

# Evaluation of a Markov Decision Process-based Coordinated Sampling Method

Shuping Liu

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

Anand Panangadan

Jet Propulsion Laboratory  
4800 Oak Grove Drive  
Pasadena, CA 91109, USA  
1-818-354-3054  
Anand.V.Panangadan@jpl.nasa.gov

## ABSTRACT

The paper evaluates the use of Markov Decision Processes (MDP) as a framework for coordinated sensing and adaptive communication between distributed sensors. The technique enables distributed sensors 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 policy that specifies the sampling rates for each node for optimal coordinated sensing. We simulated this control mechanism to study the effect that the various model parameters have on the generated control policy, and compare the performance of the proposed controller with other policies.

## Categories and Subject Descriptors

C.2.1 [Computer Communication Networks]: Network Architecture and Design – *wireless communication*. G.3 [Probability and Statistics]: Markov Processes. H.1.1 [Information Systems]: Systems and Information Theory – *value of information*.

## General Terms

Algorithms, Management, Performance, Design, Reliability, Theory

## 1. INTRODUCTION

In many earth and space science applications, it is required that environmental sensors operate autonomously for extended periods of time. However, limited battery capacity and the high energy cost of

wireless transmission of the sensed data [1], require that energy be actively conserved. An important performance metric in such applications is the life time of the collection of sensors. Adaptive sensing is one technique of extending system lifetime. Here, the sampling rates are increased only when critical events are detected. At other times, a sensor has a low sampling rate. Coordinated sensing with multiple sensors can extend system lifetime as a sensor can adapt to the energy reserves of the rest of the system.

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. 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 sensors 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. 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.

In prior work, we have presented a Markov Decision Process (MDP) based method to determine a coordinated sampling policy for the sensor network such that the system lifetime is extended without compromising detection and monitoring of critical events [2]. In our MDP formulation, the system state represents the energy reserves at every sensor and the detected criticality. The energy consumption rates and the expected changes in event criticality are modeled as stochastic transitions

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ESSA Workshop '09, April 16, San Francisco, California, USA  
Copyright © 2009 ACM 978-1-60558-533-8/09/04... \$5.00

between system states. These rates are dependent on the sensor sampling rates. A 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. This policy is then stored within each sensor for execution after deployment.

In this paper, we present detailed simulation results for this policy computation method and an analysis of the effect of different parameters in our technique on system lifetime. We also show simulation results to compare this technique with simpler adaptive sensing schemes.

## 2. RELATED WORK

Coordination using MDPs is an area of active research in the Agents community. Xuan et al. [3] consider the coordination of two separate MDPs that use a common global utility function, and model communication as an explicit action that incurs a cost. They then describe two heuristic approaches to communication. MDPs have been proposed for coordination of planetary rovers [4]. However, in [4] the authors did not calculate the optimal policy by solving MDP and robots need to learn or adapt their behavior on line. Becker R. et. al. [5] consider the problem of coordinating rovers when communication between them is not possible after deployment. Their approach is applicable only if the transition models are independent.

Boutilier [6] 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. The need to communicate due to limited observability of the environment makes Distributed Partially Observable Markov Decision Problems (Distributed POMDP) ideally suited to plan coordination policies [7-9]. However, the problem of finding the optimal joint policy for general distributed POMDPs is NEXP-complete [10]. Therefore we use completely observable MDPs assuming that all sensors can make perfect observations on the environment while generating the global policy.

## 3. MODEL FOR COORDINATION

We first describe the general formulation of a Markov Decision Process followed by our method of formulating the coordinated sensing problem as an MDP.

### 3.1 Markov Decision Processes

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:  $R: S \times A \rightarrow \mathbf{R}$ .  $R(s, a)$  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.  $\gamma$  is called the discount factor. The *value* of a state  $s$  under policy  $\pi$ , denoted by  $V^\pi(s)$ , is the expected sum of rewards obtained by following the policy  $\pi$  from  $s$ . The action to be executed in state  $s$  under  $\pi$  is that action which has the highest value from  $s$ . The optimal value function can be calculated using the Value Iteration algorithm [11].

