AN APPROXIMATE DYNAMIC PROGRAMMING APPROACH FOR COMMUNICATION CONSTRAINED INFERENCE

J. L. Williams *J. W. Fisher III* *A. S. Willsky*

Massachusetts Institute of Technology
{CSAIL, LIDS}
Cambridge, MA

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 the 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 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 target 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. Specifically, we consider tracking a single target and constrain the problem such that, at any time, a single sensor, referred to as the leader node, is activated to both sense and update the probabilistic model. The issue of selecting which sensor should be the leader at each time is of primary interest, as it directly impacts the trade-off between the estimation accuracy and the cost of communicating the probabilistic model from old leader node to new leader node. 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 find a tractable solution. Simulation results demonstrate that the resulting algorithm can provide similar estimation performance to that of the common most informative sensor election method at a fraction of the communication energy cost.

1. 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. While there are certain advantages to such approaches, they come with an additional complexity, namely the cost of transmitting the model from an old leader node to a new leader node. 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 to pure information driven approaches, comparable tracking performance can be obtained at a fraction of the communication energy cost.

Specifically, we consider a sensor network consisting of N_s sensors. 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

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

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.

2. PROBLEM STATEMENT

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 target state (i.e., position, velocity, etc) conditioned on the measurements. 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. A scheme which has been previously proposed [4] is to activate only a single sensor node (referred to as the leader node) at each time step, utilizing only this node for sensing, and maintaining the probabilistic model of target state at this sensor. Using this approach, the energy consumed in sensing is held constant (as exactly one node will be sensing in any one time step), and the energy spent on communications is limited to that due to handing off the probabilistic model from sensor to sensor. A question which naturally arises is how to select the leader node at each point in time.

2.1. Object dynamics and sensor models

In order to be concrete we consider specific object dynamics and sensor measurement models. However, we emphasize that the underlying principles have general applicability. Denoting $\boldsymbol{x}_k$ as the target state at time k, we assume that target 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 measurement taken by sensor $u \in \{1 : N_s\}$ (where N_s is the number of sensors) at time k as $\boldsymbol{z}_k^u$, a nonlinear measurement model is assumed:

$$\boldsymbol{z}_k^u \quad = \quad \boldsymbol{h}(\boldsymbol{x}_k, u) + \boldsymbol{v}_k^u \tag{2}$$

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

$$h(\boldsymbol{x}_k, u) \quad = \quad \frac{a}{||\mathbf{L}\boldsymbol{x}_k - \boldsymbol{l}^u||_2^2 + b} \tag{3}$$

where $\mathbf{L}$ is the matrix which extracts the position of the target from the target state (such that $\mathbf{L}\boldsymbol{x}_k$ is the location of the target), and $\boldsymbol{l}^u$ is the location of the u-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. (2), with variance r. The information provided by the measurement reduces as the range increases due to the nonlinearity.

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

$$\boldsymbol{z}_k^u \quad \approx \quad \boldsymbol{h}(\boldsymbol{x}^0, u) + \mathbf{H}^u(\boldsymbol{x}^0)(\boldsymbol{x}_k - \boldsymbol{x}^0) + \boldsymbol{v}_k^u \tag{4}$$

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

$$\quad = \quad \frac{-2a}{(||\mathbf{L}\boldsymbol{x}^0 - \boldsymbol{l}^u||_2^2 + b)^2}(\mathbf{L}\boldsymbol{x}^0 - \boldsymbol{l}^u)^T\mathbf{L} \tag{5}$$

where Eq. (5) is specific to Eq. (3) and will be used in the simulations presented in Section 4.

