An Approximate Dynamic Programming Approach to a Communication Constrained Sensor Management Problem

Jason L. Williams,* John W. Fisher III† and Alan S. Willsky*
*Laboratory for Information and Decision Systems and †Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
Email: jlwil@mit.edu, fisher@csail.mit.edu, willsky@mit.edu

***Abstract*— Resource management in distributed sensor networks is a challenging problem. This can be attributed to the fundamental trade-off between the value of information contained in a distributed set of measurements versus the energy costs of acquiring measurements, fusing them into a model of uncertainty, and transmitting the resulting model. Communications is commonly the highest contributor among these costs, typically by orders of magnitude. Failure to consider this trade-off 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 which integrates the value of information and the cost of transmitting data over a rolling time horizon. We formulate this trade-off 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.**

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 inter-sensor communication. In many sensor networks energy is a dear resource to be conserved so as to prolong the network's operational lifetime. Additionally, 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].

Tracking moving objects is a common application in which the quantities of interest (*i.e.*, kinematic state) are inferred largely from sensor measurements which 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 posterior distribution. This property, combined with the need to conserve energy, has led to a variety of approaches [4], [5] which effectively designate the responsibility of computing the model 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 trade-off 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 trade-off 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 to 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 purposes 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 was 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 *a priori* model and tasks sensors at the next time step. The questions which 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 controller developed in Section III extends [6] by utilizing multiple sensors at each time step rather than only activating the leader node, 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 formulation is closely related to [7], and provides an approximation which extends the Lagrangian relaxation approach to problems involving sequential replanning.

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 probability distribution 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 $\boldsymbol{x}_k$ as the state of the object (or 'object state') at time k, we assume that object dynamics evolve according to a linear Gaussian model:

$$\boldsymbol{x}_{k+1} \quad = \quad \mathbf{F}\boldsymbol{x}_k + \boldsymbol{w}_k \tag{1}$$

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 modelled 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:

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}; \; \mathbf{Q} = q \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} \tag{2}$$

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:

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

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:

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

where $\mathbf{L}$ is the matrix which extracts the position of the object from the object state (such that $\mathbf{L}\boldsymbol{x}_k$ is the location of the object), and $\boldsymbol{l}^s$ is the location of the s-th sensor. The constants a and b can be tuned to model the signal-to-noise ratio of the sensor, and the fall-off in the region close to the sensor (allowing a saturation effect to be approximated). The measurement has additive Gaussian noise as per Eq. (3), with variance R. The information provided by the measurement reduces as the range increases due to the nonlinearity.

[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 $\mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}, \mathbf{P}\} = |2\pi\mathbf{P}|^{-\frac{1}{2}} \exp\{-0.5(\boldsymbol{x} - \boldsymbol{\mu})^T\mathbf{P}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\}$.

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 through a first-order Taylor series truncation as:

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

$$\mathbf{H}^s(\boldsymbol{x}^0) \quad = \quad \boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{h}(\boldsymbol{x}, s)|_{\boldsymbol{x}=\boldsymbol{x}^0} \tag{6}$$

The linearization of the model in Eq. (4) is:

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

This specific model, which will be utilized in the simulations in Section IV, has been specified for concreteness; we reiterate that the approach described has general applicability.

B. Estimation

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 Eq. (4) is significant, and substantial multimodality can result. Accordingly, we utilize a particle filter approximation for the dynamic state estimation, whereby the Probability Density Function (PDF) of object state $\boldsymbol{x}_k$ conditioned on measurements received up to and including time k, $\boldsymbol{z}_{0:k}$, is approximated through a set of N_p weighted samples:

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

To calculate the same distribution at the next time step, $p(\boldsymbol{x}_{k+1}|\boldsymbol{z}_{0:k+1})$, we utilize an approximate Sequential Importance Sampling (SIS) algorithm [8] 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 $\boldsymbol{z}_{k+1}$ (Eq. (3)) about the point $\mathbf{F}\boldsymbol{x}_k^i$, as described in Eq. (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:

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

at time $(k+1)$. This distribution can be updated using the extended Kalman filter update equation [9] to obtain:

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

where

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

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

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

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

$$w_{k+1}^i \propto c w_k^i \frac{p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}^i)p(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i)}{q(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i, \boldsymbol{z}_{k+1})} \quad (14)$$

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:

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

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

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

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 of direct communication between two nodes is modelled as being proportional to the square distance between the two sensors:

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

Communications between distant nodes can be performed more efficiently using a multi-hop 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 Eq. (17) as arc lengths:

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

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 [10]. We assume that the complexity of the probabilistic model (*i.e.*, the number of bits required for transmission) is fixed, such that the energy required to communicate the model from node i to node j is directly proportional to C_{ij}. The ratio of the number of bits in a measurement to the number of bits in the probabilistic model is denoted as r, such that the energy required to transmit a measurement from node i to node j is rC_{ij}. These costs may be amended to incorporate the cost of activating the sensor, taking the measurement, *etc*, without changing the structure of the solution.

D. Sensor management

As discussed in Section I, the role of the sensor manager in a sensor network tracking problem is to trade off estimation performance against 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 [11], [12] has demonstrated the effectiveness of conditional entropy as an objective function. A common sensor management algorithm (*e.g.* [4]) is to select as the new leader node l_k the sensor whose measurement minimizes the expected posterior entropy, and activate only that sensor:

$$l_k = \arg\min_l H(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^l) \quad (19)$$

where the conditional entropy is defined as: [13]

$$H(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^l) = -\int p(\boldsymbol{z}_k^l|\boldsymbol{z}_{0:k-1}) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^l) \log p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^l) \mathrm{d}\boldsymbol{x}_k \mathrm{d}\boldsymbol{z}_k^l \quad (20)$$

The conditioning in Eq. (19) is on the *value* of the previous measurements $\boldsymbol{z}_{0:k-1}$, and on the *random variable* corresponding to the new measurement $\boldsymbol{z}_k^l$, implying marginalization over all possible values of the measurement. The mutual information between the object state $\boldsymbol{x}_k$ and measurement $\boldsymbol{z}_k^l$ conditioned on the previous measurement history is defined as [13]:

$$\begin{aligned} I(\boldsymbol{x}_k; \boldsymbol{z}_k^l|\boldsymbol{z}_{0:k-1}) &= H(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}) - H(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^l) \\ &= H(\boldsymbol{z}_k^l|\boldsymbol{z}_{0:k-1}) - H(\boldsymbol{z}_k^l|\boldsymbol{x}_k) \end{aligned} \quad (21)$$

Since the first term of the first form in Eq. (21) is independent of the control decision (l), it is clear that minimization of expected conditional entropy is equivalent to maximization of mutual information [14]. The latter form, which is equivalent, sometimes leads to a more efficient implementation.

Sensor management strategies which 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.*, [15]) 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.

Another intuitive heuristic approach is to select as the new leader node the sensor whose expected distance (according to some norm) to the object is minimized:

$$l_k = \arg\min_l \mathrm{E}_{\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}} \{||\mathbf{L}\boldsymbol{x}_k - \boldsymbol{l}^l||_2^2\} \quad (22)$$

One would expect this scheme to result in a reasonably small communication cost, as the active leader node will roughly follow the object trajectory, although this may come at the cost of poorer tracking performance, since the geometry of sensor observations is not considered.

III. Constrained Dynamic Programming Formulation

As discussed in Section II-D, the sensor network object tracking problem involves an inherent trade-off 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 measurements chosen 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) [7], [16]. Similarly to imperfect state information problems [10], 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 $\mathbb{X}_k \triangleq p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1})$;[2] 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}. 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 (this departs from [6], which activates only the leader node at each time step).