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 is discretized. In a multiple sensor setting, the size of the state space also increases exponentially with the number of sensors. The large state space increases both the time required to compute the optimal policy and the space required to store the resulting optimal policy within the embedded sensors. Efficient representation of MDPs and computation of approximate policies is an active area of research [12, 13]. For instance, the computation efficiency can be increased when there are conditional independencies in the model [13]. The policy table may also be represented more efficiently than explicitly enumerating every state [14]. 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 number of sensors that can be modeled.

### 3.2 MDP Model for Multiple Sensors

We now describe how the problem of coordinated sensing with resource constraints 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  $(t, h, e_i)$ .  $t \in \{1, 2, \dots, T\}$  indicates the number of control steps completed since the initial time.  $T$  corresponds to the guaranteed lifetime desired from the joint system.  $h \in \{1, 2, \dots, H\}$  is a measure of the criticality of the sensor readings.  $e_i \in \{1, 2, \dots, E\}$  is the amount of energy consumed. 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,  $(S_1, S_2, \dots, S_N)$ . Let  $S$  denote the finite set of all possible

global states. The joint action space  $A$  is the action concurrently executed by all sensors,  $A = A_1 \times A_2 \times \dots \times A_N$  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  $p(s_i, a, s_j) = p((t_i, h_i, e_i), (a_1, a_2, \dots, a_N), (t_j, h_j, e_j))$

where  $e_i = (e_{1,i}, e_{2,i}, \dots, e_{N,i})$  and  $e_j = (e_{1,j}, e_{2,j}, \dots, e_{N,j})$ . The increase in control-step, change in event criticality, and fall in energy reserves are independent and hence we can define

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

We define the component transition functions below.

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

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

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

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

$p_H$  and  $p_H^{\text{change}}$  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 1 and the evolution of the full local state is shown in Figure 2.

$R = R(s, a) = R((t, h, e_1, e_2, \dots, e_N), (a_1, a_2, \dots, a_N))$  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,  $R_{\text{powerout}}$ , if the sensing system (i.e., all sensors) runs out of power before the desired lifetime. The reward function is defined as

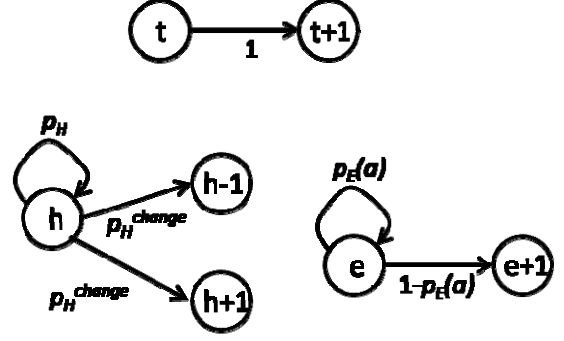


Figure 1: Component transitions of the MDP model. “t” represents the evolution of time, “h” the evolution of event criticality, and “e” the local (energy) resources at a node.

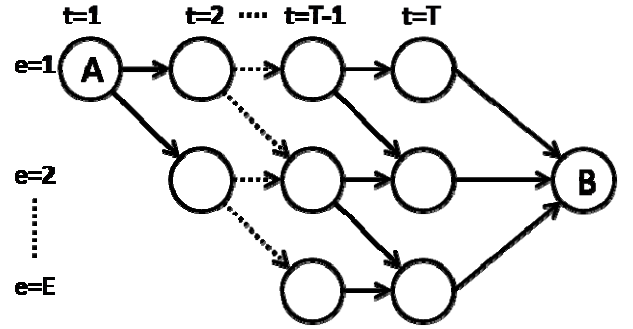


Figure 2: Evolution of the local state at a sensor node. State labeled “A” is the initial state (t=1 and full energy reserve), and state labeled “B” is the terminal state (t=T). At every control-step, the time element increases by one, and the energy element increases stochastically. The event criticality element is not shown for clarity.

