Resource-Aware Wireless Sensor-Actuator Networks

Amol Deshpande
University of Maryland
amol@cs.umd.edu

Carlos Guestrin
CMU
guestrin@cs.cmu.edu

Samuel R. Madden
MIT
madden@csail.mit.edu

Abstract

Innovations in wireless sensor networks (WSNs) have dramatically expanded the applicability of control technology *in day-to-day life, by enabling the cost-effective deployment of large scale sensor-actuator systems. In this paper, we discuss the issues and challenges involved in deploying control-oriented applications over unreliable, resource-constrained WSNs, and describe the design of our planned* Sensor Control System (*SCS*) *that can enable the rapid development and deployment of such applications.*

1 Introduction

Embedded sensing and control systems are used on a daily basis by nearly all consumers and businesses in the industrialized world. These systems are critical in manufacturing, automobiles, aviation, building heating and air-conditioning, power distribution, and a huge array of other domains. With dramatically dropping hardware prices, wireless sensor networks (WSNs) are finally becoming a reality; *low-power, wireless embedded control* systems have the potential to significantly alter and expand the applicability of control technology. In industrial control settings, for example, wireless networks can be installed for a fraction of the cost of wired devices (one study estimates that *each* physical wire in a commercial workplace costs $800.00 to install [22]), and can provide unprecedented flexibility, with high density sensing and deployments in unsafe areas that may be impossible to instrument with standard wired approaches (such as inside waterways, or in high-temperature oil refineries).

Realizing this potential, however, requires the computer science community to develop novel software technologies that can enable rapid development and deployment of wireless sensing and control systems. Though there has been much interest in developing such software for deploying and maintaining pure data collection systems, developments in integrating battery-powered, wireless technologies into closed-loop control settings have been limited. In this paper, we describe the design of the *Sensor Control System* (*SCS*) that we are currently building. *SCS* consists of an application infrastructure and a suite of embedded systems software that is *designed to allow rapid deployment of control-oriented wireless applications*.

As a simple example of a control-oriented wireless sensor network deployment, we consider systems for *building control.* Modern buildings typically feature highly sophisticated heating, ventilation and air conditioning (HVAC) systems that enable fine-grained monitoring and control over the HVAC settings. As with many embedded control systems, these systems typically have four major components: (1) A set of *preferred temperature levels* that the building users specify; (2) *sensors* that monitor the temperature, humidity, etc. in various

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

parts of the building; (3) *actuators* that control the heating/cooling devices such as air conditioning vents and furnaces; and (4) a *controller* which implements a *control law* that takes the current sensor values and selects a setting to the actuators that attempts to fulfill the users' preferences. In addition to monitoring temperatures, the sensor network can monitor occupancy in the building to optimize HVAC and lighting settings to the needs of the current occupants. Due to ever increasing energy costs, better monitoring and control of such systems can significantly reduce the building's operation expenses.

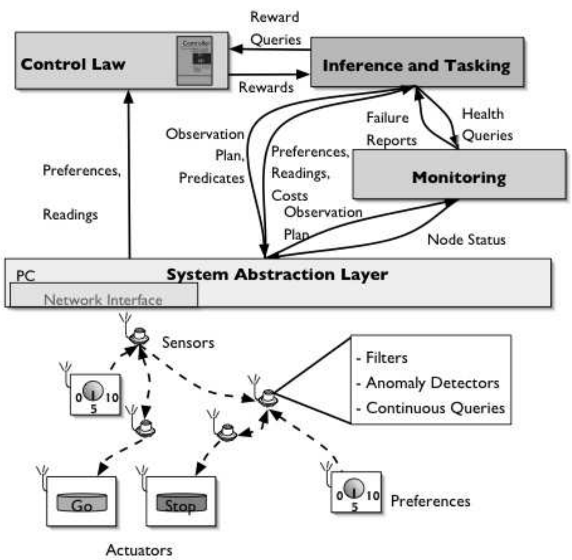


Figure 1: The basic architecture of the *SCS* platform.

HVAC systems are one example of a larger set of control systems in offices and homes that can benefit from wireless sensor networks. A modern home contains a large number of appliances and equipment that can be both remotely monitored and controlled. Besides the heating and cooling system, such systems include lights, digital media equipment, and major appliances like refrigerators and ovens. The possibilities for automation using sensing devices are numerous – for example, a user might express a preference such as "keep the floor clean", and "do not run the noisy robotic vacuum cleaner when I am listening to music". This is a simple control system that uses sensors that detect dirt and the presence of a user, and actuates the vacuum cleaner as needed.