2.2. Estimation

Underlying a typical sensor network appication is the idea of coordinating many 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. (3) 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 target 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:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k}) \quad \approx \quad \sum_{i=1}^{N_p} w_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i) \tag{6}$$

We utilize an approximate Sequential Importance Sampling (SIS) algorithm [6] to represent $p(\boldsymbol{x}_{k+1}|\boldsymbol{z}_{0:k+1})$ 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. (2)) about the point $\mathbf{F}\boldsymbol{x}_k^i$. 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:

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i) \quad = \quad \mathcal{N}(\boldsymbol{x}_{k+1}; \mathbf{F}\boldsymbol{x}_k^i; \mathbf{Q}) \tag{7}$$

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

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 [6] 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. (2)) about the point $\mathbf{F}\boldsymbol{x}_k^i$, as described in Eq. (4). 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:

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i) \quad = \quad \mathcal{N}(\boldsymbol{x}_{k+1}; \mathbf{F}\boldsymbol{x}_k^i; \mathbf{Q}) \tag{8}$$

at time $(k+1)$. This distribution can be updated using the extended Kalman filter update equation [7] to obtain:

$$q(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i, \boldsymbol{z}_{k+1}) \quad = \quad \mathcal{N}(\boldsymbol{x}_{k+1}; \hat{\boldsymbol{x}}_{k+1}^i, \mathbf{P}_{k+1}^i) \tag{9}$$

where

$$\hat{\boldsymbol{x}}_{k+1}^i \quad = \quad \mathbf{F}\boldsymbol{x}_k^i + \mathbf{K}_{k+1}^i[\boldsymbol{z}_{k+1} - \boldsymbol{h}(\mathbf{F}\boldsymbol{x}_k^i, u)] \tag{10}$$

$$\mathbf{P}_{k+1}^i \quad = \quad \mathbf{Q} - \mathbf{K}_{k+1}^i \mathbf{H}^u(\mathbf{F}\boldsymbol{x}_k^i)\mathbf{Q} \tag{11}$$

$$\mathbf{K}_{k+1}^i \quad = \quad \mathbf{Q}\{\mathbf{H}^u(\mathbf{F}\boldsymbol{x}_k^i)\}^T \cdot \cdot [\mathbf{H}^u(\mathbf{F}\boldsymbol{x}_k^i)\mathbf{Q}\{\mathbf{H}^u(\mathbf{F}\boldsymbol{x}_k^i)\}^T + \mathbf{R}^u]^{-1} \tag{12}$$

A new particle $\boldsymbol{x}_{k+1}^i$ is drawn from the distribution in Eq. (9), and weighted by w_{k+1}^i, calculated by

$$w_{k+1}^i \quad = \quad c w_k^i \frac{p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}^i) p(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i)}{q(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i, \boldsymbol{z}_{k+1})} \tag{13}$$

where c is the normalization constant necessary to ensure that $\sum_{i=1}^{N_p} w_{k+1}^i = 1$, and $p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}^i) = \mathcal{N}\{\boldsymbol{z}_{k+1}; \boldsymbol{h}(\boldsymbol{x}_{k+1}^i, u), \mathbf{R}^u\}$. The resulting approximation for the distribution of $\boldsymbol{x}_{k+1}$ conditioned on the measurements $\boldsymbol{z}_{0:k+1}$ is:

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{z}_{0:k+1}) \quad \approx \quad \sum_{i=1}^{N_p} w_{k+1}^i \delta(\boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^i) \tag{14}$$

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

$$\boldsymbol{\mu}_k = \sum_{i=1}^{N_p} w_k^i \boldsymbol{x}_k^i, \ \mathbf{P}_k = \sum_{i=1}^{N_p} w_k^i (\boldsymbol{x}_k^i - \boldsymbol{\mu}_k)(\boldsymbol{x}_k^i - \boldsymbol{\mu}_k)^T \tag{15}$$

2.3. Communications

We assume that any sensor node can communicate with any other sensor node in the network, and that the cost of 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:

$$\tilde{C}_{ij} \quad \propto \quad ||\boldsymbol{l}^i - \boldsymbol{l}^j||_2^2 \tag{16}$$

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. (16) as arc lengths:

$$C_{ij} \quad = \quad \sum_{k=1}^{n} \tilde{C}_{i_{k-1} i_k} \tag{17}$$

where $\{i_0, \ldots, i_n\}$ is the shortest path from node $i = i_0$ to node $j = i_n$. The shortest path distances can be calculated using any shortest path algorithm, such as deterministic dynamic programming or label correcting methods [8]. 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}.

