Coordinated Sampling in Communication Constrained Sensor Networks using Markov Decision Processes

Shuping Liu[1] Anand Panangadan[2] Ashit Talukder[1,2,3] Cauligi S. Raghavendra[1]

[1]University of Southern California,Ming Hsieh Department of Electrical Engineering, Los Angeles, CA 90089, USA, 1-213-821-0871, {lius,raghu}@usc.edu

[2] Childrens Hospital Los Angeles, 4650 Sunset Blvd., Los Angeles, CA 90027, USA, 1-323-361-2413, APanangadan@chla.usc.edu

[3]Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109, USA, 1-818-354-1000, Ashit.Talukder@jpl.nasa.gov

Abstract. The paper describes a Markov Decision Process (MDP) framework for coordinated sensing and adaptive communication in embedded sensor networks. The technique enables distributed sensor nodes to adapt their sampling rates in response to changing event criticality and the availability of resources (energy) at each node. The relationship between energy consumption, sampling rates, and utility of coordinated measurements is formulated as a stochastic model. The resulting model is solved as an MDP to generate a globally optimal policy that specifies the sampling rates for each node for all possible states of the system. This policy is computed offline before deployment and only the resulting policy is stored within each node. The on-line computational cost of executing the policy is minimal since it involves only a lookup of the optimal policy table. The optimal policy is computed under the assumption that the state of all sensors is completely observable. During execution, each sensor maintains a local estimate of the other sensor's states. Sensors exchange their true local states when an information theoretic model of the uncertainty in the local state estimates exceeds a pre-defined threshold. Thus, the communication cost of executing a global policy is incurred only when a relatively large gain in the accuracy of the global state estimate is expected. We show results on simulated data that demonstrate the efficacy of this distributed control framework, the effect that the various model parameters have on the generated control policy, and compare the performance of the proposed controller with other policies.

Keywords

Human health monitoring, body sensor network, control

1 Introduction

Wireless sensor networks are being increasingly used in scientific and defense applications. In these applications, multiple homogeneous or heterogeneous sensors make simultaneous measurements of their environment to obtain better estimates than is possible from one sensor alone. However, energy and computation capacity still remain the most significant bottlenecks in practical deployability of sensor networks. Limitations of battery capacity and the high energy cost of radio transmission, necessitate energy efficient communication and on-board processing to extend system lifetime in practically useable scenarios [1].

We consider a mobile continuous health monitoring body sensor network where size and ergonomic constraints severely limit the availability of power to battery sources that can be recharged only every few days or weeks. Our body sensor network hardware system is described in [2, 3] and is based on work being done at the Children's Hospital Los Angeles and the University of Southern California [2, 4]. In this application, a patient's vital signs are continuously measured by using multiple physiological and metabolic sensors attached to the patient's body (also called a "body sensor network"). Unlike other sensor network applications, only a relatively few sensors (less than 5) are attached to a person. These sensors typically

take measurements at a relatively low rate when the sensor readings are within normal limits, with higher sampling rates during critical periods. An important performance metric in this application is the life time of the body sensor network. The energy reserves within a sensor node get depleted with increasing amounts of sensing, communication, and computation. For instance, some physiological sensors have electromechanical components that make sensing an energy-intensive process [3]. Ideally, the system of sensors should operate for an extended period while maintaining an acceptable level of sensor performance.

We use a Markov Decision Process (MDP) as a formalism to determine a *coordinated* sampling policy for the sensor network such that the system lifetime is extended without compromising detection and monitoring of critical and/or life threatening events. This computation of the coordinated policy is completed before nodes are deployed and only the resulting policy is stored within each sensor node. In this way, though the computational cost of computing a sophisticated policy off-line may be high, only a simple table lookup is sufficient to execute the policy on-line. The MDP formulation represents the relevant features of the sensing system (energy consumption rates, expected changes in event criticality) as part of a stochastic model. The energy reserves, control time, and event criticality are represented as discrete variables. These variables define the state of the system at any time. In addition, the utility of coordinated sampling and the penalty for running out of energy is quantified into a reward associated with each state. The objective is for the system to operate for a minimum specified length of time without the sensors running out of power while optimizing the reward function for health monitoring. The model is "solved" to determine a policy which specifies the sampling rate for each sensor for every possible state of the system (i.e., for different amounts of energy reserves at a sensor node). The policy obtained in this manner is optimal under the assumptions of the underlying model.

The optimal policy is calculated offline with the assumption of full observation, i.e., it assumes that the energy reserves at any sensor is known by all other sensors. However, during execution the world is only partially observable since communication occurs infrequently, i.e., one sensor does not know the local state (sampling rate and consumed energy) of other sensors. In our approach, each sensor maintains an estimate of other sensor's internal state and this estimate is used by the sensor to choose its sampling rate from the stored policy table. As this error in the estimate increases over time, nodes ***periodically*** have to exchange their true local state via communication in order to reset their estimates. We discuss a new *value of information* (the information entropy in the estimate) based communication scheme to reduce communications between sensors and thereby conserve energy. Communication broadcast only occurs when the expected value of information resulting from the communication exceeds a given threshold.

This paper is organized as follows. In the second section we describe related work in energy conservation in sensor networks and MDPs. In section 3, we describe the mathematical models for representing the sensor network as an MDP and the associated stochastic functions. In section 4, we discuss our approach for communication between sensor nodes to execute the coordinated sampling policy. In section 5 we present the detailed simulation results for global policy computation and life time analysis of this sensor system for different operational parameters of the system. In Section 6 we present concluding remarks and plans for future work.

2 Background and Related Work

Distributed sensor networks are used in a number of real world applications which include habitat monitoring, vehicle detection and tracking, perimeter crossing detection, tactical surveillance, health monitoring etc. In general, a set of sensor nodes with different sensing modalities are deployed in such applications to perform sensing, data fusion, and computing results. The sensor nodes are all homogeneous in some situations, for example a sensor web monitoring events, however they often are heterogeneous, for example surveillance with acoustic sensors and infrared cameras, to improve coverage, accuracy, and life

time of such networks. These sensor nodes operate with battery and in many cases in hostile and unattended environments. Therefore, energy efficiency in sensing, communication and processing are important in a sensor network to extend its lifetime.

Distributed microsensor networks use inexpensive sensors (Motes, Mica, etc.) and wireless communications. During the past decade there has been extensive research on developing energy efficient techniques for communication protocols, processing and memory operations, as well as application software [1,5-15]. A combination of techniques can be employed to reduce the total energy budget of a system. Several distributed sensor projects were funded by DARPA programs with military applications, NASA AIST has projects with sensor webs to monitor long range earth phenomena, and we briefly discuss a few projects with real world applications. A. Mainwaring et. al. [16] deployed a 32-node network on Great Duck Island off the coast of Maine to monitor seabird nesting environment and behavior. In [17] distributed sensor webs are used for volcano activity monitoring. These types of applications can use a large number of inexpensive sensors to collect data and monitor changes to signal alarms. Increasing lifetime of sensor network will be important in such systems and they apply several techniques to reduce processing and communication energy costs.

Another approach to increasing lifetime of a sensor network is by carefully managing the energy budget to meet system function requirements. One project with a focus on overall system energy efficiency is about detecting and tracking vehicles in a field using acoustic and geophone sensors [18]. In this application, a group of sensors equipped with acoustic sensors detect a vehicle at a distance by taking acoustic samples, processing data, and collaborating with other sensors to perform beamforming to detect and track vehicles. Energy reduction for this application was important as the requirement is to operate the sensor network for many days with battery power with few events in a day. Techniques used range from using tripwires to wake up sensors when events happen, tradeoff computation and accuracy with energy, and reduce unnecessary communications. In [19], X. Tang et. al. balance the energy consumption of nodes to extend network lifetime based on data precision from different sensor nodes. In [20], M. Cardei et. al. adjust sensing ranges to achieve network lifetime maximization with the condition that each sensor set covers all targets. In [21, 22], the authors place relay or mobile nodes to achieve energy efficiency. Recently, M. Ceriotti et. al. [23] deployed a system in Torre Aquila in Italy for heritage building monitoring. They used dedicated software to collect, disseminate data, as well as time synchronization. The estimated lifetime of the system is beyond one year.