Unfortunately, low-cost wireless networks are often *resource constrained*, with limited power, computation and communication capabilities, and are *unreliable*, sensors can fail due to manufacturing defects, power overuse, or environmental impact. These two real-world limitations of sensor networks suggest opposite solution approaches: resource constraints suggest that we should we reduce sensing and communication to a minimum, while unreliability suggests that we add sensing redundancy and increase data collection rates. To address these conflicting needs, the approach we advocate uses statistical models and control theory to design robust control laws and tailor sensing requirements depending on the precision needed by the control law at each moment in time. Our resource-aware approach builds on this theory by choosing the sensor readings to capture and actuators to adjust that will incur the minimum energy cost to the system subject to the constraint that users' preferences regarding the state of the system are approximately satisfied.

1.1 Challenges

There are significant challenges that need to be addressed to enable the effective deployment of control systems with sensor networks.

Resource limitations of WSNs: Large networks of wireless sensors and actuators pose a number of problems that are not present in smaller networks of wired devices. Although current generation devices, such as the Mica motes, have limited processors (motes have an 8-bit, 7 MHz processor) and memory (motes have 4 KB of RAM and 512 KB of Flash), we expect that in a few years these limitations will be much less severe. However, there are a number of other limitations we anticipate will still be significant for the foreseeable future [16].

First, the *network channel is lossy and has variable latency*, which means that control applications have to be tolerant of missing and delayed data. Loss rates can be 50% or greater when multiple nodes are transmitting simultaneously; retransmissions can mitigate this, at the expense of increased and unpredictable latency as well as additional energy utilization [28, 29]. Second, *sensor readings are noisy and subject to drift over time*, even when carefully calibrated [26]. Third, *devices fail*, as batteries run out or harsh environmental conditions damage hardware, meaning that the control system needs to be prepared to detect those failures and work around them to satisfy the user-specified control law as best as possible [26]. Finally, the battery-powered nature of most WSNs means that *power is of utmost importance, and must be aggressively conserved* [21, 11].

Ease of use and deployment: The second major class of challenges relates to the difficulty associated with deploying, configuring, and learning how to use automated control systems, particularly in home automation settings [19]. To reap the full benefits of the technology trends in sensing and actuating technology, we must make this deployment process much less difficult than it is today.

There are many aspects to the user-system interaction in this setting. First, *deploying a sensor network should not require knowledge of the low-level functioning of sensor networks*, or the wireless technology. Similarly, addition and removal of sensors should be made very easy. Second, the specification of the *control law* that specifies the optimization goal should be intuitive and should not require detailed knowledge of control theory. Finally, the system should be able to *hide the peculiarities of the sensor networks* such as dynamic topologies, missing data and/or lossy sensors from the user.

1.2 Prior Work

Though there has been much work in developing and deploying embedded control systems that use *wired* sensors and actuators [2, 20], using low-power wireless sensor-actuator networks fundamentally changes the nature of the problem because of the bandwidth and power limitations of these networks (Section 2). Wireless sensor networks itself has been a very active area of research in recent years [1, 30], but most of this work has focused on the *sensing* aspect of WSNs, and not as much on *actuation*. Due to lack of space, we omit a detailed discussion of the prior work, and refer the reader to the full version of the paper [7].

1.3 Planned System Architecture

Figure 1 illustrates the design of our planned sensor control system, *SCS*. Main components of the system are:

1. *Inference and tasking component:* This component is responsible for deciding which sensors and preferences to observe, for inferring which readings will be most valuable and which actuators will best address the user's needs.
2. *Control law component:* This component encapsulates the relationship between the sensor readings, users preferences, and state of the actuators.
3. *Monitoring component:* This component is responsible for determining the status of each of the nodes of the network (*e.g.,*, whether it is online, whether its sensors are functioning properly, etc.).
4. *System abstraction layer:* The system abstraction layer is responsible for collecting preferences and sensor values from the network and for disseminating actuator values throughout the network while simultaneously managing the wake/sleep schedule of the nodes, synchronizing their clocks, and managing their interface to low level actuators and sensors.

In the rest of the paper, we briefly discuss issues in applying control theory in wireless sensor networks, and elaborate upon the components of *SCS*.

2 Applying Control Theory in Power-Constrained Sensor Networks

The control and monitoring systems described in Section 1 share three common requirements: observing the state of the system, obtaining user preferences, and disseminating the action choice, or actuator setting. In settings that combine wireless networks with embedded control, we must fulfill these requirements while simultaneously addressing resource constraints, such as limited power, low network bandwidth, changing link qualities, and sensor noise and faults. In this section, we formalize the requirements for a general platform for embedded control using sensor networks.

