Markov Decision Processes for Control of a Sensor Network-based Health Monitoring System

Anand Panangadan and **Syed Muhammad Ali** and **Ashit Talukder***

Childrens Hospital Los Angeles/ University of Southern California
4650 Sunset Blvd., Los Angeles, California 90027
APanangadan@chla.usc.edu, syedmuha@usc.edu, talukder@usc.edu

Abstract

Optimal use of energy is a primary concern in field-deployable sensor networks. Artificial intelligence algorithms offer the capability to improve the performance of sensor networks in dynamic environments by minimizing energy utilization while not compromising overall performance. However, they have been used only to a limited extent in sensor networks primarily due to their expensive computing requirements. We describe the use of Markov decision processes for the adaptive control of sensor sampling rates in a sensor network used for human health monitoring. The MDP controller is designed to gather optimal information about the patient's health while guaranteeing a minimum lifetime of the system. At every control step, the MDP controller varies the frequency at which the data is collected according to the criticality of the patient's health at that time. We present a stochastic model that is used to generate the optimal policy offline. In cases where a model of the observed process is not available a-priori, we describe a Q-learning technique to learn the control policy, by using a pre-existing master controller. Simulation results that illustrate the performance of the controller are presented.

Introduction

A sensor network consists of a set of spatially distributed sensors that are able to communicate with each other using a low-power wireless interface (Pottie & Kaiser 2000). The exact communication structure and the relative distribution of sensors are application and design dependent (Estrin *et al.* 2001; Mainwaring *et al.* 2002; Talukder *et al.* 2004). Due to the primary application of sensor networks in mobile monitoring or monitoring of vast structures, the size and weight of each node needs to be compact and light. Therefore, sensors in a wireless sensor network typically have a very limited amount of computing power and storage onboard (Pottie & Kaiser 2000). Recent advances in the design of small, wearable sensors and faster processors combined with better signal processing techniques have made sensor networks practicable in applications such as autonomous environment monitoring (Mainwaring *et al.* 2002). These advancements

also make sensor networks suitable for autonomous long-term human health monitoring. For instance, a network consisting of heartbeat and temperature monitors may be "worn" by a patient with a history of heart ailment. This would enable a doctor to study the long-term behavior of these bio-signals or to automatically trigger an alarm if an unusual signal pattern is detected. This long-term data may also be useful in understanding the relationship between physiological processes as measured by bio-signals and the behavior patterns of the patient (Korhonen *et al.* 2001).

Limited advances in power technology however have created a major obstacle in long-term monitoring capabilities of sensor networks and have made energy the most critical resource in sensor networks, including those used for health monitoring. Additionally, a health monitoring application often requires time-critical and immediate response which imposes a new set of challenges that should be addressed and solved prior to deployment in the real-world:

1. **Limited Energy**: The lifetime of a sensor network is limited by the battery capacity of its individual sensor nodes. The power consumption at each node is dependent on the amount of sensing and computation performed at that node. Sensing at high rates limits the lifetime of the system. In a health monitoring application, the system must operate for at least some pre-fixed length of time before running out of power.

2. **Real-time Adaptive Sensing**: In continuous health monitoring, the desired quality of sensing varies depending on the health status of the patient being monitored. During periods of normal health, the sensor may be sampled at a lower rate as compared to periods of abnormal signal activity. This enables subsequent analysis of the signal at critical times in sufficient detail.

3. **Fault Tolerance to Component Failures**: The system must be resistant to component failures. The penalty for failure in health monitoring is high as an individual's life is at stake. Since the sensors in the network communicate using low-power radio transmissions, communication errors are possible. A distributed sensing system that is responsible for health monitoring should not fail if one sensor becomes disconnected from the rest of the network.

From the above discussion, it follows that energy is the most critical resource in a health monitoring system and

---

should be used sparingly and only when necessary. Energy is consumed during sensor operation, processing of data, and transmission of data. The amount of energy that is utilized during data acquisition is dependent on the sampling rate of the sensor. Moreover, the sensor sampling rate also determines energy consumed during data transmission since higher sampling rates generate larger amounts of data to be transmitted. We present a new event-based policy to determine the sampling rate of each sensor. In our system, the sampling rate of the sensors is regulated according to both (1) the current and projected health status of the patient and (2) the required minimum lifetime of the system.

