OPTIMIZATION APPROACHES TO DYNAMIC ROUTING OF MEASUREMENTS AND MODELS IN A SENSOR NETWORK OBJECT TRACKING PROBLEM

Jason L. Williams,[1] *John W. Fisher III,*[2] *Alan S. Willsky*[3]

[1,3]MIT/LIDS
Cambridge MA 02139
{jlwil,willsky}@mit.edu

[2] MIT/CSAIL
Cambridge MA 02139
fisher@csail.mit.edu

ABSTRACT

Inter-sensor communication often comprises a significant portion of energy expenditures in a sensor network as compared to sensing and computation. We discuss an integrated approach to dynamically routing measurements and models in a sensor network. Specifically, we examine the problem of tracking objects within a region wherein the responsibility for combining measurements and updating a posterior state distribution is assigned to a single sensor at any given time step. The so called leader node may change over time. Sensor nodes communicate for two reasons: firstly, measurements of target state are transmitted from sensors to the current leader node for incorporation into the state estimate model; secondly, the state model is transmitted between sensors when the leader node changes. The trade-off between these two types of communication is of primary importance to dynamic selection of the leader node. We propose an algorithm based on a dynamic programming roll-out formulation of the minimum cost problem. We obtain a cost function which can be efficiently minimized by simplifying the problem to that of an open loop feedback controller which is an upper bound to the optimal cost. We present empirical results which compare methods previously proposed in the literature to the algorithm presented here.

1. INTRODUCTION

Energy is a limited resource in many sensor networks. It is often the case that inter-sensor communication costs are greater by orders of magnitude than local computation and sensing costs with respect to energy expenditures [1, 2]. Motivated by the need for integrated strategies which trade off energy conservation with sensing needs, we discuss two approximate dynamic programming approaches for routing measurements and state distributions in a distributed sensor network. While the methodology is general, we focus on object tracking for concreteness.

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

Quantities of interest (*i.e.*, kinematic state) in object tracking are inferred largely from sensor measurements which are in proximity to the object (*e.g.*, see [3]). Consequently, a variety of approaches [4, 5] designate the responsibility of integrating measurements to one sensor node (*i.e.*, the leader node) in the network. Over time the leader node changes dynamically as a function of the kinematic state of the object. Along with advantages comes the additional complexity of transmitting the representation of the state distribution from the current leader node to the next. In this paper we examine the problem of determining when a change in leader node is necessary in the context of object tracking in distributed sensor networks. For purposes of addressing the primary sensor resource management issue, we restrict ourselves to 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 FORMULATION

The role of the leader node is to compute a representation of the posterior distribution of the object's kinematic state conditioned on the received measurements. In the absence of energy constraints, the optimal solution is to incorporate the measurements of *all* sensors in the network. In the face of energy constraints, optimal approaches generally incorporate a subset of sensor measurements into the state distribution. The problem of jointly maximizing information gain and minimizing communication cost is a difficult optimization problem and is considered in [6]. Here, we address dynamic leader node assignment for a case when multiple sensor measurements are used and the complexity of the state distribution is assumed fixed at each time step.

2.1. Object dynamics and sensor models

While the optimization approach described in Section 3 is generally applicable, we use specific object dynamics and measurement models to clarify the discussion. Denoting the object state at time k as $\boldsymbol{x}_k$, and the measurement taken by sensor $u \in \{1 : N_s\}$ (N_s is the number of sensors) at time k as $\boldsymbol{z}_k^u$, the state dynamics and nonlinear measurement models are described as:

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \mathbf{F}\boldsymbol{x}_k + \boldsymbol{w}_k & (1)\\ \boldsymbol{z}_k^u &= \boldsymbol{h}(\boldsymbol{x}_k, u) + \boldsymbol{v}_k^u & (2)\\ h(\boldsymbol{x}_k, u) &= \frac{a}{||\mathbf{L}\boldsymbol{x}_k - \boldsymbol{l}^u||_2^2 + b} & (3)\end{aligned}$$