J. Beutel et. al. [24] implemented PermaDAQ in high-mountain permafrost in order to gather real-time environmental data. PermaDAQ system uses the Dozer protocol [25] scheme to reduce communication energy, which is achieved by coordinating MAC-layer, topology control and routing operations. As the active state of the radio consumes a relatively large amount of energy, managing sleep cycles of a collection of nodes is a viable means of energy conservation. Ye et al. [26] employs a periodic cycle to coordinate the sleeping and waking processes. Each sensor node sleeps for a fixed duration and then wakes to listen for transmissions. In most of these applications, researchers formulate the energy efficiency or lifetime as an optimization problem, and use techniques based on linear and non-linear programming to solve them [15, 21, 27-30].

Recently, there is strong interest in using multiple sensor nodes for monitoring human activity as well as patients in critical care. These senor networks are called body area networks and use short range wireless communications among the sensors and nearby base stations. In health monitoring application, energy efficient operation is critical to extend the lifetime to allow for continuous monitoring of patients. CodeBlue is a wireless network designed for use in emergency medical care [31]. Joshi et al. suggested a mobile cardiac outpatient telemetry (MCOT) system [32] which does not require activation by the patient, and is mobile and flexible. MCOT system only monitors the cardiac symptom of patients and provides a service for communicating the information to doctors through cardionet. But these applications are not focus on energy efficiency. U. Varshney [33] presents a patient monitoring solution using ad hoc wireless

networks to allow reliable transmission, where the focus is on system reliability. In [34] critical-path based low energy scheduling algorithms are used for energy efficiency in body area networks. They model patient monitoring with a task graph consisting of various sensing and processing tasks with precedence and solve the scheduling problem to achieve overall system energy efficiency. C. U. Subrahmanya et. al. [35] propose an energy-aware task-aware algorithm for body area networks, based on the existing max-flow min-cut algorithm. The algorithm maps tasks onto a heterogeneous wireless system of sensor nodes such that the tasks meet the deadline and also energy consumption is minimum. The authors have integrated physiological and metabolic sensors into a wireless network and use nonlinear optimization to manage the power efficiency [36].

There are a variety of techniques that can be applied at different aspects of a system to enhance the life time of distributed sensor networks used in healthcare applications. In our work on using multiple sensors to continuously monitor a patient, there are constraints on what processing and communication techniques can be employed. Also, the sensors attached to the body of a patient can operate for 2 to 3 days without replacement. We are interested in the problem of how best to monitor a patient based on health condition for a given period of time. For this purpose, we use sampling rates in sensors based on the health status and other state information to improve energy efficiency, accuracy and the system life time. For our application, it is assumed that other variable such as using fewer sensors or less accuracy is not applicable. A patient is monitored with multiple sensors capable of taking measurements at several different sampling rates. The energy cost is expected to be proportional to the sampling rate and careful coordination among the sensors and their sampling rates can lead to high energy efficiency. This is the focus of this paper and we formulate this as a Markov Decision Process to determine optimal sampling rates of sensors at different time steps and health status.

Due to the non-determinism in the outcomes of the actions and the limited observability of the environment, distributed partially observable Markov Decision Problems (Distributed POMDP) are ideally suited to plan such policies [37-39]. However, the problem of finding the optimal joint policy for general distributed POMDPs is NEXP-complete [40]. Therefore we use completely observable MDPs assuming that all sensors can make perfect observations on the environment while generating the global policy. MDPs have been used elsewhere for control of physical systems [41]. The problem of determining a communication protocol to coordinate the actions of distributed entities using an MDP-based framework is an area of active research in the Agents community. Xuan et al. [42] consider two separate MDPs that use a common global utility function, and model communication as an explicit action that incurs a cost. They describe two heuristic approaches to communication. This approach is applicable only if the transition models are independent. Communication using the framework of Partially Observable MDPs (POMDP) has also been studied [43, 44]. However, in distributed POMDPs, the central planner must reason explicitly about the possible observations of all agents when generating a policy for one agent [45]. This reasoning about observations makes distributed POMDPs much more complex than distributed MDPs. Makoto et al. [46] try to divide one large joint policy tree into many small joint policies through communication in POMDPs. But the number of joint policies grows exponentially to the length of the time horizon. Boutilier [47] proposes a method for solving sequential multi-agent decision problems by allowing agents to reason explicitly about specific coordination mechanisms. The assumption in that work is that agents are observable to each other and hence no communication is needed. These approaches model communication as an explicit or implicit action during the construction of the global policy. This significantly increases the size of the action space. In our work, we do not consider communication during the policy building phase. Communication decisions are made only during policy execution.

3 Model For Coordination

We consider the situation where multiple sensors observe the same phenomenon and make measurements. Consecutive sensor measurements are assumed to be independent and to contain Gaussian random errors. Under this assumption, the error variance in the measurement estimate can be reduced by averaging the individual sensor readings. This corresponds to increasing the sampling rate of the sensors. Increasing the sampling rate also increases the rate of energy consumption (in a physical system, this could be because the sensor operation itself consumes significant amounts of power or that the larger amounts of data resulting from higher sampling rates have to be transmitted using wireless transmission). We assume that the energy reserves at each sensor node are limited. We desire the system to remain in operation for some desired duration, i.e., the sensor nodes should not all run out of power before this time. In addition, we expect the system to respond to changes in the criticality of the event being monitored. Event criticality will be application dependent. For instance, in a human health monitoring network where sensors measure physiological parameters (such as heart rate), anytime the sensor measurements are outside normal ranges, could be classified as a critical event. It is expected that more frequent measurements will be taken when the event criticality is high and less frequent measurements will be sufficient at other times. The problem then is to determine the sampling rates of the individual sensors such that the phenomenon can continue to be measured until the desired lifetime with highest possible sampling rates. If the sampling rates are too high, then the sensors may run out of power before the desired lifetime. On the other hand, if they are too low then the phenomenon is sampled suboptimally.

Our approach is to model the system described above as a Markov Decision Process (MDP). An MDP uses a discrete state space representation of the system being modeled with transitions between system states being defined stochastically (with the Markov assumption). The state transitions are influenced by actions that can be performed in a state. In addition, a reward function ascribes a value to each state that represents the desirability of reaching that state. The decision process is then to determine the optimal sequence of actions that will maximize the reward collected after starting from the initial state.

In our MDP formulation, the system state represents the energy reserves at every sensor and the event criticality. The energy consumption rates and the expected changes in event criticality are modeled as stochastic transitions between system states. These rates are dependent on the sensor sampling rates (the actions). The reward in each state quantifies the utility of sampling at particular rates and the penalty of running out of power. The MDP formulation enables us to compute the optimal policy, which specifies the optimal sampling rate for every sensor at every possible state of the system. This optimal policy is then stored within each sensor node for execution after deployment.

The state space formulation of the system used for calculating the optimal sampling policy assumed that the internal state (current energy reserve) of all the sensors is observable by a sensor at every control-step. But during execution, the world is only partially observable, i.e., one sensor does not know the local status (consumed energy) of the other sensors. In a distributed network, such a *global* policy can be executed only if the nodes exchange state information before every control-step. In a wireless network, such exchanges are energy intensive. We have formulated a method that is less communication intensive for executing the optimal global policy. In our method, every sensor node maintains an estimate of the other sensors' internal states utilizing the same stochastic model used for policy creation. A sensor uses this estimated global state to choose the action from the MDP global policy table.

Over time, the accuracy of these local estimates will decrease. If the communication is cost-free, the agents can communicate their local states at every control step and the optimal global policy can be executed. As communication incurs some energy cost, at every control step, a node computes the information entropy in the local estimate. Only if this entropy is over a threshold, does the node trigger an exchange of the true state information with all other sensors. Thus, communication is initiated only if the expected *value of information* that is to be received is above a pre-defined limit. After a communication

step, the sensor nodes learn the true global state and the entropy of their local estimates reduces to the minimum value. This process is illustrated in Figure 1.

Note that the entire MDP policy is computed offline before node deployment using the stochastic model. Only the resulting policy is stored in the memory of each node. At every control-step, the sensor estimates the current state of the system and then performs a table lookup of the pre-computed policy. Thus, the computational cost of executing the control policy is minimal. (The computational cost of calculating the optimal policy is substantial and increases exponentially with the size of the state space.)