Architecture for Patient Monitoring

In this work, we describe our real-time control of sensor operation in a sensor network based system for continuous long-term health monitoring, using a fault-tolerant two-stage control procedure. Multiple sensors are each connected to a node of the sensor network. Each node continuously transmits the sensed data via radio links to a central processing unit in the form of a handheld computer. The sensors that have been interfaced with the network include body temperature, heart-rate, blood oxygenation, and interstitial fluid (ISF) alcohol level sensors. The energy consumed at each sensor node depends on the sampling rate at that sensor. For instance, the ISF alcohol sensor requires a pump to draw out ISF before a measurement can be made. This is an energy-intensive process and hence reducing the duration for which the pump is in operation increases the lifetime of that node. The handheld computer has a centralized controller that makes coordinated control decisions. Additionally, each sensor node has a local controller that commences operation in the event of communication loss that invalidates the centralized controller. Figure 1 shows the components of the health monitoring system and Figure 2 shows the sensors and handheld computer attached to a person.
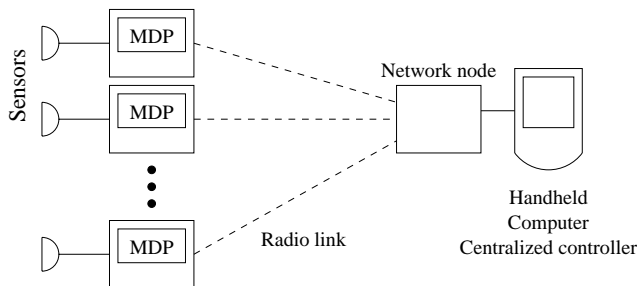

Figure 1: Block diagram of the health monitoring system.

We have earlier used Model Predictive Control (MPC) to implement the centralized controller to regulate the sensing frequencies of all the sensors (Talukder *et al.* 2004). Access to the state of the entire system allows the centralized controller to make optimal coordinated control decisions. This two-tier controller architecture contributes to the robustness of the system as a whole. During periods of complete network connectivity, the centralized controller regulates the individual sensing rates taking into account the data criticality

Figure 2: Person wearing sensors. The handheld computer is attached at the waist.

and reliability of each sensor. However, if a sensor is unable to communicate with this central node, its local controller then begins regulating the sensing rates so as to guarantee operation until the desired system lifetime. The focus of this paper is on the design of the local controller at each sensor.

We use the framework of Markov Decision Processes (MDP) (Russell & Norvig 2003) to control the sampling rate at each sensor. Each sensor node is equipped with an MDP controller to regulate the sensing frequency at that node according to the criticality of the data being measured. The controller also takes the minimum desired lifetime of the system as an input. The battery consumption is continuously monitored so that the system is in operation for at least the desired length of time. The main advantage of the MDP controller is that its policy can be computed offline. Only the policy table needs to be stored in the network node. Generating a control signal when the system is deployed corresponds to looking up the appropriate pre-calculated action (the sampling rate) from the policy table. Thus, the operation of this controller places a minimal computational load on the network node.

Other candidates for controller design include Partially Observable Markov Decision Processes (POMDPs) (Kaelbling, Littman, & Cassandra 1998) and MPC (Garcia, Prett, & Morari 1989). POMDPS have a greater representational power than MDPs as they can model the uncertainty in observing world state. MPC has been used extensively for industrial control of processes. However, algorithms to generate the control signal using these techniques are too computationally intensive to be run in real-time using the limited computing resources available at the sensor nodes.

Generating an optimal policy for the MDP controller requires that the rate of energy consumption with respect to sensing frequency and a stochastic model of the criticality of the data be known in advance. If this information is not available, we make use of the centralized controller to train the controller at each sensor.

Related Work

Markov decision processes have been used elsewhere for control of physical systems. Zilberstein *et al.* use MDPs to plan the path of a lunar rover (2002). As in our work, the ability to compute the entire policy offline and thus make only limited use of the computing power onboard the rover was listed as one of the advantages of this scheme. The state vectors used in modeling of the rover control problem represented the passage of time and resources and the quality of the scientific work being performed by the rover at every step. The Markov model was further factored into a hierarchical model to speed up the learning process.