where $\boldsymbol{w}_k \sim \mathcal{N}\{\boldsymbol{w}_k; \mathbf{0}, \mathbf{Q}\}$ and $\boldsymbol{v}_k^u \sim \mathcal{N}\{\boldsymbol{v}_k^u; \mathbf{0}, \mathbf{R}^u\}$[1] are independent white Gaussian noise processes and $\mathbf{F}$ and $\mathbf{Q}$ are known matrices. The state is comprised of position and velocity in two dimensions ($\boldsymbol{x}_k = [p_x\ v_x\ p_y\ v_y]^T$). Velocity is modelled as a random walk with constant diffusion strength q (independently in each dimension), while position is the integral of velocity. The corresponding discrete-time model can be calculated using the methods detailed in [7]. The matrix $\mathbf{L}$ extracts the position variables of the state and $\boldsymbol{l}^u$ is the location of the uth 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). Due to the nonlinearity in $h(\boldsymbol{x}_k, u)$, measurements from sensors in close proximity to the object are more informative than those from distant sensors.

2.2. 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). We model the cost of direct communication between two nodes as proportional to the square distance between the two sensors, $\tilde{C}_{ij} \propto ||\boldsymbol{l}^i - \boldsymbol{l}^j||_2^2$. 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 $\tilde{C}_{ij}$ as arc lengths:

$$C_{ij} = \sum_{k=1}^{n} \tilde{C}_{i_{k-1}i_k} \qquad (4)$$

where $\{i_0, \dots, i_n\}$ is the shortest path from node $i = i_0$ to node $j = i_n$. We omit discussion of methods for determining shortest path distances (*e.g.*, see [8]).

2.3. Sensor management

Selecting the subset of sensors which actively sense and transmit their measurements to the leader node involves a fundamental trade-off between estimation performance, and the energy consumed in achieving that performance. Given the sensor measurement model of Section 2.1, a natural heuristic is to set a threshold for transmitting measurements, such that the measurement from sensor u is transmitted if and only if $z_k^u \geq \eta$. We assume that the threshold η is fixed to a value such that the probability that a measurement from a sensor far from the object will exceed η is small. Even with this assumption, the expected cost of transmitting measurements from sensors far from the object will be significant, since the cost of transmitting a measurement increases with range, and the probability of transmission is roughly fixed. To avoid unnecessary false alarms and conserve energy, sensors are deactivated if $P(z_k^u \geq \eta) < \epsilon$. This simplified formulation yields an optimization problem in which it is possible to consider communication costs aggregated over multiple time steps, as discussed in Section 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 $\mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}, \mathbf{P}\} = |2\pi\mathbf{P}|^{-\frac{1}{2}} \exp\{-0.5(\boldsymbol{x} - \boldsymbol{\mu})^T\mathbf{P}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\}$.

3. DYNAMIC LEADER NODE SELECTION

Using the scheme described in Section 2.3, we examine the problem of which node should be selected as leader. This process involves a trade-off between the cost of transmitting measurements to the leader, and the cost of transmitting the state distribution to a new leader node. We neglect the cost of activating and deactivating sensors, as this transmission contains only a single bit of information, and occurs infrequently (sensors are activated when the object enters their vicinity, and deactivated when it leaves).

Combining the cost of Eq. (4), the probability of measurement transmission from Section 2.3, and assuming constant data per measurement, the expected cost for a single sensor s to send its measurement to the leader node u_k is proportional to:

$$\tilde{g}_s(\mathbb{X}_k, u_k) = C_{u_k s} \mathop{\mathrm{E}}_{\boldsymbol{x}_k|\boldsymbol{Z}_{0:k-1}} P_d^s(\boldsymbol{x}_k) \qquad (5)$$

where $\mathbb{X}_k$ denotes conditional probabilistic model of object state, $p(\boldsymbol{x}_k|\boldsymbol{Z}_{0:k-1})$ ($\boldsymbol{Z}_{0:k-1}$ denotes all measurements received up to and including time $(k-1)$), which is a sufficient statistic for the dynamic programming state [8], and $P_d^s(\boldsymbol{x}_k)$ is the probably of detecting the object (*i.e.*, that $z_k^s \geq \eta$) if its true state is $\boldsymbol{x}_k$. Incorporating the effect of the sensor deactivation, we obtain an expected communication cost for sensor s of:

$$g_s(\mathbb{X}_k, u_k) = \begin{cases} \tilde{g}_s(\mathbb{X}_k, u_k), & \mathrm{E}_{\boldsymbol{x}_k|\boldsymbol{Z}_{0:k-1}} P_d^s(\boldsymbol{x}_k) \geq \epsilon \\ 0, & \mathrm{E}_{\boldsymbol{x}_k|\boldsymbol{Z}_{0:k-1}} P_d^s(\boldsymbol{x}_k) < \epsilon \end{cases} \qquad (6)$$