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

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.

The state space formulation of the system used for calculating the optimal sampling policy assumes 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 is executed by the sensors exchanging state information before every control-step.

#### 4. 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. 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). Recall that we assign a large negative reward to the MDP when all sensors run out of power before the pre-defined lifetime. Thus, these metrics quantify how closely the MDP policy follows the desired behavior (all sensors do not run out of power) during actual executions. Note that as the policy is stochastic, the metrics are expected values (obtained in simulation experiments by averaging the results from several policy executions).

##### 4.1 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 3 shows the probability of the system running out of power under these two cases. The number of sensors in the system is 2, 3, and 5 and all the sensors have the same error covariance. We first quantify the effect of the full and no communication schemes on the execution of the MDP policy. 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. For the  $N = 2$  case,  $T = 70$ ,  $H = 10$ ,  $P = 10$ ,  $R_{\text{powerout}} = -10000$ . For the  $N = 3$  case,  $T = 40$ ,  $H = 10$ ,  $P = 10$ ,  $R_{\text{powerout}} = -20000$ . For the  $N = 5$  case,  $T =$

$14$ ,  $H = 4$ ,  $P = 4$ ,  $R_{\text{powerout}} = -4000$ . 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.

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,  $N = 2$ ,  $T = 20$ ,  $H = 10$ ,  $P = 10$ ,  $R_{\text{powerout}} = -10000$ ,  $a \in \{1, 2, \dots, 10\}$ ,  $p_H = 0.75$ ,  $\gamma = 0.99$ . Figure 4 shows the change in sampling rate over time with changes in event criticality. The sampling rate increases during periods of high criticality (as would be desired intuitively) while the policy still ensures that the system does not run out of power.

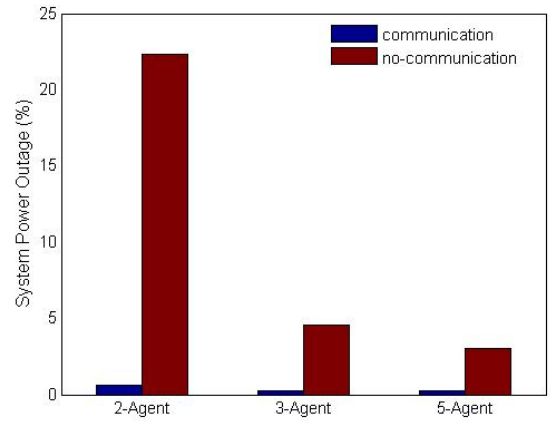


Figure 3: Probability of system running out of power under the full communication and no communication case.

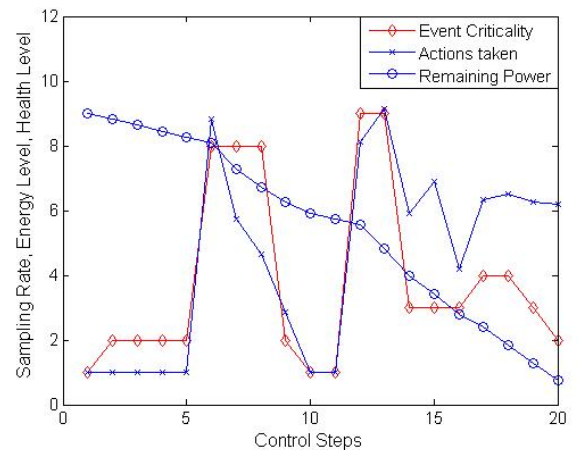
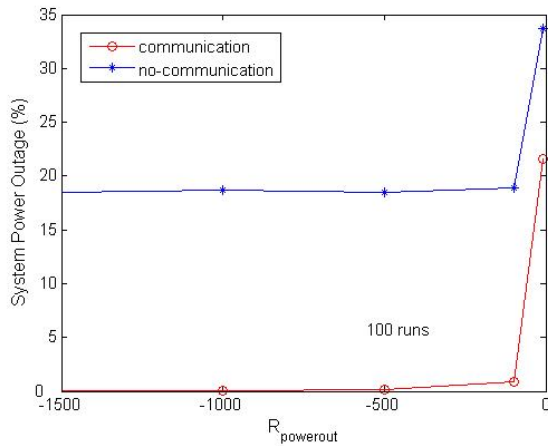


Figure 4: Change in sampling rate over time with changes in event criticality.