Long-term human health monitoring systems have been built using different technologies (Korhonen *et al.* 2001; Ogawa, Tamura, & Togawa 1998). The system implemented by Korhonen *et al.* records signals such as heart-rate, and blood pressure (2001). However, the data from the sensors must be entered directly into a computer with the aid of the patient and therefore requires the participation of the patient. On the other hand, in the system designed by Ogawa, Tamura, & Togawa, the environment inhabited by the patient (for instance, the bed) was instrumented with sensors to record health information without the aid of the patient (1998). However, this means that the data can be collected only when the patient is within that environment. The advantage of using a wireless network of sensors is that sensors may be attached without regard to the location of the central data repository. Moreover, sensors may be physically added to or removed from a wireless monitoring system with relatively less effort as compared to a wired system or an instrumented environment. Other sensor network-based systems that are currently being developed for health monitoring include CodeBlue (Malan *et al.* 2004), LifeGuard (Montgomery *et al.* 2004), and Ubimon (Ng *et al.* 2004). However, these systems do not regulate energy usage by explicitly adapting the sampling rates of sensors.

Sensor networks have also been used for "health" monitoring in other application domains. A two-tiered sensor network architecture has been used for monitoring the state of physical structures (Kottapalli *et al.* 2003). Power saving is listed as the main advantage of this two-tier architecture.

Markov Decision Process modeling

A Markov Decision Process (MDP) is a 4-tuple (S, A, P, R). S is a finite set of states. The world exists in exactly one of the states in S. A is a finite set of actions that may be executed at any state. P is a probability function that defines how the world 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: $R : S \times A \rightarrow \mathcal{R}$. $R(s, a)$ 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 state $s \in S$. 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 $V^\pi(s)$, is the expected (discounted) sum of rewards obtained by following the policy π from s. The value function determines the action to be executed in state s under policy π:

$$argmax_{a \in A}(R(s, a) + \gamma \sum_{s' \in S} p(s, a, s')V^\pi(s)$$

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:

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

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 as follows.

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

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

The state of a sensor mote is represented by the state vector (t, h, p). $t \in \{t_1, t_2, \ldots, t_T\}$ indicates the time the system has been in operation. Since only a finite number of time-steps can be explicitly modeled, t_T corresponds to the guaranteed lifetime desired from the system. $h \in \{h_1, h_2, \ldots, h_H\}$ is a measure of the patient state as measured in the previous time-step, and $p \in \{p_1, p_2, \ldots, p_P\}$ is the amount of energy consumed. At every state, the system monitors patient health by sampling from the sensor at a particular rate. Denote the set of sampling rates by A.

The probability of transitioning from state (t_i, h_i, p_i) to state (t_j, h_j, p_j) while sampling at rate $a \in A$, $p((t_i, h_i, p_i), a, (t_j, h_j, p_j))$, is defined as $p_T(t_i, t_j)p_H(h_i, h_j)p_P(p_i, a, p_j)$ where

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

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

$$p_P(p_i, a, p_j) = \begin{cases} 1, & \text{if } i = P \\ p_P(a), & \text{if } i = j \\ 1 - p_P(a), & \text{if } j = i + 1 \\ 0, & \text{otherwise} \end{cases}$$

Thus, the state feature representing time is advanced at every time-step until the desired lifetime is reached by the system. p_H^{same} and p_H^{change} model the change in patient's health status stochastically. The rate at which energy is consumed by the system is modeled with probabilities $P(a), a \in A$.
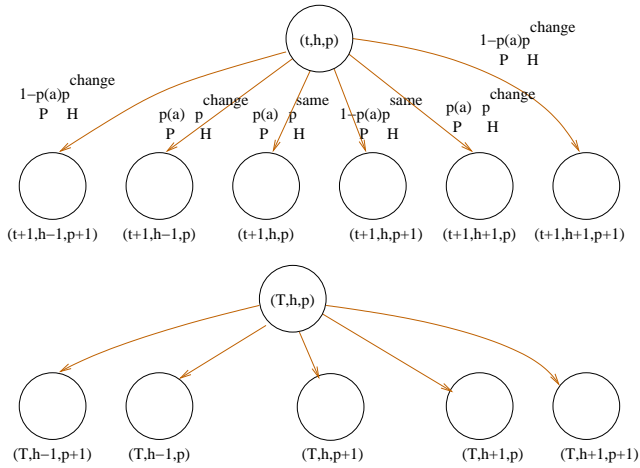
Figure 3: State transitions in the MDP model. The current state is labeled (t, h, p). The transitions of the model fragment below are for the case when $t_i = t_T$.