The standard formulation of the algorithms used to solve an MDP involves an explicit enumeration of the complete state space and transition table. The size of the state space is proportional to how finely the real world features (time, energy reserves, event criticality, sampling rates) is discretized. In a multiple node setting, the size of the state space also increases exponentially with the number of sensor nodes. The large state space increases both the time required to compute the optimal policy (this is done offline) and the space required to store the resulting optimal policy within the embedded sensor nodes. Efficient representation of MDPs and computation of approximate policies is an active area of research [48, 49]. For instance, the computation efficiency can be increased when there are conditional independencies in the model [49]. The policy table may also be represented more efficiently than explicitly enumerating every state [50]. However in our current work, we have only implemented the standard formulation of the value iteration algorithm which requires an explicit enumeration of the state space. This restricts the maximum number of sensor nodes that can be modeled (though this is sufficient to model all the different sensors in our health monitoring system).

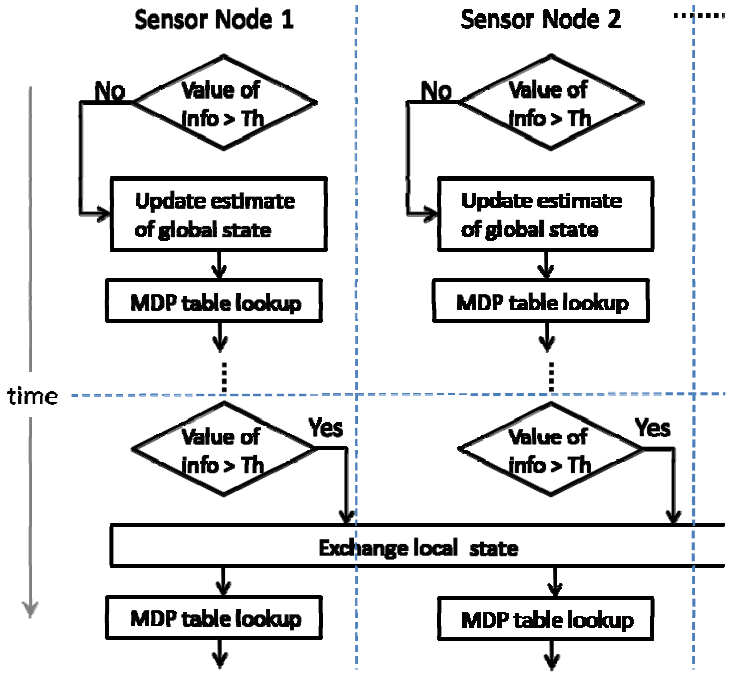

Fig. 1. Interlacing of maintaining local state estimates and learning true global state via communication

3.1 MDP Model for Multiple Sensor Nodes

An MDP is a 4-tuple (S, A, P, R). S is a finite set of states, in one of which the world exists. A is a set of actions that may be executed at any state. P is a probability function that defines how the state changes when an action is executed:$P: S \times A \times S \rightarrow [0,1]$. The probability of moving from state s to state s' after executing action $a \in A$ is denoted $p(s, a, s')$. The probability of moving to a state is dependent only on the current state (the Markov property). R is the reward function:$R: S \times A \rightarrow \boldsymbol{R}$. $R(s, a)$ is the real-valued reward for performing action a when in state s. A *policy* is defined as a function that determines an action for every states $\in S$. The quality of a policy is the expected sum of future rewards. Future rewards are discounted to ensure that the expected sum of rewards converges to a finite value, i.e., a reward obtained t steps in the future is worth $\gamma^t, 0 < \gamma < 1$, compared to receiving it in the current state. γ is called the discount factor. The *value* of a state s under policy π, denoted by$V^{\Pi}(\mathrm{s})$, is the expected sum of rewards obtained by following the policy π from s. The value function determines the action to be executed in state s under π:

$$\underset{a \in A}{\operatorname{argmax}} \left(R(s, a) + \gamma \sum_{s' \in S} p(s, a, s')\, V^{\Pi}(s') \right) \tag{1}$$

A policy is optimal if the value of every state under that policy is maximal. If all the model parameters are known, the optimal policy can be computed by solving the Bellman equations:

$$V(s) = \max_{a \in A} \left(\sum_{s' \in S} p(s, a, s') \left[R(s, a) + \gamma V(s') \right] \right) \tag{2}$$

The Bellman equations are solved using the Value Iteration algorithm. In this algorithm, the value function is initialized to arbitrary values $V_0(s)$. At iteration $k > 0$, the value function is updated:

$$V_k(s) = \max_{a \in A} \left(\sum_{s' \in S} p(s, a, s') \left[R(s, a) + \gamma V_{k-1}(s') \right] \right) \tag{3}$$

As $k \rightarrow \infty$, V_k converges to the optimal policy values.

We now describe how the problem of coordinated sensing in a resource constrained sensor network can be formulated as an MDP. We discretize all real-world features since the MDP formulation we use requires a discrete state space. Let N denote the number of sensors that are to coordinate with each other. The *local state* of node N_i is represented by the state vector (t, h_i, e_i). $t \in \{1,2, \cdots, T\}$ indicates the number of control steps completed since the initial time. T corresponds to the guaranteed lifetime desired from the joint system. $h_i \in \{1,2, \cdots, H\}$ is a measure of the criticality of the sensor readings. This is application dependent. For instance, this could correspond to the health condition of a patient in a human health monitoring application. $e_i \in \{1,2, \cdots, E\}$ is the amount of energy consumed. We assume that the event criticality can be estimated at each sensor node based on its local sensor measurements. is available to each sensor node i.e.,$h_i = h, i = 1,2, \ldots, N$. Representing the state using components each representing an independent entity is called a feature space representation.

The *global state* is the joint local states of all the sensors, $(S_1, S_2, \ldots, S_N)$. Let S denote the finite set of all possible global states. The joint action space A is the action concurrently executed by all sensors, $A = A_1 \times A_2 \times \ldots \times A_N$ where A_i is the action space of sensor node N_i and A denotes the set of all possible sensor sampling rates.

P is the transition probability function defining how the global state changes when a joint action is executed, $P: S \times A \times S \rightarrow [0,1]$. The probability of moving from state s_i to state s_j after taking action a is

denoted by $p(s_i, a, s_j) = p\left((t_i, h_i, \boldsymbol{e_i}), (a_1, a_2, \dots, a_N), (t_j, h_j, \boldsymbol{e_j})\right)$ where $\boldsymbol{e_i} = (e_{1,i}, e_{2,i}, \dots, e_{N,i})$ and $\boldsymbol{e_j} = (e_{1,j}, e_{2,j}, \dots, e_{N,j})$. The increase in control-step, change in event criticality, and fall in energy reserves are independent and hence we can define

$$p(s_i, a, s_j) = p_T(t_i, t_j) p_H(h_i, h_j) \prod_{k=1}^{N} p_E(e_{k,i}, a_k, e_{k,j}) \quad (4)$$

We define the component transition functions below.

$$p_{T(t_i,t_j)} = \begin{cases} 1, \text{if } t_i = t_j = T \\ 1, \text{if } t_j = t_i + 1 \\ 0, \text{otherwise} \end{cases} \quad (5)$$

$$p_E(e_i, a, e_j) = \begin{cases} 1, \text{if } e_i = E \\ p_P(a), \text{if } e_j = e_i \\ 1 - p_P(a), \text{if } e_j = e_i + 1 \\ 0, \text{otherwise} \end{cases} \quad (6)$$

The rate at which energy is consumed by a sensor is dependent on the sampling rate (action) and modeled with probability $p_E(a)$. The energy consumption rate increases with the sampling rate.

$$p_{H(h_i,h_j)} = \begin{cases} p_H, \text{if } i = j \\ 2p_H^{\text{change}}, \text{if } h_i = 1, h_j = 2 \text{ or } h_i = H, h_j = H - 1 \\ p_H^{\text{change}}, \text{if } |h_i - h_j| = 1 \\ 0, \text{otherwise} \end{cases} \quad (7)$$

p_H and p_H^{change} are probabilities that model the change in event criticality (the two values are dependent since the sum of all transition probabilities out of a state must sum to 1). These component transitions are illustrated in Figure 2 and the evolution of the full local state is shown in Figure 3.