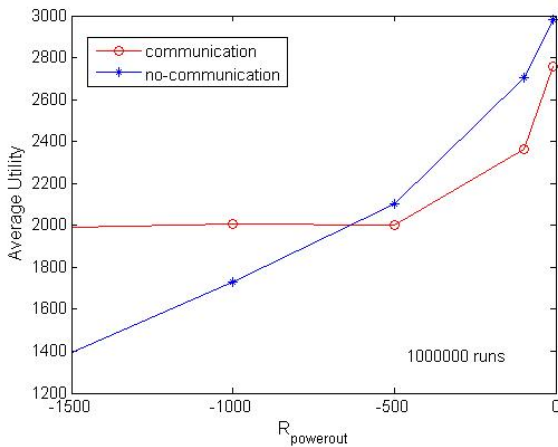
##### 4.1.1 Effect of penalty for running out of power

We changed the magnitude of the negative penalty that was used in the reward function ( $R_{\text{powerout}}$ ) to study its

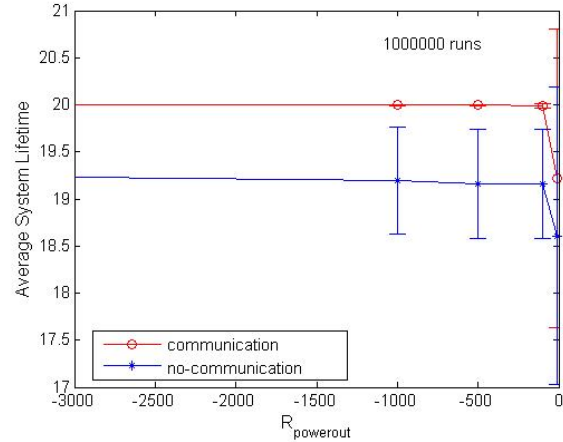
impact on the policy. Figure 5 shows the system energy outage percentage for a range of penalties in the 2-sensor case. The plot shows the benefit of (full) communication. Figure 6 shows the average utility of the corresponding policies. The utility of full communication is below that of the no communication case when the magnitude of the penalty becomes small. This simulation is thus useful to determine an appropriate penalty as the magnitude of the penalty should be higher than the crossover point shown in Figure 6. The average system lifetime is shown in Figure 7. As expected, the lifetime in case of full communication is higher than that of no-communication.



**Figure 5:** Effect of penalty in the reward function on the system power outage percentage for both the full and no communication case.



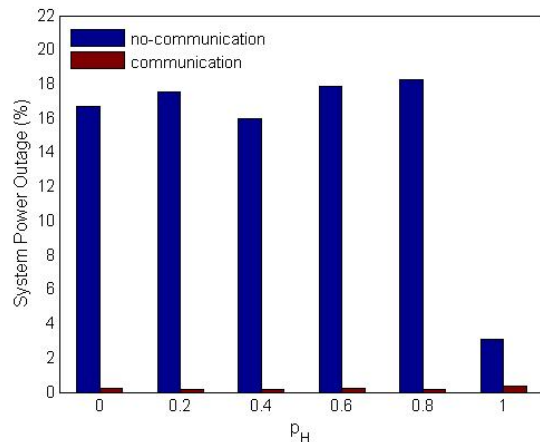
**Figure 6:** The utility of the MDP policy with different penalties in the reward function for both the full and no communication case.