The rate is higher for higher sensing rates. The reward function depends only on the sensing rate and the health status of the patient. Intuitively, the sensing rate should be higher during critical times. There is also a penalty, $-R^{\text{powerout}}$, for the system running out of power before the desired lifetime. Thus, the reward function is defined as:

$$R((t, h, p), a) = \begin{cases} -R^{\text{powerout}}, & \text{if } p = P, t < T \\ k_R.a.h, & \text{otherwise} \end{cases}$$

where k_R is a constant of proportionality. These model update rules are illustrated in Figure 3.

MDP learning using centralized controller

Generating an optimal policy using value iteration requires a stochastic model describing the time evolution of the criticality of the data and the energy consumption of the sensor. If such a stochastic model is not available, we learn the policies using reinforcement learning. The reinforcement is obtained from our pre-existing global controller. The MPC mechanism that is used to implement this centralized controller is described in detail in (Talukder *et al.* 2004).

Q-learning is a reinforcement learning procedure to determine the optimal policy of an MDP when the model parameters are unknown (Sutton & Barto 1998). For every state-action pair (s, a), define its "Q" value as

$$V(s) = \max_{a \in A} Q(s, a)$$

Unlike value iteration which required the model transition probabilities to calculate the optimal V values, Q-learning calculates the Q values from a given state-action sequence and its corresponding rewards. When the agent performs action a in state s, receives reinforcement r and moves to state s', the following update rule is applied:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a' \in A} Q(s', a') - Q(s, a))$$

Table 1: Parameters determining the transition probabilities

p_H^{same}	0.7
p_H^{change}	0.15
$p_P(a)$	$0.5 + \left(\frac{0.9 - 0.5}{20}\right) a$

where α, $0 < \alpha < 1$ is the learning rate. The Q values converge to their optimal values if every state-action pair is updated an infinite number of times.

During the learning phase, the agent has to make a trade-off between exploration and exploitation of the policy that is being learned, i.e., whether to execute the learned policy at a state or to act randomly to explore an untested action. One solution is to select the action to be executed from a Boltzmann distribution based on the Q-values of the candidate actions (Sutton & Barto 1998). In our system, we have access to the centralized controller which produces a reinforcement signal for any state of the MDP policy table. Hence, in our learning phase, we select states that have not received reinforcement and generate actions from this state. This ensures that the Q-values for all states converge faster as compared to the case where the states are always updated in the order in which they appear in a learning trial.

Results

We now present simulation results using a Markov model obtained by discretizing the guaranteed lifetime of the system into $T = 50$ time-steps, battery capacity into $P = 30$ levels, and criticality of the patient's health status into $H = 20$ levels. Thus, this model has 30,000 states. Our health monitoring system is implemented using commercially available Mica2 motes which has a program memory of 128KB (Horton *et al.* 2002). Thus, the entire MDP policy table can be fitted into memory. The transition probabilities are determined by the parameters listed in Table 1.

In a real application, these parameters would be determined by the specific characteristics of the domain. For instance, in a physical activity monitoring system, periods where heart-rate increases beyond 120bpm are the most relevant since at that range there is a linear relationship between heart-rate and calorie expenditure (Saris 1986). The number of modeled time-steps, T, would be determined by how often the sampling rate of a sensor can be changed and the desired life-time of the system.

Policy from Value Iteration

Figure 4 shows the sensing frequency and energy consumption at every time-step from one simulation trial. Note that the available energy did not drop to zero during the length of the trial. The patient criticality data was obtained from a sensor that measures ISF alcohol levels.

We explored the effect of a mismatch between the stochastic energy consumption model that is used in the MDP controller and the actual rate of energy consumption. Figure 5 shows the energy consumed during the lifetime of the system using the same patient criticality data used in Figure 4. As the mismatch between the model and the real
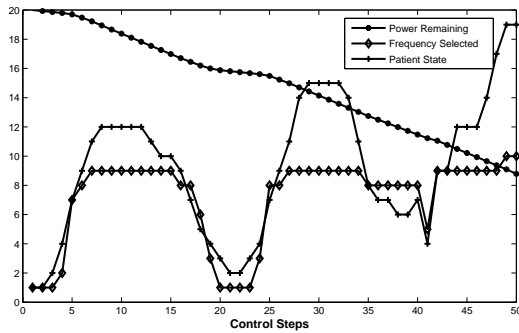
Figure 4: Sensing frequency generated by the MDP and energy consumed (y-axis) at every time-step (x-axis).