A. Constrained communication formulation

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

$$g(\mathbb{X}_k, l_{k-1}, u_k) = -I(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k}|\boldsymbol{z}_{0:k-1}) \quad (23)$$

$$= -\sum_{j=1}^{|\mathcal{S}_k|} I(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k^j}|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^{\mathcal{S}_k^{1:j-1}}) \quad (24)$$

where $\mathcal{S}_k^j$ is the j-th element of $\mathcal{S}_k$ and $\mathcal{S}_k^{1:j-1}$ is the set containing the first $(j-1)$ elements of $\mathcal{S}_k$. The dynamic program for minimizing Eq. (23) over the N-step rolling horizon $\{k : k+N-1\}$ subject to a communication constraint can be described by the following recursive cost-to-go function:

$$J_i(\mathbb{X}_i, l_{i-1}) = \min_{u_i} \left\{ g(\mathbb{X}_i, l_{i-1}, u_i) + \mathop{\mathrm{E}}_{\mathbb{X}_{i+1}|\mathbb{X}_i, u_i} J_{i+1}(\mathbb{X}_{i+1}, l_i) \right\} \quad (25)$$

for $i \in \{k : k+N-1\}$, terminated by $J_{k+N}(\mathbb{X}_{k+N}, l_{k+N-1}) = 0$, subject to the communication constraint:

$$G(\mathbb{X}_k, l_{k-1}) = \mathrm{E}\left\{ \sum_{i=k}^{k+N-1} \left[C_{l_{i-1}l_i} + \sum_{j\in\mathcal{S}_i} rC_{l_ij} \right] - C_{\max} \right\} \leq 0 \quad (26)$$

The belief state at the next time $\mathbb{X}_{i+1}$ is calculated using the recursive Bayes update described in Section II-B. The expectation in the communication constraint of Eq. (26) is over the values of future measurements, noting that the future control decisions u_l depend on the values of the measurements received in the interim. To address the inequality constraint, we introduce a Lagrange multiplier λ and solve the dual problem $\bar{J}_k(\mathbb{X}_k, l_{k-1}) = \max_{\lambda\geq 0} \bar{J}_k(\mathbb{X}_k, l_{k-1}, \lambda)$, where $\bar{J}_k(\mathbb{X}_k, l_{k-1}, \lambda)$ is the Lagrangian, which incorporates the original dynamic program, plus the constraint term $\lambda G(\mathbb{X}_k, l_{k-1})$. The dynamic programming cost recursion of Eq. (25) consists of a sequence of nested expectations and minimizations; when we integrate the constraint into the recursion, each term of the summation must fall inside the corresponding nesting. A form which achieves this, and leads to an efficient approximation, is to integrate the cost elements into the per-stage cost:

$$\bar{g}(\mathbb{X}_k, l_{k-1}, u_k, \lambda) = g(\mathbb{X}_k, l_{k-1}, u_k) + \lambda \left[C_{l_{k-1}l_k} + \sum_{j\in\mathcal{S}_k} rC_{l_kj} \right] \quad (27)$$

The recursion then follows a form identical to Eq. (25) using the modified stage cost of Eq. (27), terminated by $\bar{J}_{k+N}(\mathbb{X}_{k+N}, l_{k+N-1}, \lambda) = -\lambda C_{\max}$.

B. 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 which optimizes the communication energy subject to a constraint on the object entropy. We commence by formulating a constraint on the joint entropy of the state of the object at each time in the planning horizon:

$$G(\mathbb{X}_k, l_{k-1}) = \mathrm{E}\left\{ H(\boldsymbol{x}_{k:k+N-1}|\boldsymbol{z}_{0:k+N-1}) - H_{\max} \right\} \leq 0 \quad (28)$$

Manipulating this expression using Eq. (21), we obtain

$$G(\mathbb{X}_k, l_{k-1}) = -\mathrm{E}\left\{ \sum_{i=k}^{k+N-1} \sum_{j=1}^{|\mathcal{S}_i|} I(\boldsymbol{x}_i; \boldsymbol{z}_i^{\mathcal{S}_i^j}|\boldsymbol{z}_{0:i-1}, \boldsymbol{z}_i^{\mathcal{S}_i^{1:j-1}}) - I_{\min} \right\} \quad (29)$$

where $I_{\min} = H(\boldsymbol{x}_{k:k+N-1}|\boldsymbol{z}_{0:k-1}) - H_{\max}$. Following the same procedure as described previously, the elements of the information constraint in Eq. (29) can be integrated into the per-stage cost (*cf* Eq. (27)), resulting in a formulation which is identical to the previous one, except that the Lagrange multiplier is on the mutual information terms, rather than the communication cost terms.