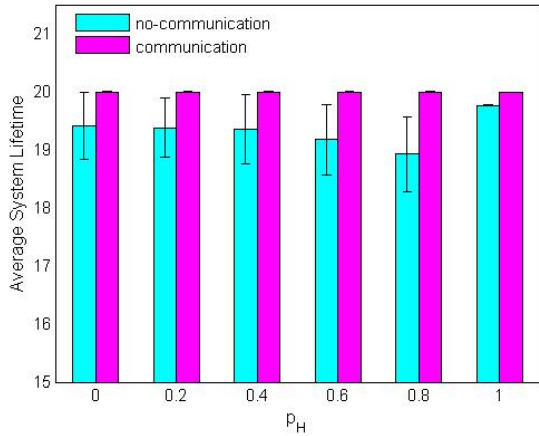
**Figure 7:** Effect of penalty in the reward function on the average system lifetime for both the full and no communication case.

#### 4.1.2 Effect of probability of change in criticality

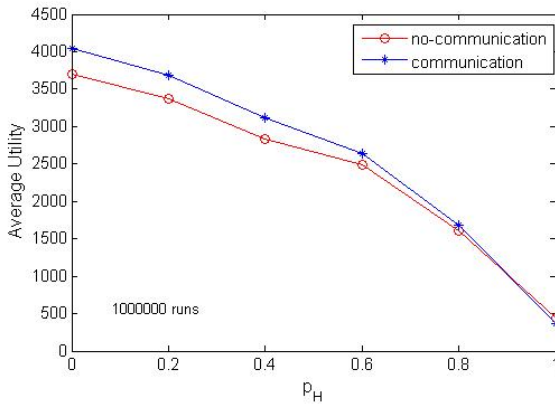
We changed the magnitude of the probability that was used in the transition function to model the change in event criticality with time ( $p_H$ ) to study its impact on the policy. This probability will be dependent on the specific application and the granularity of the model. Figure 8, Figure 9, and Figure 10 shows the system energy outage probability, lifetime, and utility 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. This is because the large magnitude of the penalty for running out of power dominates the sum of smaller rewards obtained for sampling during critical periods. The policy becomes more sensitive to the expected data criticality when the magnitude of the penalty is reduced (at the cost of a higher chance of the system running out of power).



**Figure 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.**



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



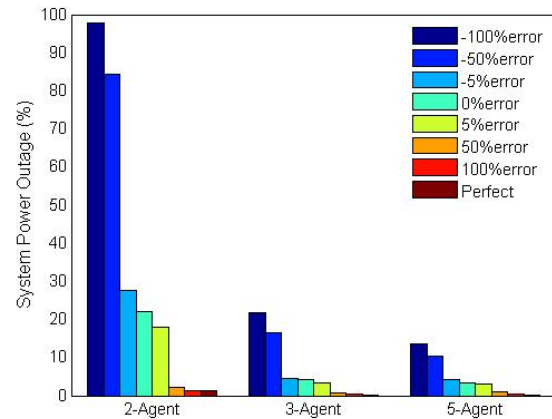
**Figure 10: Effect of probability of change in event criticality on the utility for both the full and no communication case.**

## 4.2 Sensitivity to errors in model parameters

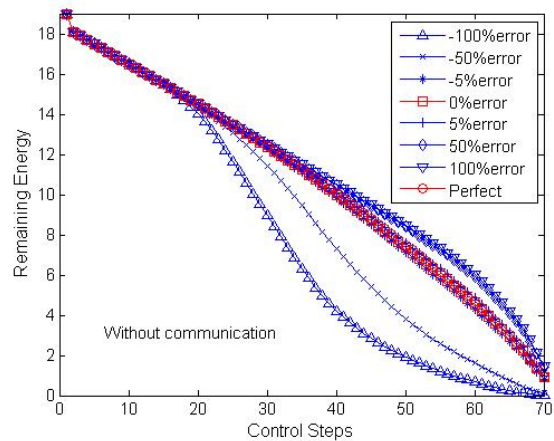
In the earlier simulations, it was assumed that the stochastic model used during offline policy computation exactly matched the stochastic energy consumption model of the physical system during execution. Here, we generate the optimal MDP policy by underestimating and overestimating the true rate of 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.