$R = R(s, a) = R\left((t, h, e_1, e_2, \dots, e_N), (a_1, a_2, \dots, a_N)\right)$ is the reward function and it depends only on the sensor sampling rates and the event criticality. Intuitively, the sampling rate should be higher during critical events. There is a penalty, R_{powerout}, if the system (i.e., *all* sensor nodes) runs out of power before the desired lifetime. The reward function is defined as

$$R\left((t, h, e_1, e_2, \dots, e_N), (a_1, a_2, \dots, a_N)\right) = \begin{cases} R_{\text{powerout}}, \text{if } t < T, e_i = E, i = 1,2, \dots, N \\ k \times \sum_{i=1, e_i < E}^{N} a_i \times h, \text{otherwise} \end{cases} \quad (8)$$

If multiple sensors are able to sense simultaneously, then the reward is proportional to the sum of the sampling rates, i.e., the reward is inversely proportional to the expected variance in the fused estimate. This is obtained from the assumption that successive sensor measurements are independent and that the true measurement is corrupted by zero-mean Gaussian noise. The above formulation holds true when the sensors have the same error variance and zero co-variance. This term can be modified to reflect unequal error variances and co-variances (for instance, using a Kalman filter formulation).

The global policy is obtained by solving this MDP through the value iteration or policy iteration algorithms.

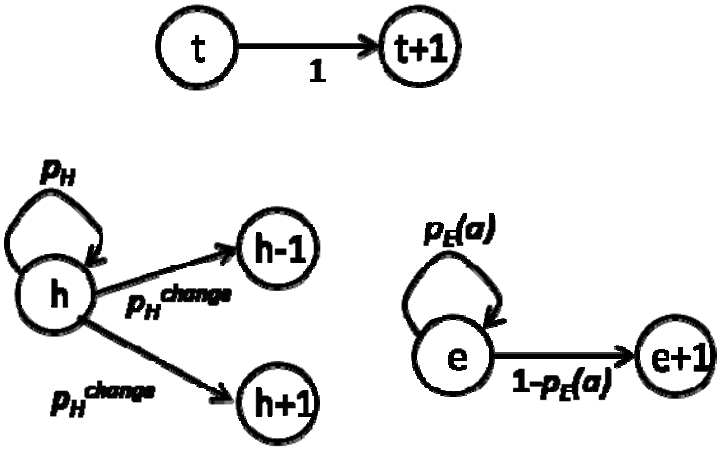

Fig. 2. Component transitions of the MDP model. "t" represents the evolution of time, "h" the evolution of event criticality, and "e" the local (energy) resources at a node.

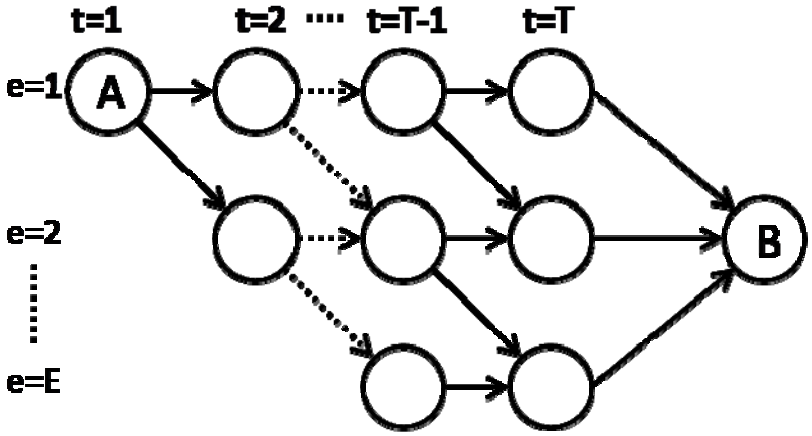

Fig. 3. Evolution of the local state at a sensor node. State labeled "A" is the initial state (t=1 and full energy reserve), and state labeled "B" is the terminal state (t=T). At every control-step, the time element increases by one, and the energy element increases stochastically. The event criticality element is not shown for clarity.

4 Communication

We have formulated a method to decide when to communicate that makes use of the inherent ambiguity in set of possible global states at a control step. If the level of ambiguity is over a pre-defined threshold then the sensor nodes exchange information to identify the true global state (no ambiguity). We use the information theoretic measure of Shannon entropy to quantify this ambiguity. The entropy is least when the exact global state is known and maximum when the global state may be any of the possible states with equal probability. We now express these concepts in terms of the MDP.

Let t denote the control step when the decision to communicate has to be made by node N_i. Denote by subscript $-i$ the set of parameters of all nodes other than N_i (for example, $e_{t,-i}$ denotes the set of energy reserves of all nodes other than N_i at time t). Let the global state at $t-1$be precisely known to be $s_{t-1} = \left(t, h, e_{t-1,i}, e_{t-1,-i}\right)$ and the action executed at $t-1$ be $a_{t-1} = \left(a_{t-1,i}, a_{t-1,-i}\right)$. At control step t, N_i only knows its local energy reserve, $e_{t,i}$ (local state) but not that of the other nodes. (For simplicity of notation, we assume that the event criticality, h, does not change.) Let e_{-i} denote an arbitrary vector with components $(e_1, e_2, \dots, e_{i-1}, e_{i+1}, \dots, e_N)$. The set of possible states at t contains those states which have a non-zero transition from the state at $t-1$:

$$S_t = \left\{\left(t, h, e_{t,i}, e_{-i}\right) | p_E\left(e_{t-1,j}, a_{t-1,j}, e_j\right) > 0, \forall j \neq i\right\} \quad (9)$$

Here, $p_E(a)$ is the transition probability function. The probability of the global state being a particular state $s \in S_t$, with components $s = \left(t, h, e_{t,i}, e_{-i}\right)$ is given by

$$\Pr_t(s) = \prod_{j=1, j\neq i}^{N} p_E\left(e_{t-1,j}, a_{t-1,j}, e_j\right) \quad (10)$$

Note that $\Pr_t(s)$is dependent only on the transition probability and the action taken in the previous step. We desire to calculate the ambiguity in the set of possible states after the current control-step, i.e., at $t+1$. Let S_{t+1}denote this set. S_{t+1} contains all the states that can be reached from a state in S_t by taking the action specified by the policy. Let $\pi(s)$ denote the action (sampling rates) specified by the policy at a state s. Then, S_{t+1} is given by

$$S_{t+1} = \{s | \exists s' \in S_t, p(s', \pi(s'), s) > 0\} \quad (11)$$

The probability of the global state at $t+1$ being a particular state $s \in S_{t+1}$ is given by

$$\Pr_{t+1}(s) = \sum_{s' \in S_t} \Pr_t(s') p(s', \pi(s'), s) \quad (12)$$

Note that the distribution of states at $t+1$ is dependent on the policy, π. We use the information entropy of this distribution as the value of communication, V_t, and is given by

$$V_t = -\sum_{s \in S_{t+1}} \Pr_{t+1}(s) \log\left(\Pr_{t+1}(s)\right) \quad (13)$$

If V_t exceeds a pre-defined threshold (determined empirically in our current work), the sensor node triggers a communication step, which leads to exchange of local state information and exact knowledge of the global state at control step t.

Each communication action incurs a cost, which is also modeled stochastically with a probability parameter p_c. For each communication action, the energy reserve of the sensor node decreases by one with probability p_c and remains the same with probability of $1-p_c$. The average energy consumed at each communication step is p_c.

5 Simulation Results

The MDP models that we described have several parameters such as the various transition probabilities (energy consumption rate), relative amounts of rewards/penalties, and communication cost. We first evaluate the effect of changing an MDP parameter on the computed policy. We next study the parameters that affect the amount of communication between sensors during policy execution. We then evaluate the

sensitivity of the policy to differences in the stochastic model parameters used during policy computation and policy execution. We also compare the performance of the MDP policy with other fixed and random policies.

We use two metrics of system performance. Note that as the policy is stochastic, the metrics are expected values (obtained in simulation experiments by averaging the results from several policy executions). The first is the *system energy outage* percentage which is defined as the proportion of policy executions that ended with *all* sensors running out of power before the desired lifetime (T). The second metric is the *system lifetime* which is defined as the expected number of control steps that the system is in operation (at least one of the sensors has power).

5.1 Effect of MDP parameters on policy

In these experiments, we execute the computed policy under two extremes of communication: full or no exchange of state information. In the full exchange case, the sensors communicate before every control step to learn the full state. In the no communication case, each sensor relies on a stochastic model of the other sensor's performance in lieu of the true state. Figure 4 shows the probability of the system running out of power under these two cases. The parameters used in these simulations vary with the number of sensors to ensure that the policy can be computed in a reasonable amount of time (Table 1). The discount factor used to calculate the optimal MDP policy is $\gamma = 0.99$. The results are averaged from 10,000 simulation runs. As expected, full exchange of state information enables the execution of the optimal policy and hence the system has a longer lifetime.