energy consumption rate increases, the energy used by the system increases. However, the system does not run out of power before the desired lifetime until the mismatch reaches 200%. The sensing frequencies at every time-step during the trials with mismatched energy consumption models is shown in Figure 6. As the amount of mismatch increases, the sensing frequency at times close to the end of the system lifetime do not increase with an increase in the criticality of the patient's health status.
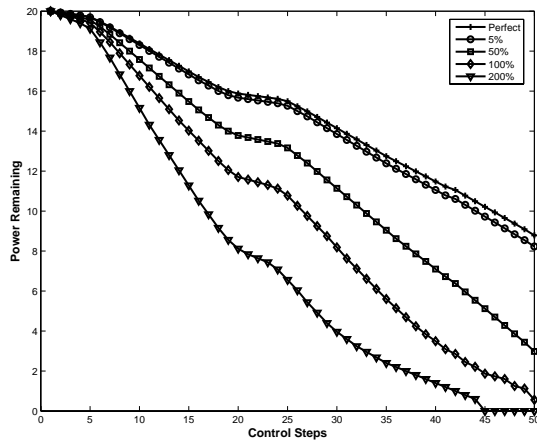

Figure 5: Energy consumed during the lifetime of the system when the energy consumption model used in the MDP controller does not match the actual consumption rate.

Policy from Q-learning

We performed similar simulation trials when the policy was learned using Q-learning with the centralized controller providing the reinforcement. Figure 8 shows the energy consumption when there is a mismatch between the model and the real energy consumption rate. As in the earlier case, the system does not run out of power before the desired lifetime until the mismatch reaches 200%.
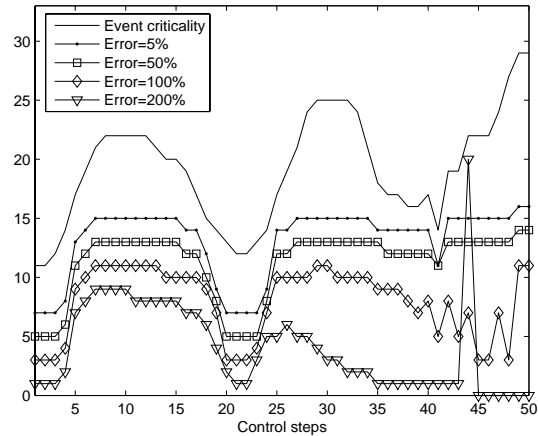

Figure 6: Event criticality and sensing frequencies during the lifetime of the system when there is a mismatch between the energy consumption model and the actual energy consumption rate. (The frequencies have been offset relative to each other to enhance viewability).

A comparison of the power consumption rates when the policy is obtained from value iteration and from learning indicates that the learned policy exhausts the sensor power at the end of the desired lifetime while in the case of the value iterated policy, some power remains. This reflects a bias in the MPC controller (that was used to generate reinforcement) to exhaust power at the end of the desired lifetime.

Conclusions and Future Work

We described how a Markov decision process can be used to control of sensor sampling rates in a sensor network for human health monitoring. The controller was able to guarantee the minimum lifetime of the system by varying the resolution at which the data is sampled according to a model of the future criticality of the patient's health. We presented a stochastic model that was used to generate the optimal policy offline. The entire policy can be fitted into the program memory of a node in the sensor network and since execution of the policy requires only a memory lookup, this mechanism utilizes minimal computational resources. In cases where a model is not available, we described how the control policy could be learned from a pre-existing controller.