Figure 11 shows the effect that the model error has on the expected performance of the resulting policy and Figure 12 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 13 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 14, Figure 15, and Figure 16 show the probability of the system running out of power, average utility, 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 and utility, and decreases the percentage of system power outage.



**Figure 11: Probability of system power outage with varying amounts of model error ( $N = 2, 3, 5$ )**



**Figure 12: Remaining energy over time with varying amounts of model error and no communication ( $N = 2$ )**

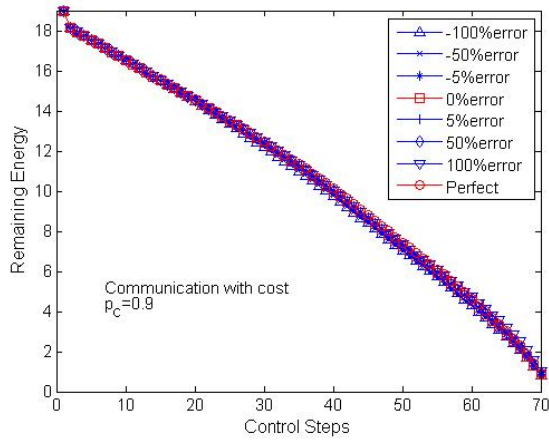


Figure 13: Remaining energy over time with varying amounts of model error and communication during policy execution ( $N = 2$ ,  $p_c = 0.9$ , threshold of communication = 1.75)

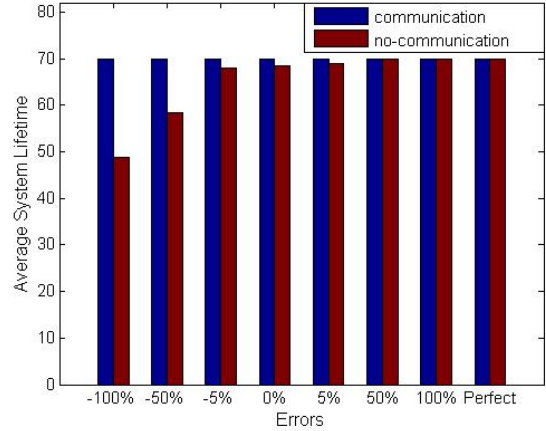


Figure 16: Average system lifetime with varying amounts of model error ( $N = 2$ )

### 4.3 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:

- “Min”: always sample at the lowest sampling rates.
- “Max”: always sample at the highest sampling rates.
- “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”)

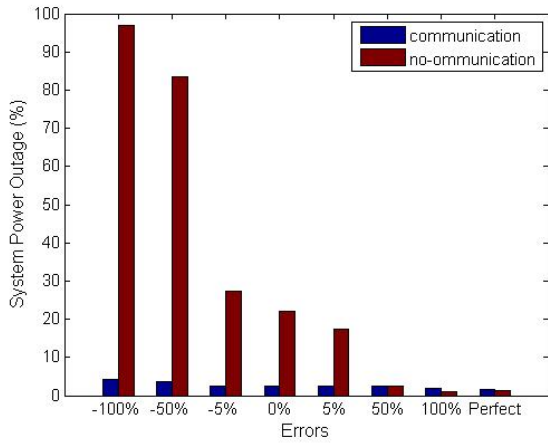


Figure 14: Probability of system power outage with varying amounts of model error ( $N = 2$ )