C. Subgradient optimization

Conceptually, the dual problems in Section III-A and Section III-B can be solved using a subgradient method [17]. The following expression can be seen to be a subgradient of the dual objective:

$$\partial\bar{J}_k(\mathbb{X}_k, l_{k-1}, \lambda) \ni \begin{cases} 0, & \lambda = 0,\ G(\mathbb{X}_k, l_{k-1}) \leq 0 \\ G(\mathbb{X}_k, l_{k-1}), & \text{otherwise} \end{cases} \quad (30)$$

[2] Conditioning on previous control decisions is assumed throughout. In contrast to the convention of [10], 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.

where $\partial \bar{J}_k(\mathbb{X}_k, l_{k-1}, \lambda)$ is the subdifferential with respect to λ (*i.e.*, the set of subgradients). 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 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 dual problem is the Lagrangian relaxation of the original constrained optimization, which is a common approximation method for discrete optimization problems.

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 which are utilized to obtain a practical implementation.

D. Evaluation through Monte Carlo simulation

Conceptually, the dynamic program of Eq. (25) 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 $\bar{J}_k(\mathbb{X}_k, l_{k-1}, \lambda)$ 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 $\bar{J}_{k+1}(\mathbb{X}_{k+1}, l_k, \lambda)$ 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 lookahead length) increases. Such an approach quickly becomes intractable even for a small number of possible controls (N_s) and simulated measurement samples (N_p), hence we seek to exploit additional structure in the problem to find a computable approximate solution.

E. Linearized Gaussian approximation

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

$$\boldsymbol{z}_k^{\mathcal{S}_k} = \mathbf{H}_k^{\mathcal{S}_k} \boldsymbol{x}_k + \boldsymbol{v}_k^{\mathcal{S}_k} \tag{31}$$

and the *a priori* distribution of $\boldsymbol{x}_k$ is $\mathcal{N}\{\boldsymbol{x}_k; \boldsymbol{\mu}_k, \mathbf{P}_k\}$, then from Eq. (21):

$$I(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k}|\boldsymbol{z}_{0:k-1}) = H(\boldsymbol{z}_k^{\mathcal{S}_k}|\boldsymbol{z}_{0:k-1}) - H(\boldsymbol{z}_k^{\mathcal{S}_k}|\boldsymbol{x}_k)$$

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

$$H(\boldsymbol{z}_k^{\mathcal{S}_k}|\boldsymbol{x}_k) = \tfrac{1}{2}\log|2\pi e \mathbf{R}^{\mathcal{S}_k}| \tag{32}$$

Similarly, with the linear measurement model, $\boldsymbol{z}_k^{\mathcal{S}_k}|\boldsymbol{z}_{0:k-1} \sim \mathcal{N}(\boldsymbol{z}_k^{\mathcal{S}_k}; \mathbf{H}_k^{\mathcal{S}_k}\boldsymbol{\mu}_k, \mathbf{H}_k^{\mathcal{S}_k}\mathbf{P}_k{\mathbf{H}_k^{\mathcal{S}_k}}^T + \mathbf{R}^{\mathcal{S}_k})$ [18], thus

$$H(\boldsymbol{z}_k^{\mathcal{S}_k}|\boldsymbol{z}_{0:k-1}) = \tfrac{1}{2}\log|2\pi e(\mathbf{H}_k^{\mathcal{S}_k}\mathbf{P}_k{\mathbf{H}_k^{\mathcal{S}_k}}^T + \mathbf{R}^{\mathcal{S}_k})| \tag{33}$$

Collecting results, we obtain

$$I(\boldsymbol{x}_k; \boldsymbol{z}_k^{\mathcal{S}_k}|\boldsymbol{z}_{0:k-1}) = \tfrac{1}{2}\log\left[|\mathbf{H}_k^{\mathcal{S}_k}\mathbf{P}_k{\mathbf{H}_k^{\mathcal{S}_k}}^T + \mathbf{R}^{\mathcal{S}_k}|/|\mathbf{R}^{\mathcal{S}_k}|\right] \tag{34}$$

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 non-linear, 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 Eq. (16). We then calculate the nominal trajectory as the mean at each of the following N steps:

$$\boldsymbol{x}_k^0 = \boldsymbol{\mu}_k \tag{35}$$

$$\boldsymbol{x}_i^0 = \mathbf{F}\boldsymbol{x}_{i-1}^0,\ i \in \{k+1 : k+N-1\} \tag{36}$$