Table 1. The simulation parameters for the networks with different size

# of sensors	T	H	E	$R_{powerout}$
2	70	10	10	-20000
3	40	10	10	-20000
5	14	4	4	-4000

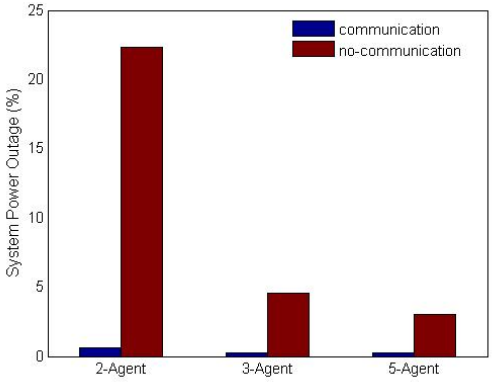

Fig. 4. Probability of system running out of power under the full communication and no communication case

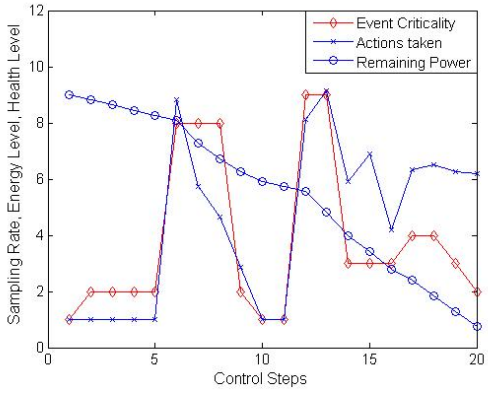

Fig. 5. Change in sampling rate over time with changes in event criticality

We evaluated the policy over 10,000 executions with a time varying event criticality profile to study how the sampling rates (actions of the MDP policy) vary with event criticality. In this simulation, $N = 2, T = 20, H = 10, P = 10, R_{\text{powerout}} = -10000, a \in \{1,2,\ldots,10\}, p_H = 0.75,\ \gamma = 0.99$. Figure 5 shows the change in sampling rate over time with changes in event criticality. The sampling rate increases during periods of high criticality while the policy still ensures that the system does not run out of power.

5.1.1 Effect of penalty for running out of power

We changed the magnitude of the negative penalty that was used in the reward function (R_{powerout}) to study its impact on the policy. Figure 6 shows the system energy outage percentage for a range of penalties in the 2-sensor case. The plot shows the benefit of (full) communication. Figure 7 shows the average utility of the corresponding policies. The utility of full communication is below that of the no communication case when the magnitude of the penalty becomes small. Thus, the magnitude of the penalty should be higher than the crossover point shown in Figure 7.

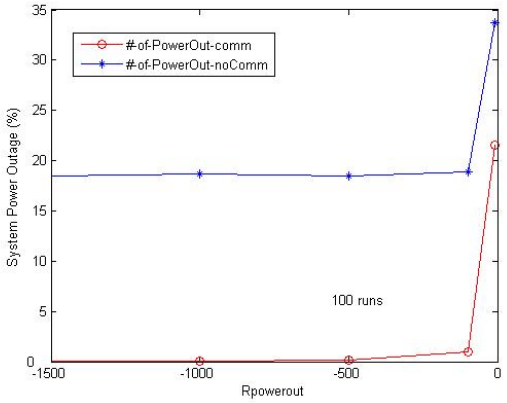

Fig. 6. Effect of penalty in the reward function on the system power outage percentage for both the full and no communication case

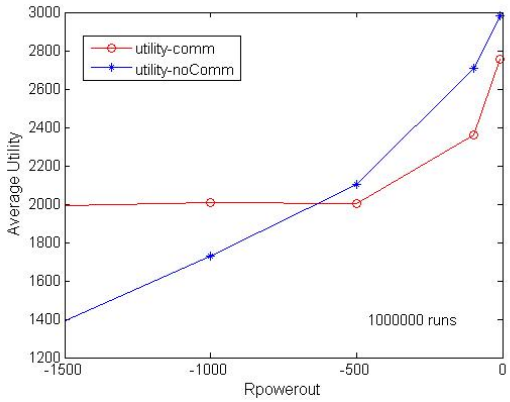

Fig. 7. The utility of the MDP policy with different penalties in the reward function for both the full and no communication case

5.1.2 Effect of probability of change in criticality

We changed the magnitude of the probability that was used in the transition function to model the change in event criticality with time (p_H) to study its impact on the policy. This probability will be dependent on the specific application and the granularity of the model. Figure 8 and Figure 9 shows the system energy outage percentage and lifetime respectively for a range of probabilities. Each plot shows the benefit of (full) communication. These results show that while the policy responds appropriately to changes in event criticality, it is relatively insensitive to the specific values used to model the change in event criticality.

5.1.3 Effect of energy consumption rate on lifetime

In the MDP formulation, the rate at which energy is consumed at a sensor when sampling at a particular rate is modeled stochastically. We derived analytical equations to relate the energy consumption rate at a particular sampling rate to the expected system lifetime. In these derivations, we only consider the case where the sensors sample at a fixed rate. The goal is to understand the relationship between the stochastic energy consumption rate and the expected system lifetime in order to help us decide the appropriate discretization to be used while formulating the MDP model (i.e., to decide how finely time and energy have to be subdivided in the model).

Let p denote the probability that the energy level increases by one at a particular sampling rate. Let P denote the number of discrete energy levels and T the number of discrete time levels. We now derive the expected lifetime for such a system. The lifetime is the number of time-steps that at least one of the sensors has not run out of energy. Note that the system lifetime could be an arbitrarily large number with some small but non-zero probability. As the time model in the MDP is finite, we set the lifetime to T for all cases when the system survives beyond the maximum time level, T, in the model. This enables us to compare the analytical results with finite simulations.

Let $\Pr_{\text{fail}}(t,p)$ denote the probability that the lifetime of a single sensor is less than t. p is the probability that the energy level of the sensor increases in one time-step. Then,

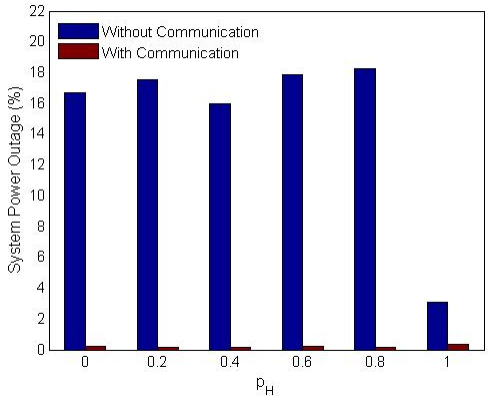

Fig. 8. Effect of probability of change in event criticality on the number of times the system has a complete power outage for both the full and no communication case

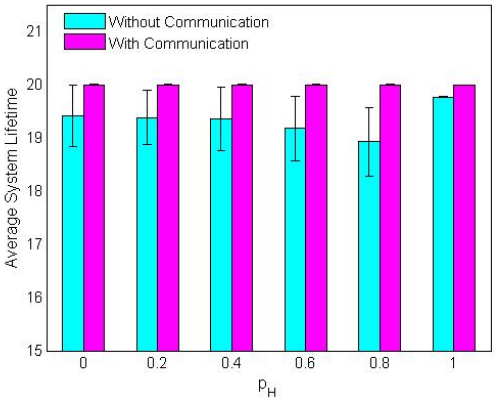

Fig. 9. Effect of probability of change in event criticality on the system lifetime for both the full and no communication case

$$\Pr_{\text{fail}}(t,p) = \sum_{j=P-1}^{t-2} \binom{j-1}{P-2} p^{P-1} (1-p)^{j-P+1} \quad \textbf{(14)}$$

Define $\Pr_{\text{fail}}(t,p) = 0$if P-1>t-2. This is the probability that the energy level increased at least $(P-1)$ times in $(t-1)$ time-steps. The probability of the lifetime being exactly $(t-1)$ is

$$\Pr_{\text{fail,t}}(t,p) = \Pr_{\text{fail}}(t,p) - \Pr_{\text{fail}}(t-1,p) \quad \textbf{(15)}$$

The expected lifetime of the single sensors is given by

$$\sum_{t=P+1}^{T}(t-1)\times \Pr_{\text{fail,t}}(t,p) \; + T\times\left(1-\sum_{t=P+1}^{T}\Pr_{\text{fail,t}}(t,p)\right) \tag{16}$$

The second term denotes the case where the sensor survives for more than T time steps. The probability of the system (two sensors) lifetime is $(t-1)$ is given by

$$\Pr_{\text{fail,system}}(t,p) = 2\times\Pr_{\text{fail,t}}(t,p)\times\Pr_{\text{fail}}(t,p) - \left(\Pr_{\text{fail,t}}(t,p)\right)^2 \tag{17}$$

The expected system lifetime is then given by

$$\sum_{t=P+1}^{T}(t-1)\times \Pr_{\text{fail,system}}(t,p) \; + T\times\left(1-\sum_{t=P+1}^{T}\Pr_{\text{fail,system}}(t,p)\right) \tag{18}$$

We plot the expected system lifetime with different values for energy consumption rate and maximum available energy in Figure 10. The maximum number of time-steps is 20. The plot enables us to set the range of probabilities that should be used to model the energy consumption rate at different sampling rates, and the relative values of maximum energy and time in the MDP model.