We are currently developing algorithms to enable the learning of MDP control policies that take into account the influence of sensors in the network on each other (i.e., the learning of multi-agent MDP policies). In multi-agent MDPs, the actions are distributed among different agents (Boutilier 1999). For our health monitoring application, we will learn individual policies without the aid of a joint model but from the outputs of the centralized controller. As the number of states required for control in a particular application may be larger than the memory capacity of a sensor node, we are developing policy approximation methods to reduce the memory requirements of the policy.
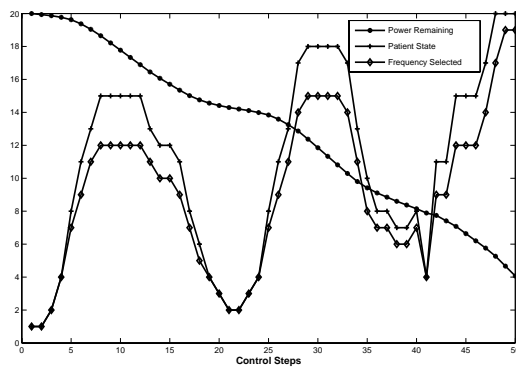
Figure 7: Sensing rate obtained from the learned policy and energy consumed (y-axis) at every time-step (x-axis).
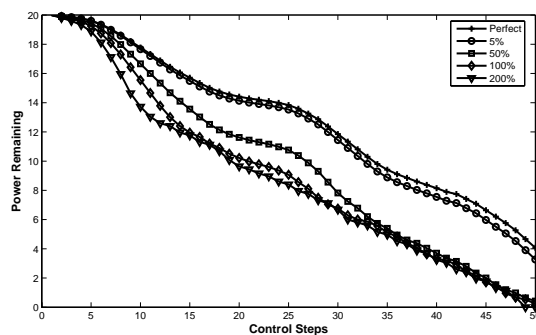

Figure 8: Energy consumed during the lifetime of the system when the energy consumption model used during learning does not match the actual consumption rate.

Acknowledgment

References

Boutilier, C. 1999. Sequential optimality and coordination in multiagent systems. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Estrin, D.; Girod, L.; Pottie, G.; and Srivastava, M. 2001. Instrumenting the world with wireless sensor networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Garcia, C.; Prett, D.; and Morari, M. 1989. Model predictive control: Theory and practice – a survey. *Automatica* 25:335–348.

Horton, M.; Culler, D.; Pister, K.; Hill, J.; Szewczyk, R.; and Woo, A. 2002. MICA, the commercialization of microsensor motes. *Sensors* 19(4):40–48.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101:99–134.

Korhonen, I.; Iivainen, T.; Lappalainen, R.; Tuomisto, T.; Koobi, T.; Pentikainen, V.; Tuomisto, M.; and Turjanmaa, V. 2001. TERVA: system for long-term monitoring of wellness at home. *Telemedicine Journal and e-Health* 7(1):61–72.

Kottapalli, V. A.; Kiremidjian, A. S.; Lynch, J. P.; Carryer, E.; Kenny, T. W.; Law, K. H.; and Lei, Y. 2003. Two-tiered wireless sensor network architecture for structural health monitoring. In *10th Annual International Symposium on Smart Structures and Materials*.

Mainwaring, A.; Polastre, J.; Szewczyk, R.; Culler, D.; and Anderson, J. 2002. Wireless sensor networks for habitat monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*.

Malan, D.; Fulford-Jones, T.; Welsh, M.; and Moulton, S. 2004. Wireless sensor networks for habitat monitoring. In *International Workshop on Wearable and Implantable Body Sensor Network*.

Montgomery, K.; Mundt, C.; Thonier, G.; Tellier, A.; Udoh, U.; Kovacs, G.; Barker, V.; Ricks, R.; Davies, P.; Yost, B.; and Hines, J. 2004. LifeGuard - a personal physiological monitor for extreme environments. In *Medicine Meets Virtual Reality (MMVR04)*.

Ng, J. W. P.; Lo, B. P. L.; Wells, O.; Sloman, M.; Toumazou, C.; Peters, N.; Darzi, A.; and Yang, G. Z. 2004. Ubiquitous monitoring environment for wearable and implantable sensors (UbiMon). In *International Conference on Ubiquitous Computing (Ubicomp)*.

Ogawa, M.; Tamura, T.; and Togawa, T. 1998. Automated acquisition system for routine, noninvasive monitoring of physiological data. *Telemedicine Journal* 4(2):177–185.

Pottie, G., and Kaiser, W. 2000. Wireless integrated network sensors. *Communications of the ACM* 43(5):51–58.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A modern approach*. Pearson Education.

Saris, W. H. M. 1986. Habitual physical activity in children: methology and findings in health and disease. *Medicine and Science in Sports and Exercise* 18:253–263.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning*. The MIT Press.

Talukder, A.; Bhatt, R.; Sheikh, T.; Pidva, R.; Chandramouli, L.; and Monacos, S. 2004. 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.

Zilberstein, S.; Washington, R.; Bernstein, D. S.; and Mouaddib, A.-I. 2002. Decision-theoretic control of planetary rovers. In Beetz, M., ed., *Plan-Based Control of Robotic Agents*, volume 2466. LNAI. 270–289.