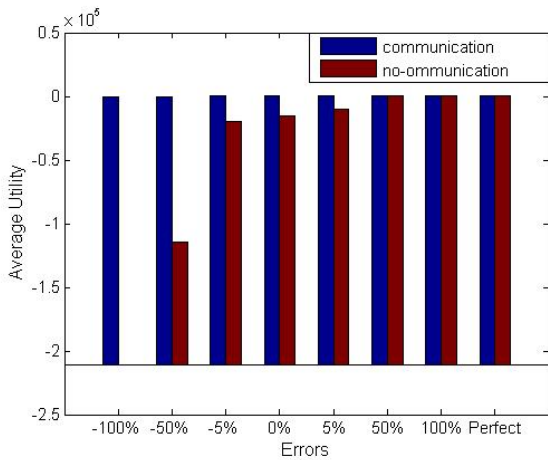


Figure 15: Average utility with varying amounts of model error ( $N = 2$ )

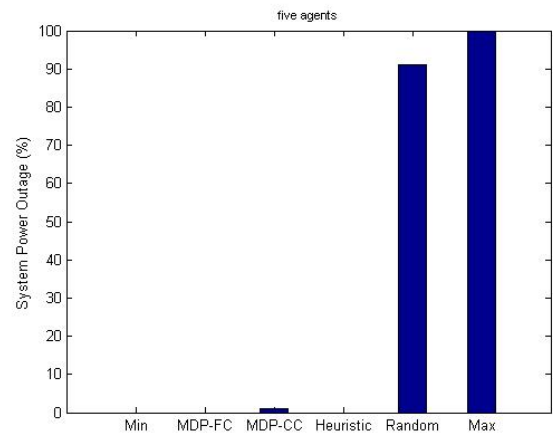
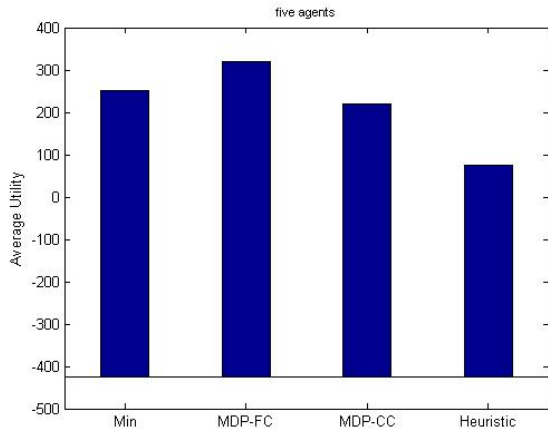


Figure 17: System power outage of MDP, Random, and Fixed policies ( $N = 5$ )



**Figure 18: Average system lifetime of MDP, Random, and Fixed policies ( $N = 5$ )**

and with a non-zero cost of communication (“MDP-CC”). The corresponding probability of system power outage is shown in Figure 17. Figure 18 shows the corresponding average utility. 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.

## 5. 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. 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.

## 6. ACKNOWLEDGMENT

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

## 7. REFERENCES

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

- [2] S. Liu, A. Panangadan, C. Raghavendra, and A. Talukder, "MDP Framework for Sensor Network Coordination," in *8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, San Francisco, USA, 2009.
- [3] 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.
- [4] 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.
- [5] R. Becker, S. Zilberstein, V. Lesser, and C. Goldman, "Transition-Independent Decentralized Markov Decision Processes," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2003, pp. 41-48.
- [6] 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.
- [7] 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.
- [8] R. Nair, P. Varakantham, M. Tambe, and S. Marsella, "Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings," in *Proceedings of IJCAI*, 2003.
- [9] D. Szer, F. Charpillet, and S. Zilberstein, "MMA\*: A heuristic search algorithm for solving decentralized POMDPs," in *Proceedings of IJCAI*, 2005.
- [10] D. S. Bernstein, S. Zilberstein, and N. Immerman, "The complexity of decentralized control of MDPs," in *Proceedings of UAI*, 2000.
- [11] M. L. Puterman, *Discrete Stochastic Dynamic Programming*: Wiley, 1994.
- [12] 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.
- [13] 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.
- [14] P. Liberatore, "On Polynomial Sized MDP Succinct Policies," *Journal of Artificial Intelligence Research*, vol. 21, pp. 551-577, 2004.