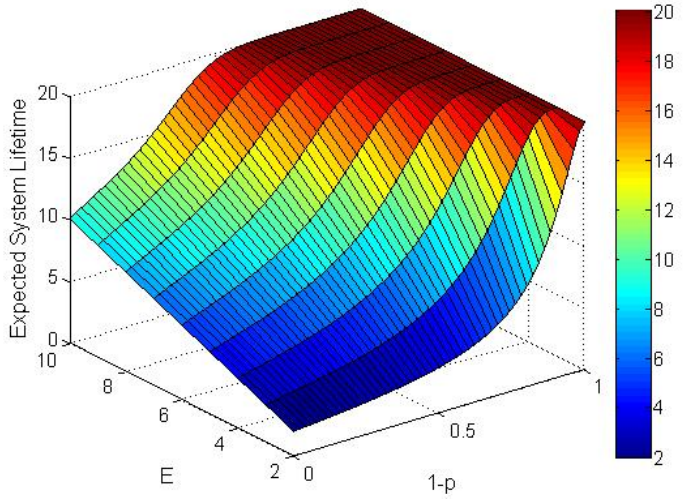

Fig. 10. Expected system lifetime with different discretization of available energy (P) and energy consumption rate ($\boldsymbol{p_E = 1-p}$)

5.2 Communication

Figure 11 shows the change in the probability of system power outage with variation of the communication threshold for $N = 5$ cases. Figure 12 shows the corresponding average utilities. The communication threshold represents the minimum uncertainty in the global state estimate required to trigger an exchange of local state information between nodes via communication. p_c represents the cost (probability that energy reserves keeps the same) of a communication step. Higher values of p_c corresponds to lower communication cost. These figures show the trade-off between the energy cost of communication and the

amount of information available from communication. Increased communication leads to better execution of the optimal policy but it also incurs a higher cost from communication. When the threshold is low, the number of communication steps increases and this leads to execution of the optimal policy and low system outage. On the other hand, when the communication threshold is high, communication reduces and system performance begins to reflect the no communication case.

5.3 Sensitivity to errors in model parameters

In the earlier simulations, it was assumed that the stochastic model used during offline policy computation exactly matched the stochastic energy consumption model of the physical system during execution. Here, we generate the optimal MDP policy by underestimating and overestimating the true rate of energy expenditure of the sensor system (and which is used during execution). The amount of over- or underestimation is the difference in probability of energy increase (p_E) between the model used for computing the MDP policy and the true model used during execution.

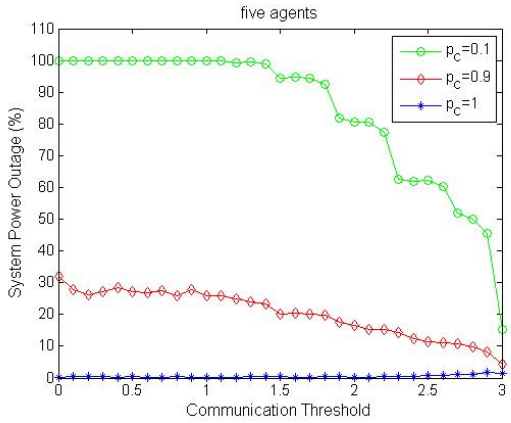

Fig. 11. Effect of varying the threshold of communication on the system power outage. N=5

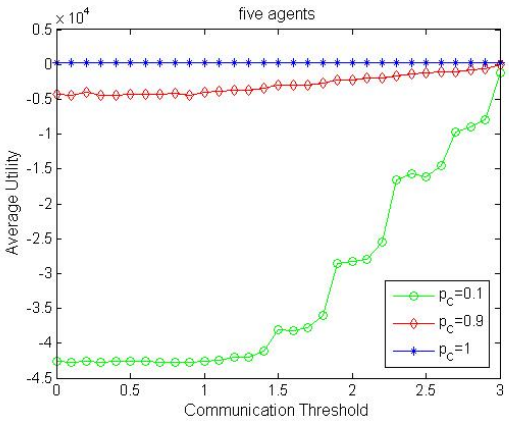

Fig. 12. Effect of varying the threshold of communication on the average utility of executing the MDP policy. N=5

Figure 13 shows the effect that the model error has on the expected performance of the resulting policy and Figure 14 shows the decrease in energy reserves over time with varying amounts of model error ($N = 2$) when the agents do not communicate at all. (In these figures, an error of x% indicates that the probability of energy increase used for simulation is $1 + x$ times the probability used during policy calculation). In these cases, the resulting policy is impacted by model errors. Figure 15 shows the corresponding decrease in energy reserves when the agents communicate during policy execution ($p_c = 0.9$, threshold of communication = 1.75). These plots show that communication during execution offsets the inherent error in the model and removes the variation in the expected performance of the resulting policies. Figure 16 and Figure 17 show the probability of the system running out of power and the expected lifetime when the model over or underestimates the energy expenditure rate. The ability to communicate local state information (local energy reserve) increases the system lifetime.

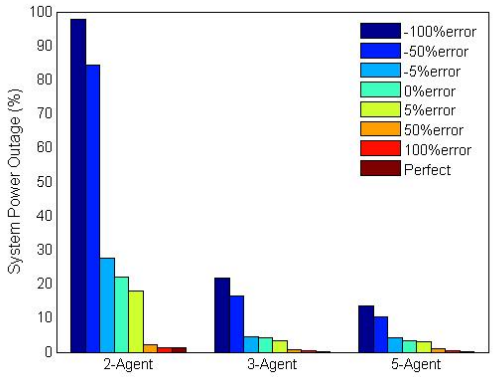

Fig. 13. Probability of system power outage with varying amounts of model error ($\boldsymbol{N = 2, 3, 5}$)

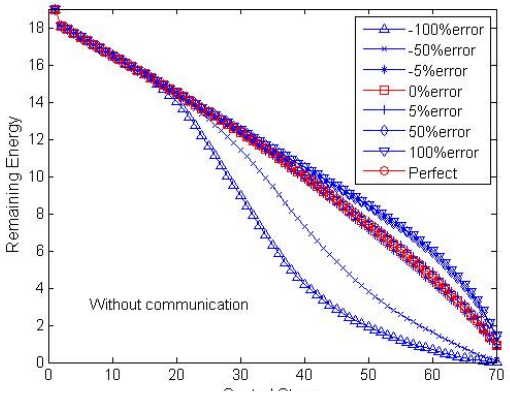

Fig. 14. Remaining energy over time with varying amounts of model error and no communication ($\boldsymbol{N = 2}$)

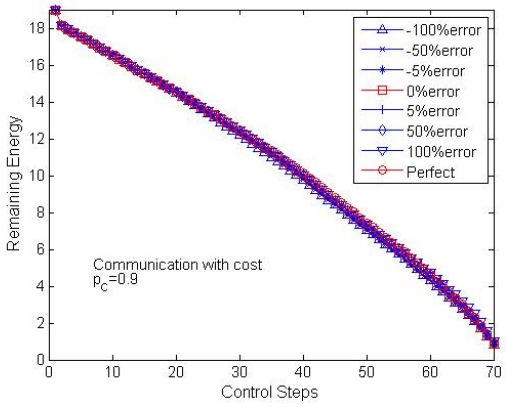

Fig. 15. Remaining energy over time with varying amounts of model error and communication during policy execution ($\boldsymbol{N = 2, p_c = 0.9}$, threshold of communication = 1.75)