2.4. Sensor management

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. When the problem is constrained such that a single node is activated at any one time, the decision becomes one of when to transfer control from sensor to sensor, trading off the gain in estimation performance which results against the cost of the communications energy which is spent in transmitting the probabilistic model to the new leader node. The first task in optimizing the estimation performance of a system is to decide upon an objective function which measures that performance. Recent research [9, 10] 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 the sensor whose measurement minimizes the expected posterior entropy:

$$u_k = \arg\min_u H(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^u) \tag{18}$$

The conditioning in Eq. (18) is on the *value* of the previous measurements $\boldsymbol{z}_{0:k-1}$, and on the *random variable* corresponding to the new measurement $\boldsymbol{z}_k^u$, implying marginalization over all possible values of the measurement. The mutual information between the target state $\boldsymbol{x}_k$ and measurement $\boldsymbol{z}_k^u$ conditioned on the previous measurement history is defined as [11]:

$$I(\boldsymbol{x}_k; \boldsymbol{z}_k^u|\boldsymbol{z}_{0:k-1}) \quad = \quad H(\boldsymbol{z}_k^u|\boldsymbol{z}_{0:k-1}) - H(\boldsymbol{z}_k^u|\boldsymbol{x}_k) \tag{19}$$

$$\quad = \quad H(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}) - H(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k^u) \tag{20}$$

Since the first term in Eq. (20) is independent of the control decision (u), it is clear that minimization of expected conditional entropy is equivalent to maximization of mutual information [12]. 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*, as they only consider information which is obtainable in the next time step. Situations commonly arise (e.g., [13]) in which alternative strategies have poorer performance in the next time step but better performance over several time steps.

3. DYNAMIC PROGRAMMING SENSOR MANAGEMENT ALGORITHM

As discussed in Section 2.4, the estimation performance obtained using the minimum expected posterior entropy algorithm Eq. (18) is potentially inferior to the performance achievable by an algorithm which plans over multiple time steps. In practice, this difference is often small when information is the sole contributor to the objective function. This situation changes greatly when other elements such as communication cost are incorporated into the objective. While a particular leader node transition may lead to improved estimation performance over many time steps, the communication cost may be large, hence the cost of the transmission must be considered in comparison with the time-aggregated estimation gain which will result. Accordingly, it is necessary to consider several time steps when selecting the leader node.

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 choose to maximize the information obtained from the measurements made by the leader node subject to a constraint on the expected communication cost incurred by transmitting the model when the leader node changes. This can be formulated as a dynamic program. Similarly to imperfect state information problems [8], the *dynamic programming* state is the PDF of *target* state conditioned on previous controls and measurements. Throughout the following we make the definition

$$\mathbb{X}_k \triangleq p(\boldsymbol{x}_k|u_{0:k-1}, \boldsymbol{z}_{0:k-1}) \tag{21}$$

and we treat $\mathbb{X}_k$ as the dynamic programming state.[2] Following the discussion in Section 2.4, we utilize mutual information as our objective, and define

$$g(\mathbb{X}_k, u_k) = -I(\boldsymbol{x}_k; \boldsymbol{z}_k^{u_k}|u_{0:k-1}, \boldsymbol{z}_{0:k-1}) \tag{22}$$

[2]Note that in contrast to the convention of [8], the measurement at time k, $\boldsymbol{z}_k$, is received after the control at time k, u_k, has been applied, and the distribution $p(\boldsymbol{z}_k|\boldsymbol{x}_k, u_k)$ depends upon the value of the control applied at time k.

so that $-g(\mathbb{X}_k, u_k)$ is the expected single-stage reward. The dynamic program for minimizing Eq. (22) over the N-step rolling horizon $\{k : k+N-1\}$ subject to a communication constraint can be described by the following recursive reward-to-go function:

$$\begin{aligned} J_l(\mathbb{X}_l) &= \max_{u_l} \Big\{ -g(\mathbb{X}_l, u_l) \\ &\quad + \mathop{\mathrm{E}}_{\boldsymbol{z}_l|\mathbb{X}_l,u_l} J_{l+1}(\mathbb{X}_l, u_l, \boldsymbol{z}_l) \Big\}, \\ &\quad l \in \{k : k+N-1\} \\ J_{k+N}(\mathbb{X}_{k+N}) &= 0 \end{aligned} \tag{23}$$

subject to the communication constraint:

$$\mathrm{E}\left\{ \sum_{l=k}^{k+N-1} C_{u_{l-1}u_l} \right\} \leq C_{\max} \tag{24}$$

