication energy is saved. The lower diagram in Fig. 3 shows the average minimum entropy in blocks of the same length as the planning horizon, demonstrating that the information constraint is met more often with a longer planning horizon (as well as resulting in a larger communication saving).

Fig. 5 compares the adaptive Lagrangian relaxation method discussed in Section III with the decoupled scheme discussed in Section IV, which adaptively selects the leader node to minimize the expected communication cost expended in implementing the decision of the fixed sensor management method. The fixed sensor management scheme activates the sensor or two sensors with the measurement or measurements producing the largest expected reduction in entropy. The results demonstrate that for this case the decoupled method using a single sensor at each time step results in similar estimation performance and communication cost to the Lagrangian relaxation method using an information constraint with the given level. Similarly, the decoupled method using two sensors at each time step results in similar estimation performance and communication cost to the Lagrangian relaxation method using a communication constraint with the given level. The additional flexibility of the Lagrangian relaxation method allows one to select the constraint level to achieve various points on the estimation performance/communication cost tradeoff, rather than

being restricted to particular points corresponding to different numbers of sensors.

VI. CONCLUSION AND FUTURE WORK

This paper has demonstrated how an adaptive Lagrangian relaxation can be utilized for sensor management in an energy-constrained sensor network. The introduction of secondary objectives as constraints provides a natural methodology to address the tradeoff between estimation performance and communication cost.

The planning algorithm may be applied alongside a wide range of estimation methods, ranging from the Kalman filter to the particle filter. The algorithm is also applicable to a wide range of sensor models. The linearized Gaussian approximation in Section III-E results in a structure identical to the OLFC. The remainder of our algorithm (removing the linearized Gaussian approximation) may be applied to find an efficient approximation of the OLFC as long as an efficient estimate of the reward function (mutual information in our case) is available.

The simulation results in Section V demonstrate that approximations based on dynamic programming are able to provide similar estimation performance (as measured by entropy), for a fraction of the communication cost in comparison to simple heuristics which consider estimation performance alone and utilize a single sensor. The discussion in Section III-F provides a guide for efficient implementation strategies that can enable implementation on the latest generation wireless sensor networks. Future work includes incorporation of the impact on planning caused by the interaction between objects when multiple objects are observed by a single sensor, and developing approximations that are less coarse than the linearized Gaussian model.

REFERENCES

[1] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
[2] K. Pister, "Smart dust (keynote address)," presented at the 2nd Int. Workshop Information Processing in Sensor Networks (IPSN) '03, Palo Alto, CA, Apr. 22–23, 2003.

[3] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.

[4] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP J. Appl. Signal Process.*, no. 4, pp. 378–391, 2003.

[5] M. Jones, S. Mehrotra, and J. H. Park, "Tasking distributed sensor networks," *Int. J. High Perform. Comput. Appl.*, vol. 16, no. 3, pp. 243–257, 2002.

[6] D. A. Castañón, "Approximate dynamic programming for sensor management," in *Proc 36th Conf. Decision Control*, Dec. 1997, pp. 1202–1207.

[7] Y. He and E. K. P. Chong, "Sensor scheduling for target tracking: A Monte Carlo sampling approach," *Digit. Signal Process.*, vol. 16, no. 5, pp. 533–545, Sep. 2006.

[8] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Energy efficient target tracking in a sensor network using non-myopic sensor scheduling," presented at the 8th Int. Conf. Information Fusion, Philadelphia, PA, Jul. 25–29, 2005.

[9] J. L. Williams, J. W. Fisher, III, and A. S. Willsky, "An approximate dynamic programming approach for communication constrained inference," in *Proc. IEEE Workshop Statistical Signal Processing*, Jul. 2005, pp. 1201–1207.

[10] J. L. Williams, J. W. Fisher, III, and A. S. Willsky, "An approximate dynamic programming approach to a communication constrained sensor management problem," presented at the 8th Int. Conf. Information Fusion, Philadelphia, PA, Jul. 25–29, 2005.

[11] J. L. Williams, J. W. Fisher, III, and A. S. Willsky, "Optimization approaches to dynamic routing of measurements and models in a sensor network object tracking problem," in *Proc. Int. Conf. Acoustics, Speech Signal Processing*, Mar. 2005, pp. 1061–1065, IEEE.