Subsequently, the measurement model of Eq. (3) is approximated using Eq. (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 [9]. The controller which results has a structure similar to the Open Loop Feedback Controller (OLFC) [10]: 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 trade-off 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'. 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 [10]). The DP recursion becomes:

$$\bar{J}_i(\mathbb{X}_i, l_{i-1}, \lambda) = \min_{l_i}\{\lambda C_{l_{i-1}l_i} + \bar{J}_{i,0}(\mathbb{X}_i, l_i, \{\emptyset\}, \lambda)\} \quad (37)$$

where

$$\bar{J}_{i,i'}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, \lambda) = \min\Bigg\{ \mathop{\mathrm{E}}_{\mathbb{X}_{i+1}|\mathbb{X}_i, \mathcal{S}_{i,i'}} \bar{J}_{i+1}(\mathbb{X}_{i+1}, l_i, \lambda), \min_{s_{i,i'} \in \mathcal{S}\setminus\mathcal{S}_{i,i'}} \{\bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, s_{i,i'}, \lambda) + \bar{J}_{i,i'+1}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'} \cup s_{i,i'}, \lambda)\}\Bigg\} \quad (38)$$

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

$$\bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, s_{i,i'}, \lambda) = \lambda r C_{l_i s_{i,i'}} - I(\boldsymbol{x}_i; \boldsymbol{z}_i^{s_{i,i'}} | \boldsymbol{z}_{0:i-1}, \boldsymbol{z}_i^{\mathcal{S}_{i,i'}}) \quad (39)$$

While this formulation is algebraically equivalent to the one described previously, it is in a form which 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 $\bar{g}(\mathbb{X}_i, l_i, \mathcal{S}_{i,i'}, s_{i,i'}, \lambda) \leq 0$, then we progress to the next stage; otherwise the sensor $s_{i,i'}$ with the lowest substage cost is added. The dissemination of the constraint terms of the Lagrangian 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.

The worst-case complexity of this algorithm is $O(N_s^2)$. In practice, the set of sensors from which the subset selection is performed would be limited to sensors close to the 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 [19]. Setting $n = 1$, the algorithm commences by calculating the above greedy sensor subset selection for each candidate leader node (the set of candidate leader nodes would, in practice, be limited to sensors close to the object, similar to the sensor subset selection). The sensor selections for each leader node are then extended with each candidate leader node at the following time step. All sequences ending with each new 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 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 computational complexity is $O(NN_s{}^3)$. The resulting tree structure is illustrated in Fig. 1.

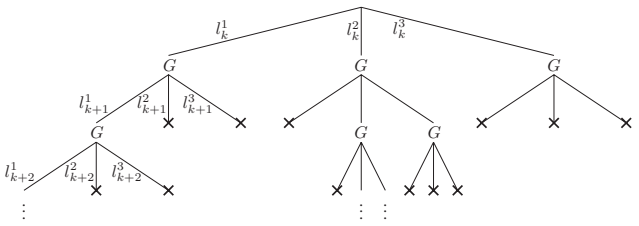

Fig. 1. Tree structure for n-scan pruning algorithm with $n = 1$. At each stage new leaves are generated extending each remaining sequence with using each new leader node. Subsequently, all but the best sequence ending with each leader node is discarded (marked with '$\times$'), and the remaining sequences are extended using greedy sensor subset selection (marked with 'G').

Because the communication cost structure is Markovian (*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, is approximated using the greedy method.