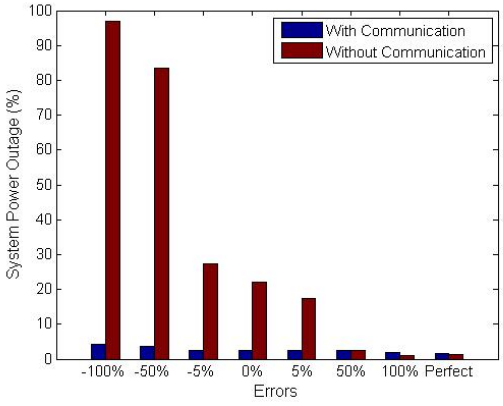

Fig. 16. Probability of system power outage with varying amounts of model error ($\boldsymbol{N = 2}$)

5.4 Comparison with Random and Fixed Policies

We now compare the performance of the sensor sampling policies as computed by the MDP model with other fixed and random sampling policies. The fixed policies are:

a. "Min": always sample at the lowest sampling rates.

b. "Max": always sample at the highest sampling rates.

c. "Heuristic": sample at a rate determined by remaining energy (R_E), remaining time steps (R_T), and energy consumption model (p(a)) as the following equation: $R_T = \frac{R_E}{1-p(a)}$.

The random policy ("Random") is to sample at a random rate at every time step. The MDP policies are executed with communication with no energy cost ("MDP-FC") and with a non-zero cost of communication ("MDP-CC"). The corresponding probability of system power outage is shown in Figure 18. Figure 19 and Figure 20 show the corresponding average utility and average system lifetime. Note that the MDP-FC policy results in a power outage probability as low as the minimum sampling rate case (the best possible) but while enabling the sensors to sample at higher sampling rates towards the end of the desired (the highest utility). The MDP policy is thus able to use the current time and the desired lifetime to increase the sensor sampling rates when it becomes likely that the system will not run of power before the desired duration.

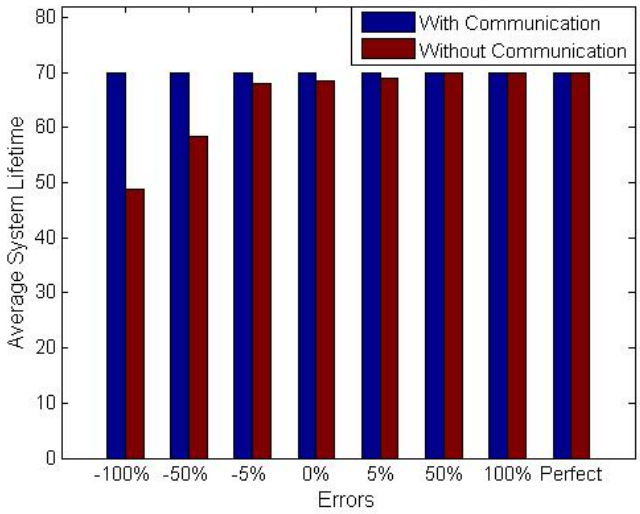

Fig. 17. Average system lifetime with varying amounts of model error ($\boldsymbol{N = 2}$)

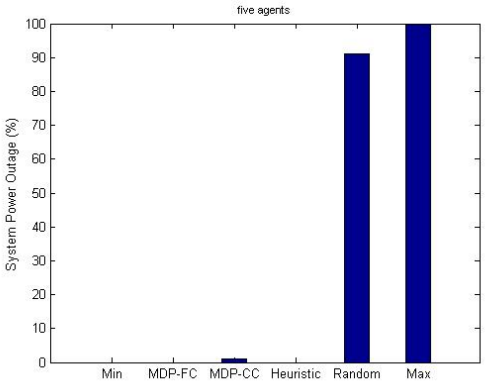

Fig. 18. System power outage of MDP, Random, and Fixed policies ($\boldsymbol{N = 5}$)

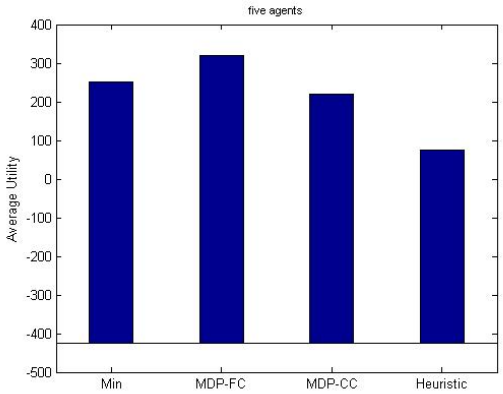

Fig. 19. Average Utility of MDP, Random, and Fixed policies ($\boldsymbol{N = 5}$)

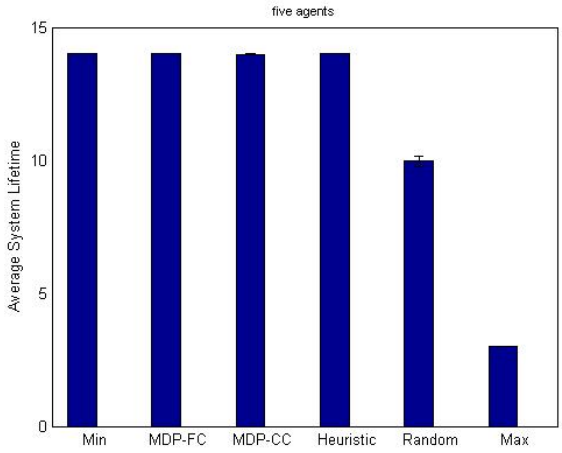

Figure 20: Average System Lifetime of MDP, Random, and Fixed policies ($N = 5$)

6 Conclusions and Future Work

We have shown how the Markov Decision Process framework can be used as the basis for coordinated sampling in a sensor network. Our approach enables an optimal policy to be computed before deployment under the assumptions of full observability of the local state of all sensors. We have formulated communication scheme that enables individual sensors to execute this global policy by communication with other sensors only when the expected value of information to be gained from the communication is high. We have shown simulation results that characterize the performance of this control framework. This method is suitable for networks of relatively few sensors and where the computational capabilities and energy reserves at each node are limited.

In future work, we plan to modify the policy computation algorithms and policy representation so that the technique can be scaled to larger number of node. One of the reasons for the large state space is because of the discretization of continuous parameters. We will also explore variants of MDPs that use continuous Markov models for this reason.

Acknowledge

This work is supported by NSF grant no. 0615132 from the Division of Computer and Network Systems.

References

[1]. C. S. Raghavendra, K. M. Sivalingam, and T. Znati, "Wireless Sensor Networks," Springer, 2006, pp. 3-107.

[2]. M. Venugopal, K. E. Feuvrel, D. Mongin, S. Bambot, M. Faupel, A. Panangadan, A. Talukder, and R. Pidva, "Clinical Evaluation of a Novel Interstitial Fluid Sensor System for Remote Continuous Alcohol Monitoring," *IEEE Sensors Journal,* vol. 8, pp. 71-80, 2008.

[3]. A. Talukder, "Motes for mobile health monitoring and tele-medicine," in *Crossbow Solutions*. vol. 6, 2005, pp. 1-4.

[4]. A. Panangadan, S. M. Ali, and A. Talukder, "Markov Decision Processes for Control of a Sensor Network-based Health Monitoring System," in *Proceedings of the Seventeenth Innovative*

Applications of Artificial Intelligence Conference Pittsburgh: AAAI Press, Menlo Park, California, 2005, pp. 1529-1534.

[5]. L. Benini and G. DeMicheli, "Dynamic Power Management: Design Techniques & CAD Tools," Norwell, MA: Kluwer Academic Publishers, 1997.

[6]. A. P. Chandrakasan and R. W. Brodersen, "Low Power CMOS Digital Design," Norwell, MA: Kluwer Academic Publishers, 1996.

[7]. P. Lettieri, C. Schurgers, M. B. Srivastava, "Adaptive Link Layer Strategies for Energy Efficient Wireless Networking," Wireless Network, October 1999. (TR-UCLA-NESL-199910-01).

[8]. T. A. Pering, T. D. Burd, and R. W. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms," in *Proceedings of ISLPED*, 1998, pp. 76-81.

[9]. J. Rabaey, M. J. Ammer, J. L. d. Silva, D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra low power wireless networking," *IEEE Computer Magazine,* vol. 33, pp. 42 - 48, 2000-07 2000.