Assuming that the probabilistic model of object state consists of fixed amount of data, the cost of transmitting the probabilistic model from the existing leader node u to a new leader node $\tilde{u}$ is $R \cdot C_{u\tilde{u}}$, where R is the ratio of the number of bits in the probabilistic model to the number of bits in a measurement. We commence by defining the base control policy [8] $\tilde{\mu}(\mathbb{X}_l)$, which selects as leader node the sensor which has the smallest expected square distance to the object. The cost of employing this policy over M steps starting from time step $(k+N)$ can be calculated as:

$$
\begin{aligned}
J_l^{\tilde{\mu}}(\mathbb{X}_l, u_{l-1}) = \sum_{s=1}^{N_s} g_s(\mathbb{X}_l, \tilde{\mu}(\mathbb{X}_l)) + R \cdot C_{u_{l-1}\tilde{\mu}(\mathbb{X}_l)} \\
+ \underset{\boldsymbol{Z}_l|\boldsymbol{Z}_{0:l-1}}{\mathrm{E}} J_{l+1}^{\tilde{\mu}}(\mathbb{X}_{l+1}, \tilde{\mu}(\mathbb{X}_l)), \\
l \in \{k+N : k+N+M-1\} \quad (7)
\end{aligned}
$$

where $\boldsymbol{Z}_l$ denotes all sensor measurements received at time l, and $\mathbb{X}_{l+1}$ incorporates $\mathbb{X}_l$ and $\boldsymbol{Z}_l$ through a Bayes update operation. The recursion of Eq. (7) terminates with $J_{k+M+N}^{\tilde{\mu}}(\mathbb{X}_{k+M+N}, u_{k+M+N-1}) = 0$. Note that the only dependence of $J_l^{\tilde{\mu}}(\mathbb{X}_l, u_{l-1})$ on u_{l-1} is through the term $R \cdot C_{u_{l-1}\tilde{\mu}(\mathbb{X}_l)}$, hence we can define $\tilde{J}_l(\mathbb{X}_l) = J_l^{\tilde{\mu}}(\mathbb{X}_l, u_{l-1}) - R \cdot C_{u_{l-1}\tilde{\mu}(\mathbb{X}_l)}$ which will have no dependence on u_{l-1}. We utilize this base policy to construct an N-step roll-out [8] described at time k by the following recursive equation:

$$
\begin{aligned}
J_l(\mathbb{X}_l, u_{l-1}) = \min_{u_l} \Bigg\{ \sum_{s=1}^{N_s} g_s(\mathbb{X}_l, u_l) + R \cdot C_{u_{l-1}u_l} \\
+ \underset{\boldsymbol{Z}_l|\boldsymbol{Z}_{0:l-1}}{\mathrm{E}} J_{l+1}(\mathbb{X}_{l+1}, u_l) \Bigg\}, \\
l \in \{k : k+N-1\} \quad (8)
\end{aligned}
$$

The recursion of Eq. (8) terminates with

$$
\begin{aligned}
J_{k+N}(\mathbb{X}_{k+N}, u_{k+N-1}) = J_{k+N}^{\tilde{\mu}}(\mathbb{X}_{k+N}, u_{k+N-1}) \\
= \tilde{J}_{k+N}(\mathbb{X}_{k+N}) + R \cdot C_{u_{k+N-1}\tilde{\mu}(\mathbb{X}_{k+N})} \quad (9)
\end{aligned}
$$