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 Eq. (30) yields a subgradient which 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 in $G(\mathbb{X}_k, l_{k-1})$ 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.[3] Since the planning is over many time steps, in practice the level of the constraint (*i.e.*, the value of $G(\mathbb{X}_k, l_{k-1})$) 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 which 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:

$$\lambda_{k+1} = \begin{cases} \min\{\lambda_k + \gamma^+, \lambda^{\max}\}, & G(\mathbb{X}_k, l_{k-1}) > 0 \\ \max\{\lambda_k - \gamma^-, \lambda^{\min}\}, & G(\mathbb{X}_k, l_{k-1}) \leq 0 \end{cases} \quad (40)$$

or a multiplicative update:

$$\lambda_{k+1} = \begin{cases} \min\{\lambda_k \beta^+, \lambda^{\max}\}, & G(\mathbb{X}_k, l_{k-1}) > 0 \\ \max\{\lambda_k / \beta^-, \lambda^{\min}\}, & G(\mathbb{X}_k, l_{k-1}) \leq 0 \end{cases} \quad (41)$$

[3]The rolling horizon formulation necessitates reoptimization of the dual variable at every time step, as opposed to [7].

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.

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 add to the terminal cost in the DP recursion (Eq. (25)) the cost of transmitting the probabilistic model to the sensor with the smallest expected distance to the object at the final stage (this is effectively the base policy in a roll-out [10]). This 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. This modification will often make the problem infeasible for short planning horizons, but the limiting of the dual variables discussed in Section III-H should avoid anomalous behavior.

J. Surrogate constraints

A form of information constraint which is often more desirable is one which 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, such as the minimum entropy constraint:

$$\tilde{G}(\mathbb{X}_k, l_{k-1}) = \mathrm{E}\left\{ \min_{i \in \{k:k+N-1\}} H(\boldsymbol{x}_i | \boldsymbol{z}_{0:i-1}) - H_{\max} \right\} \leq 0 \tag{42}$$

While constraint in Eq. (42) does not lead to a form which can be incorporated into the per-stage and per-substage cost (*cf* Eq. (29)), we can use the constraint in Eq. (29) to generate plans for a given value of the dual variable λ, and perform the dual variable update using the desired constraint, Eq. (42). This effectively uses the constraint $G(\mathbb{X}_k, l_{k-1})$ in Eq. (29) as a surrogate for the desired constraint $\tilde{G}(\mathbb{X}_k, l_{k-1})$ in Eq. (42), allowing us to use the computationally convenient method described above with a more meaningful criterion.

IV. 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 region; each trial used a different sensor layout. The initial position of the object is in one corner of the region, and the velocity is 2 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 $T = 0.25$ sec, diffusion strength was $q = 10^{-2}$, and the measurement model parameters were $a = 2000$, $b = 100$ and $R = 1$. 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 $C_{\max} = 10N$. For 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).

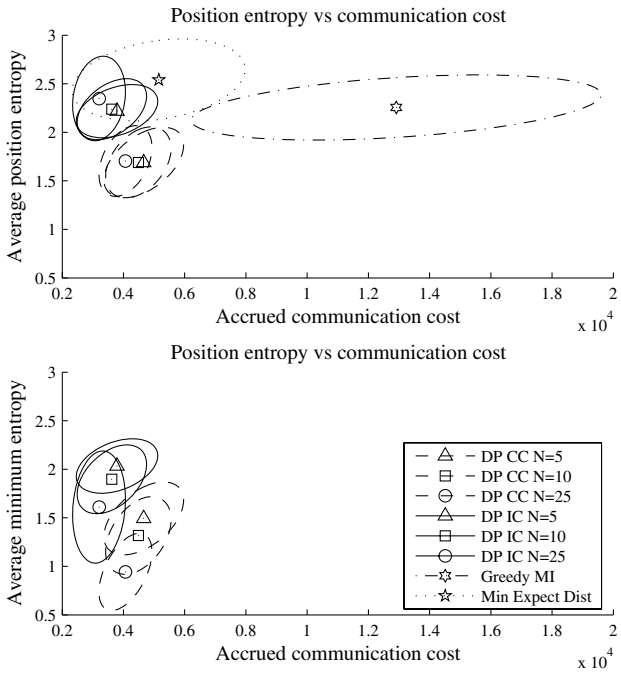

Fig. 2. Position entropy and communication cost for dynamic programming method with communication constraint (DP CC) and information constraint (DP IC) with different planning horizon lengths (N), compared to the methods selecting as leader node and activating the sensor with the largest mutual information (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 (and hence error bounds). 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 simulation results are summarized in Fig. 2. The top diagram demonstrates that the communication-constrained formulation provides a way of controlling sensor selection and leader node which reduces the communication cost substantially over the myopic single-sensor methods, and improves estimation performance substantially. The information-constrained formulation allows for an additional saving in communication cost while meeting an estimation criterion wherever possible. The diagram 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 is 180, the average communication cost if the constraint were always met with equality would be 1800. However, because