[12] A. T. Ihler, J. W. Fisher, III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[13] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[14] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Arlington, VA: Navtech, 1994, vol. 2.

[15] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2000.

[16] A. Ihler, J. Fisher, III, and A. Willsky, "Communications-constrained inference," Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Tech. Rep. 2601, 2004.

[17] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 61–72, Mar. 2002.

[18] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *Int. J. High Perform. Comput. Appl.*, vol. 16, no. 3, pp. 293–313, 2002.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[20] E. Ertin, J. W. Fisher, and L. C. Potter, "Maximum mutual information principle for dynamic sensor query problems," in *Proc 2nd Int. Workshop Information Processing in Sensor Networks (IPSN) 2003*, Palo Alto, CA, Apr. 22–23, 2003, pp. 405–416, Springer-Verlag.

[21] C. M. Kreucher, A. O. Hero, III, K. Kastella, and D. Chang, "Efficient methods of non-myopic sensor management for multitarget tracking," in *Proc. 43rd IEEE Conf. Decision Control*, Dec. 2004, vol. 1, pp. 722–727.

[22] E. Altman, *Constrained Markov Decision Processes*. London, U.K.: Chapman & Hall, 1999.

[23] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[24] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Arlington, VA: Navtech, 1994, vol. 1.

[25] A. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Scheduling multiple sensors using particle filters in target tracking," in *Proc. IEEE Workshop Statistical Signal Processing*, Sep./Oct. 2003, pp. 549–552.

[26] J. L. Williams, "Information theoretic sensor management," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, Feb. 2007.

[27] T. Kurien, "Issues in the design of practical multitarget tracking algorithms," in *Multitarget-Multisensor Tracking: Advanced Applications*. Norwood, MA: Artech-House, 1990, pp. 43–83.

Jason L. Williams (S'01) received the B.Eng. degree in electronics and the B.Inf.Tech. degree from Queensland University of Technology, Brisbane, Australia, in 1999 and the M.S.E.E. degree from the Air Force Institute of Technology, Fairborn, OH, in 2003. He is currently working towards the Ph.D. degree at the Massachusetts Institute of Technology (MIT), Cambridge.

He is also a Research Assistant at the Stochastic Systems Group at MIT. His research interests include stochastic estimation and control theory, sensor resource management, sensor networks, and object tracking.

Mr. Williams received the Mervin E. Gross Award for exceptional academic achievement and the Commandant's Award for his Master's thesis research, both from the Air Force Institute of Technology.

John W. Fisher, III (M'90) received the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 1997.

He was previously affiliated with the University of Florida as both a faculty member and graduate student since 1987, during which time he conducted research in the areas of ultrawideband radar for ground penetration and foliage penetration applications, radar signal processing, and automatic target recognition algorithms. He is currently a Principal Research Scientist in the Computer Science and Artificial Intelligence Laboratory and is affiliated with the Laboratory for Information and Decision Systems, both at the Massachusetts Institute of Technology, Cambridge. His current area of research focus includes information theoretic approaches to signal processing, multimodal data fusion, machine learning, and computer vision.

Alan S. Willsky (S'70–M'73–SM'82–F'86) joined the Massachusetts Institute of Technology, Cambridge, in 1973, where he is currently the Edwin Sibley Webster Professor of Electrical Engineering. He was a founder of Alphatech, Inc., and Chief Scientific Consultant, a role he continues at BAE Systems Advanced Information Technologies. He is coauthor of the text *Signals and Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1996, 2nd ed.). His research interests are in the development and application of advanced methods of estimation and statistical signal and image processing.

Dr. Willsky served on the U.S. Air Force Scientific Advisory Board from 1998 to 2002. He has received several awards, including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, the 1980 IEEE Browder J. Thompson Memorial Award, the IEEE Control Systems Society Distinguished Member Award in 1988, the 2004 IEEE Donald G. Fink Prize Paper Award, and Doctorat Honoris Causa from Universitá de Rennes in 2005. He has delivered numerous keynote addresses.