Noting that the control choices $u_{k:k+N-1}$ do not affect the probabilistic state $\mathbb{X}_{k+N} = p(\boldsymbol{x}_{k+N}|\boldsymbol{Z}_{0:k+N-1})$ (since sensor management and state estimation are invariant to the leader node selection), we can define $\hat{J}_l(\mathbb{X}_l, u_{l-1})$ using the same recursion as Eq. (8), terminating with $\hat{J}_{k+N}(\mathbb{X}_{k+N}, u_{k+N-1}) = R \cdot C_{u_{k+N-1}\tilde{\mu}(\mathbb{X}_{k+N})}$, and we will have

$$
\hat{J}_l(\mathbb{X}_l, u_{l-1}) = J_l(\mathbb{X}_l, u_{l-1}) + \underset{\boldsymbol{Z}_{l:k+N-1}|\boldsymbol{Z}_{0:l-1}}{\mathrm{E}} \tilde{J}_{k+N}(\mathbb{X}_{k+N}) \quad (10)
$$

and the choice of controls to produce the optimum will be identical. Noting that this holds for *any* value of M (the number of steps over which the expected cost of the base heuristic is evaluated), we view the control policy associated with $\hat{J}$ as the N-step roll-out of the infinite horizon base policy $J^{\tilde{\mu}}$.[2] The dynamic programs described by J and $\hat{J}$ possess a cost improvement property, *i.e.*, the cost of the control policy obtained using the roll-out policy is guaranteed to not exceed the cost of the base policy. Similarly, the cost of the roll-out with more steps is guaranteed not to exceed the cost of the roll-out with fewer steps.

[2]For this view to be rigorous, we must have $J^{\tilde{\mu}}$ finite for all states. If the sensor field is finite, and we can guarantee that the object will leave the coverage region after a certain number of steps, we can see that such an assumption is reasonable.

The dynamic programs for J and $\hat{J}$ have an infinite number of states, hence approximate methods must be used. In order to enable evaluation over a large planning horizon, we choose two common suboptimal approximations: the Open Loop Feedback Controller (OLFC), and the Partially Stochastic Certainty Equivalent Controller (PSCEC) [8]. At each stage the OLFC calculates a control trajectory (*i.e.*, a plan which node will be leader at each step) for the next N stages, neglecting the fact that the controller will have opportunity to change its decisions in the interim, after further measurements are received. The objective which is minimized by the OLFC can be seen to be an upper bound to the true dynamic program cost function: applying the bound $\mathrm{E}_x \min_i f(x,i) \leq \min_i \mathrm{E}_x f(x,i)$, and neglecting the impact of sensor deactivation, it can be shown (see [6] for details) that an N-step rollout of Eq. (8) has the following upper bound:

$$
\begin{aligned}
\hat{J}_k(\mathbb{X}_k, u_{k-1}) \lesssim \min_{u_{k:k+N-1}} \Bigg\{ \sum_{l=k}^{k+N-1} \Bigg\{ R \cdot C_{u_{l-1}u_l} \\
+ \sum_{s=1}^{N_s} C_{u_l s} \underset{\boldsymbol{x}_l|\boldsymbol{Z}_{0:k-1}}{\mathrm{E}} P_d^s(\boldsymbol{x}_l) \Bigg\} \\
+ \underset{\boldsymbol{Z}_{k:k+N-1}|\boldsymbol{Z}_{0:k-1}}{\mathrm{E}} R \cdot C_{u_{k+N-1}\tilde{\mu}(\mathbb{X}_{k+N})} \Bigg\} \quad (11)
\end{aligned}
$$

The upper bound is approximate since the impact of sensor deactivation has been neglected; in practice this will be performed approximately using the test $\mathrm{E}_{\boldsymbol{x}_l|\boldsymbol{z}_{0:k-1}} P_d^s(\boldsymbol{x}_l) \gtreqless \epsilon$ rather than $\mathrm{E}_{\boldsymbol{x}_l|\boldsymbol{z}_{0:l-1}} P_d^s(\boldsymbol{x}_l) \gtreqless \epsilon$ (*i.e.*, neglecting the impact of measurements received between time k and time $(l-1)$). We approximate the final expectation by $C_{u_{k+N-1}\tilde{\mu}(\bar{\mathbb{X}}_{k+N})}$, where $\bar{\mathbb{X}}_{k+N} = p(\boldsymbol{x}_{k+N}|\boldsymbol{Z}_{0:k-1})$. Eq. (11) can be solved easily as a finite state deterministic dynamic program, where the state is the leader node at the previous stage.