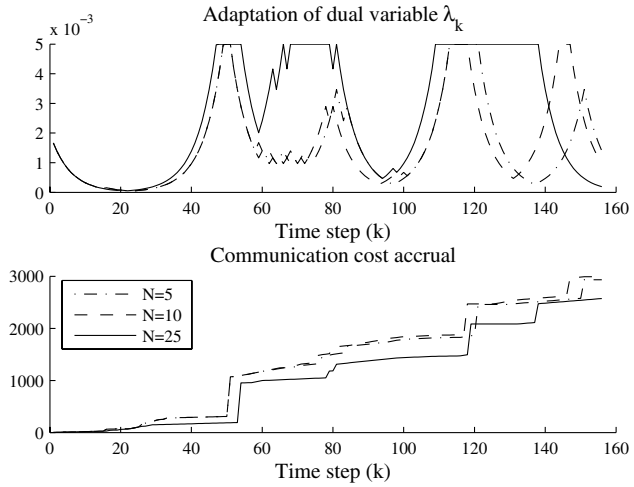

Fig. 3. Adaptation of communication constraint dual variable λ_k for different horizon lengths for a single Monte Carlo run, and corresponding cumulative communication costs.

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 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). The adaptation of the dual variable is shown in Fig. 3 for a single Monte Carlo run. A longer planning horizon allows for more consistent behavior, since there is a hand-off within the planning horizon for a larger portion of the time.

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 entropy is reduced, and additional communication energy is saved. The lower diagram in Fig. 2 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).

V. 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 trade-off between estimation performance and communication cost. The simulation results in Section IV demonstrate that approximations based on dynamic programming are able to provide similar entropy to that achieved using simple heuristics which consider estimation performance alone and utilize a single sensor, for a fraction of the communication cost. While the algorithm presented in Section III is computationally tractable on a modern computer, the constraints imposed by a wireless sensor network will typically require further simplification. The structure of the dynamic program developed in Section III provides a solid foundation which could be analyzed to develop rule-based approximations which maintain much of the performance benefit demonstrated in the empirical results. 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 which are less coarse than the linearized Gaussian model.

Acknowledgment

This work was supported in part by ODDR&E MURI through ARO grant DAAD19-00-0466 and MIT Lincoln Laboratory through ACC PO#3019934.

References

[1] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
[2] K. Pister, "Smart dust (keynote address)," in *IPSN '03*, April 2003.
[3] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, March 2002.
[4] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP Journal on Applied Signal Processing*, no. 4, pp. 378–391, 2003.
[5] M. Jones, S. Mehrotra, and J. H. Park, "Tasking distributed sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 243–257, 2002.
[6] J. L. Williams, J. W. Fisher III, and A. S. Willsky, "An approximate dynamic programming approach for communication constrained inference," in *Proc. IEEE Workshop on Statistical Signal Processing*, July 2005.
[7] D. A. Castañon, "Approximate dynamic programming for sensor management," in *Proc 36th Conference on Decision and Control*. IEEE, December 1997, pp. 1202–1207.
[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[9] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Arlington, VA: Navtech, 1994, vol. 2.
[10] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2000.
[11] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, March 2002.
[12] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293–313, 2002.
[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: John Wiley and Sons, 1991.
[14] E. Ertin, J. W. Fisher, and L. C. Potter, "Maximum mutual information principle for dynamic sensor query problems," in *Proc IPSN 2003*. Springer-Verlag, April 2003, pp. 405–416.
[15] C. M. Kreucher, A. O. Hero III, K. Kastella, and D. Chang, "Efficient methods of non-myopic sensor management for multitarget tracking," in *43rd IEEE Conference on Decision and Control*, December 2004.
[16] E. Altman, *Constrained Markov Decision Processes*. London, UK: Chapman and Hall, 1999.
[17] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.
[18] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Arlington, VA: Navtech, 1994, vol. 1.
[19] 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.