[10]. V. Raghunathan, S. Ganeriwal and M. B. Srivastava, "Energy efficient wireless packet scheduling and fair queuing," ACM Transactions on Embedded Computing Systems, February 2004. (TR-UCLA-NESL-200402-02).

[11]. C. Schurgers, O. Aberthorne, and M. Srivastava, "Modulation scaling for energy aware communication systems," in *Proceedings of ISLPED*, 2001.

[12]. A. Sinha, A. Wang, and A. P. Chandrakasan, "Algorithmic transforms for efficient enerrgy scalable computation," in *Proceedings of ISLPED*, 2000.

[13]. A. Wang, S.-H. Cho, C. G. Sodini, and A. P. Chandrakasan, "Energy-efficient modulation and MAC for asymmetric microsensor systems," in *Proceedings of ISLPED*, 2001.

[14]. F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *Proceedings of Annual Symp. on Foundations of Computer Science*, 1995, pp. 374-382.

[15]. S.C. Ergen and P. Varaiya, "Energy Efficient Routing with Delay Guarantee for Sensor Networks", ACM Wireless Networks Journal, vol.13, no. 5, pp. 679-690, October 2007.

[16]. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," ACM Wireless Sensor Networks and Applications (WSNA), Spetmeber 28, 2002, Atlanta, Georgia, USA.

[17]. S. Kedar, S. Chien, F. Webb, D. Tran, J. Doubleday, A. Davis, D. Pieri, W. Song and B. Shirazi, "Optimized Autonomous Space In-situ Sensor-Web for Volcano Monitoring," IEEE Aerospace 2008.

[18]. Caimu Tang, C. S. Raghavendra, "Energy Efficient Detection Algorithms for Wireless Sensor Networks," Book chapter in *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, edited by Jie Wu, CRC Press, June 2005.

[19]. X. Tang and J. Xu, "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," In *Proceedings of IEEE INFOCOM'2006*, April 2006, pp. 755-766.

[20]. M. Cardei, J. Wu, M. Lu and M. O. Pervaiz, "Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges," in Proceedings of Wireless and Mobile Computing, Networking and Communications (WiMob'2005), 22 August 2005.

[21]. S. C. Ergen and P. Varaiya, "Optimal placement of relay nodes for energy efficiency in sensor networks," in Proceedings of International Conference on Communications (ICC2006), Istanbul, June 2006.

[22]. A. Somasundara, A. Kansal, D. Jea, D. Estrin, M. B Srivastava, "Controllably Mobile Infrastructure for Low Energy Embedded Networks," IEEE Transactions on Mobile Computing (TMC) , August 2006. (TR-UCLA-NESL-200503-06).

[23]. M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta and P. Zanon, "Monitoring Heritage Buildings with Wireless Sensor Networks: The Torre Aquila Deployment," In Proceeding of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) Poster Session, San Francisco, CA, US, 2009.

[24]. J. Beutel, S. Gruber, A. Hasler, R. Lim, A. Meier, C. Plessl, L. Talzi, L. Thiele, C. Tschudin, M. Woehrle and M. Yuecel, "PermaDAQ: A Scientific Instrument for Precision Sensing and Data Recovery in Evnironment Extremes," In Proceeding of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) Poster Session, San Francisco, CA, US, 2009.

[25]. N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *Proceedings of 6th Int'l Conference on Information Processing Sensor Networks (IPSN '07)*, pages 450–459. ACM Press, New York, April 2007.

[26]. W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2002.

[27]. A. Talukder, R. Bhatt, T. Sheikh, R. Pidva, L. Chandramouli and S. Monacos, "Dynamic control and power management algorithm for continuous wireless monitoring in sensor networks." In *Proceedings of the 29th Conference on Local Computer Networks, EmNetS*, 498–505.

[28]. A. Krause and R. Rajagopal, "Simultaneous Placement and Scheduling of Sensors." In Proceeding of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) Poster Session, San Francisco, CA, US, 2009.

[29]. P. Wan and M. D. Lemmon, "Event-triggered Distributed Optimization in Sensor Networks." In Proceeding of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) Poster Session, San Francisco, CA, US, 2009.

[30]. M. R. Benjamin, "Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control," Ph.D. dissertation, Broun University, Providence, RI, May 2002.

[31]. D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care," in *MobiSys Workshop on Applications of Mobile Embedded Systems (WAMES)*, 2004.

[32]. A. K. Joshi, P. R. Kowey, E. N. Prystowsky, D. G. Benditt, D. S. Cannom, C. M. Pratt, A. McNamara, and R. M. Sangrigoli, "First experience with a Mobile Cardiac Outpatient Telemetry (MCOT) system for the diagnosis and management of cardiac arrhythmia," *Am J Cardiol* vol. 95, pp. 878-881, 2005.

[33]. U. Varshney, "Using Wireless Networks for Enhanced Monitoring of Patients," in Proceedings of the 10th Americas Conference on Information Systems, New York, August 2004.

[34]. Y. Liu, B. Veeravalli, S. Viswanathan, "Critical-Path based Low-Energy Scheduling Algorithms for Body Area Network Systems," in Proceeding of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 07), 21-24 Aug. 2007.

[35]. C. U. Subrahmanya, B. Veeravalli, Y. Liu, S. Viswanathan, "On the Design of Static and Dynamic Energy-Aware Task Mapping Algorithms for Body Area Networks," In proceeding of 5th International Workshop on Wearable and Implantable Body Sensor Networks (BSN), June 1-3, 2008, Hong Kong, China.

[36]. A. Talukder, A. Panangadan, L. Chandramouli, and S. Ali, "Optimal sensor scheduling and power management in sensor networks," in *Invited Talk at the SPIE Defense and Security Symposium, Optical Pattern Recognition XVI*, Orlando, FL, 2005.

[37]. R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, "Approximate solutions for partially observable stochastic games with common payoffs," in *Proceedings of AAMAS*, 2004.

[38]. R. Nair, P. Varakantham, M. Tambe, and S. Marsella, "Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings," in *Proceedings of IJCAI*, 2003.

[39]. D. Szer, F. Charpillet, and S. Zilberstein, "MMA*: A heuristic search algorithm for solving decentralized POMDPs," in *Proceedings of IJCAI*, 2005.

[40]. D. S. Bernstein, S. Zilberstein, and N. Immerman, "The complexity of decentralized control of MDPs," in *Proceedings of UAI*, 2000.

[41]. S. Zilberstein, R. Washington, D. S. Bernstein, and A. I. Mouaddib, "Decision-theoretic control of planetary rovers," *Plan-Based Control of Robotic Agents, LNAI,* vol. 2466, pp. 270-289, 2002.

[42]. P. Xuan, V. Lesser, and S. Zilberstein, "Communication Decisions in Multi-agent Cooperation: Model and Experiments," in *Fifth International Conference on Autonomous Agents*, Montreal, Canada, 2001.

[43]. S. Williamson, E. Gerding, and N. Jennings, "A principled information valuation for communications during multi-agent coordination," in *Proc AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*, Estoril, Portugal, 2008.

[44]. C. V. Goldman and S. Zilberstein, "Optimizing Information Exchange in Cooperative Multi-agent Systems," in *Second international joint conference on Autonomous agents and multiagent systems*, Melbourne, Australia, 2003, pp. 137-144.

[45]. R. Nair, M. Roth, and M. Yohoo, "Communication for Improving Policy Computation in Distributed POMDPs," in *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, New York, New York, 2004, pp. 1098-1105.

[46]. M. Tasaki, Y. Yabu, Y. Iwanari, M. Yokoo, M. Tambe, J. Marecki, and P. Varakantham, "Introducing communication in Dis-POMDPs with locality of interaction," in *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Sydney, Australia, 2008.

[47]. C. Boutilier, "Sequential optimality and coordination in multiagent systems," in *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence(IJCAI)*, San Francisco, CA, USA, 1999, pp. 478-485.

[48]. L. Li and M. L. Littman, "Lazy approximation for solving continuous finite-horizon MDPs " in *Twentieth National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA, 2005, pp. 1175-1180.

[49]. C. Boutilier, R. Dearden, and M. Goldszmidt, "Exploiting structure in policy construction," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, 1995, pp. 1104-1111.

[50]. P. Liberatore, "On Polynomial Sized MDP Succinct Policies," *Journal of Artificial Intelligence Research,* vol. 21, pp. 551-577, 2004.