Using the alternative approximation of the PSCEC algorithm, we design the control law assuming that the state (*i.e.*, object location) is known exactly at each time step, and then implement the control law using the mean or maximum likelihood estimate of the state. Using a particle approximation for the PDF of object state, one can view the problem structure as a finite state stochastic dynamic program, where the state at time k is the combination of the location of the object at time k (discretized to N_p values hypothesized by the particle filter approximation), and the leader node at time $(k-1)$. To determine the possible locations of the object, the particle filter representation is propagated through the

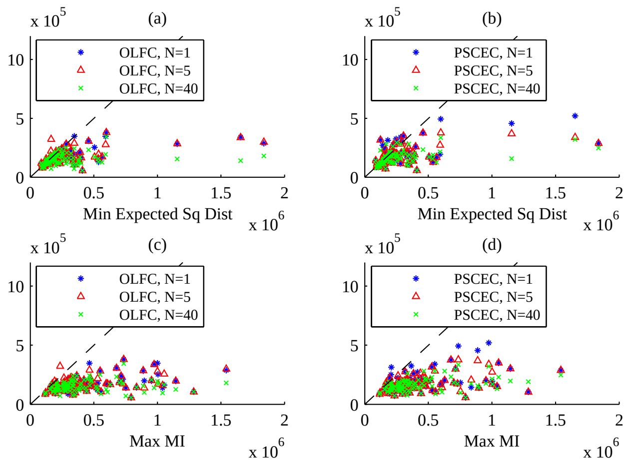

Fig. 1. Comparison of total communication cost. Each point represents a single Monte Carlo simulation; the x-axis value shows communication cost for the reference algorithm (MESD in (a) and (b), and MMI in (c) and (d)), while the y-axis value shows the communication cost of the test algorithm, as listed in the legend.

simulation model of the system dynamics. Transition probabilities describing the distribution of object state at time $(l+1)$ conditioned on the state at time l can be derived using importance sampling ideas such as [9].

4. SIMULATION RESULTS

The model presented in Section 2.1 was simulated for 100 Monte Carlo trials using 20 sensors positioned randomly under 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/sec in each dimension, moving into the region. The simulation ends when the object leaves the 100×100 region or after 200 time steps. The sample time was 0.25 sec, the measurement model parameters were $a = 2000$, $b = 100$ and $r = 1$, and the ratio of the model transmission cost to the measurement transmission cost was $R = 64$. Experiments were performed with different values of R, and the same basic structure remained.

The performance measure was the total communication energy expended up until 60 simulation steps before the end of the simulation, chosen to avoid edge effects associated with the object leaving the region populated with sensors. Performance was compared against two heuristic methods: firstly, a method which selects as leader node the sensor with the Minimum Expected Square Distance (MESD) from the object, and secondly, a method which selects as leader node the sensor with the Maximum Mutual Information (MMI) between its measurement and the object state conditioned on previous measurements.

The results shown in Fig. 1 and Table 1 compare the communication cost expended by the various algorithms. The data demonstrates that the dynamic programming methods are able to provide a moderate reduction in communication cost over the simple MESD heuristic, and a more substantial improvement over the MMI method. The major benefit of the dynamic programming methods was obtained with a lookahead of a single time step; longer lookahead values yielded small improvements. With a single lookahead step, the dynamic programming methods continually evaluate whether it is preferable to transfer the model to the node chosen by the base policy (the closest node) at the current time step, or to wait until the next time step. The results indicate that this action captures a significant proportion of the energy saving achievable through planning in this class of problem.

Lookahead	1	3	5	10	20	40	**Heuristics**	
OLFC	1.68	1.67	1.70	1.65	1.54	1.51	**MESD**	2.70
PSCEC	1.85	1.83	1.75	1.67	1.65	1.64	**MMI**	4.44

Table 1. Average communication cost ($\times 10^5$).

5. CONCLUSION AND FUTURE WORK

The analysis in Section 3 demonstrates that dynamic programming provides a principled approach to the problem of leader node selection in an energy-constrained sensor network. The simulation results in Section 4 demonstrate that tractable approximations of dynamic programming are able to substantially reduce the communications cost incurred in tracking a object as it passes through a field of sensors.

References

[1] G.J. Pottie and W.J. 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, *IPSN '03*.

[3] D. Li, K.D. Wong, Y.H. Hu, and A.M. Sayeed, "Detection, classification, and tracking of targets," *Signal Processing*, 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," *Int 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, "Dynamic prog approaches to sensor management in sensor networks," Tech. Rep. LIDS-TR-2637, Mass Inst of Tech, 2005.

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

[8] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA, second edition, 2000.

[9] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, 2004.