The arguments of J_{l+1} in Eq. (23) collectively define $\mathbb{X}_{l+1}$ through the following expression:

$$\mathbb{X}_{l+1} \triangleq p(\boldsymbol{x}_{l+1}|u_{0:l-1}, \boldsymbol{z}_{0:l-1}, u_l, \boldsymbol{z}_l) \tag{25}$$

which is calculated from $\mathbb{X}_l$ and $(u_l, \boldsymbol{z}_l)$ using the recursive Bayes update described in Section 2.2 (u_l is not a random variable, but rather a parameter which indexes the probabilistic model which relates the measurement $\boldsymbol{z}_l$ to the state $\boldsymbol{x}_l$).

The expectation in the communication constraint of Eq. (24) 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 dual variable (i.e., a Lagrange multiplier) λ and solve the related problem

$$J_k(\mathbb{X}_k) = \max_{\lambda \geq 0} \tilde{J}_k(\mathbb{X}_l, \lambda) \tag{26}$$

$$\begin{aligned} \tilde{J}_l(\mathbb{X}_l, \lambda) &= \min_{u_l} \big\{ g(\mathbb{X}_l, u_l) \\ &\quad + \mathop{\mathrm{E}}_{\boldsymbol{z}_l|\mathbb{X}_l,u_l} \tilde{J}_{l+1}(\mathbb{X}_l, u_l, \boldsymbol{z}_l, \lambda) \big\}, \\ &\quad l \in \{k : k+N-1\} \\ \tilde{J}_{k+N}(\mathbb{X}_{k+N}, \lambda) &= \lambda \left\{ \sum_{l=k}^{k+N-1} C_{u_{l-1}u_l} - C_{\max} \right\} \end{aligned} \tag{27}$$

where we have changed Eq. (23) to an equivalent minimization to align the above expression with the convention. For the above expression to be valid in its current form, the dynamic programming state $\mathbb{X}_l$ must be redefined to incorporate the history of controls, $u_{k-1:l}$, such that the required variables are present in the argument of $\tilde{J}_{k+N}(\mathbb{X}_{k+N}, \lambda)$. Equivalently, we redefine the dynamic programming state to explicitly include the most recently applied control, and redistribute the elements of the sum throughout the cost function recursion, so that the cost of control decisions is

modelled within the per-stage cost wherever possible, rather than through the cost-to-go function:

$$\begin{aligned} \mathbb{X}_k &\triangleq (p(\boldsymbol{x}_k|u_{0:k-1}, \boldsymbol{z}_{0:k-1}), u_{k-1}) \quad (28) \\ \tilde{J}_l(\mathbb{X}_l, \lambda) &= \min_{u_l} \big\{ g(\mathbb{X}_l, u_l) + \lambda C_{u_{l-1}u_l} \\ &\quad + \mathop{\mathrm{E}}_{\boldsymbol{z}_l|\mathbb{X}_l,u_l} \tilde{J}_{l+1}(\mathbb{X}_l, u_l, \boldsymbol{z}_l, \lambda) \big\}, \\ &\quad l \in \{k : k+N-1\} \\ \tilde{J}_{k+N}(\mathbb{X}_{k+N}, \lambda) &= -\lambda C_{\max} \quad (29) \end{aligned}$$

The optimization of Eq. (26) provides a lower bound to the minimum value of the original constrained problem; the presence of a duality gap is possible as the optimization space is discrete. The size of the duality gap is

$$\lambda \, \mathrm{E} \left\{ \sum_{l=k}^{k+N-1} C_{u_{l-1}u_l} - C_{\max} \right\} \quad (30)$$

If it happens that the optimal solution produced by the dual problem of Eq. (26) satisfies complementary slackness, i.e., that the duality gap of Eq. (30) evaluates to zero, then the solution produced by the optimal value of Eq. (26) 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 communication constraint, or the solution yields a result for which the communications constraint is tight. If a duality gap exists, a better solution may exist satisfying the communication constraint; however, the solution returned would be optimal if the maximum communication cost $C_{\max}$ was reduced to make the communication constraint tight. The problem in Eq. (26) is the Lagrangian relaxation of the original constrained optimization, which is a common approximation method for discrete optimization problems.

The formulation of the energy constraint using a Lagrange multiplier is similar to [14], although in this case the dual problem does not have an easy solution for a given value of λ. In practice, we set the dual variable λ to a fixed value, and view the above development as motivation for using a linear combination of the information reward and communication cost as our overall objective. The algorithm could be made more robust by dynamically adjusting the dual variable: if the communication constraint is being met with excess, then the value could be lowered, and if it is being exceeded then the value could be raised.

4. SIMULATION RESULTS

The model presented in Section 2.1 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; sensor layouts which were obviously degenerate, such as layouts where sensors were clustered in one portion of the region, were qualitatively eliminated. The initial position of the target 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 sample time was $T = 0.25$ sec, diffusion strength was $q = 10^{-2}$, and the measurement model parameters were $a = 2000$, $b = 100$ and $r = 1$. The planning horizon was set to $N = 30$ time steps.

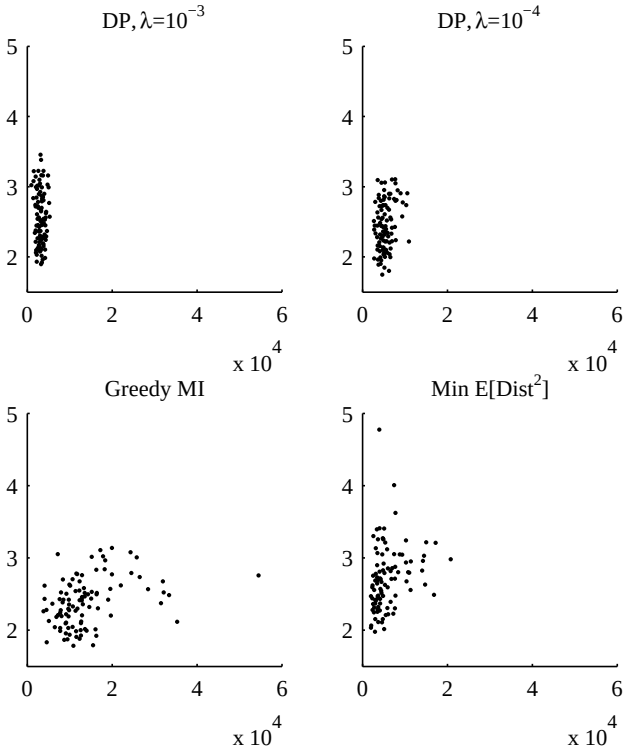

Fig. 1. Scatter plot of average position entropy (y-axis) versus accrued communication cost (x-axis) for dynamic programming method with λ set to 10^{-3} and 10^{-4}, greedy maximum mutual information method, and minimum expected square distance method.

The simulation results are detailed in Fig. 1 which shows the average position entropy versus the communication cost accrued for each simulation, for the approximate dynamic programming algorithm described with λ set to 10^{-3} and 10^{-4}, and the greedy maximum mutual information and heuristic minimum expected square distance algorithms described in Section 2.4. The scatter plots demonstrate that the dynamic programming method achieves similar position entropy to the other methods for a substantially reduced communication cost. The dynamic programming method achieves similar

entropy performance to the maximum mutual information algorithm for a fraction of the communication cost, and significantly better entropy performance than the minimum expected square distance heuristic, again for a substantially lower communication cost.

5. CONCLUSION AND FUTURE WORK

The analysis in Section 3 demonstrates that dynamic programming provides a principled approach to the problem of sensor management in an energy-constrained sensor network. The simulation results in Section 4 demonstrate that approximations based on dynamic programming are able to provide similar entropy to that achieved using simple heuristics which consider estimation performance alone, for a fraction of the communication cost.

The approach constrains the problem such that a single node senses and maintains the probabilistic model at each point in time. Relaxing these constraints is the subject of future work. Extending the dynamic programming formulation to allow simultaneous utilization of multiple sensors when necessary, and distributing the probabilistic model across multiple sensors is of particular interest.

6. 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] 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.

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

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] D. A. Castañon, "Approximate dynamic programming for sensor management," in *Proc 36th Conference on Decision and Control*. IEEE, December 1997, pp. 1202–1207.