2.1 Sensor Network Abstraction for Embedded Control

Embedded control systems typically use a *control law* to decide appropriate settings for various actuators given the current state of the system. More formally, a *closed-loop control law* $\pi(\mathbf{x})$ provides a mapping from states $\mathbf{x}$ of sensors and user preferences to actions $\mathbf{a}$ [4, 2, 20] taken on the actuators. For example, in HVAC, the control law will describe whether air conditioning vents need to be opened or closed, given the current temperatures at every location. More generally, if there is uncertainty about the system state, *e.g.,* due to unreliable sensors, the control law maps a probability distribution $\boldsymbol{b}(\mathbf{x})$ over the state of the system (also called a belief state) to an action choice [25, 12, 15]. Again in HVAC, if the sensors are making *noisy* observations of the temperature values, instead of having perfect knowledge of the state, we might only have a probability distribution over the possible states of the system. The action choice then depends on the distribution, *e.g.,* on its mean and variance.

In *SCS*, the sensor network provides the control law with a probability distribution over the possible states of the system $\boldsymbol{b}$, and with the current setting for the user preferences. These are used by the control law to specify the current action choice, which is then disseminated to the specific actuators in the network. In order to specify the belief state $\boldsymbol{b}(\mathbf{x})$, we need a probabilistic model that relates the noisy observations made by the sensor network with the true state of the system [13, 23]. To do this, we need: (1) a *sensor model* that specifies a probability distribution over possible sensor measurements $\mathbf{o}$ given the current state of the system $\mathbf{x}$, and (2) a *prior distribution* that specifies our initial belief about the system state. Consider an HVAC system with a single temperature sensor i with Gaussian noise. In that case, the sensor model $P(o_i \mid x_i)$ would have Gaussian distribution, centered around the *true* temperature. The prior $P(x_i)$ could also be a high-variance Gaussian that represents the usual distribution of temperatures during the current time of year. Given these, we can obtain our belief state $\boldsymbol{b}(x_i \mid o_i)$ as the distribution over possible states after the sensor value is observed (also called the *posterior distribution*) using Bayes rule:

$$\boldsymbol{b}(x_i \mid o_i) = P(x_i \mid o_i) \propto P(o_i \mid x_i)P(x_i).$$

In a more general setting, we may have observed a subset $\mathcal{O}$ of a total of n sensors, and would like to compute the posterior distribution over the state of the whole system $\mathbf{x}$ given the observed value $\mathbf{o}$. Note that in general *hybrid* systems the state $\mathbf{x}$ contains both discrete and continuous variables [24, 3]. We again use Bayes rule:

$$\boldsymbol{b}(\mathbf{x} \mid \mathbf{o}) = P(\mathbf{x} \mid \mathbf{o}) \propto P(\mathbf{o} \mid \mathbf{x})P(\mathbf{x}). \tag{1}$$

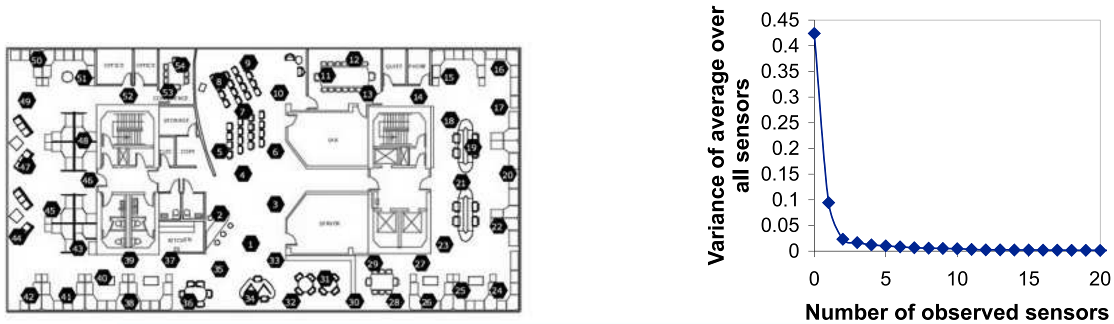
In HVAC, this belief state is the posterior probability of possible temperatures at all locations, given the observed sensor values at a subset of the locations. The prior distribution $P(\mathbf{x})$ and the sensor model $P(\mathbf{o} \mid \mathbf{x})$ can be learned from historical data using standard machine learning algorithms [18, 8].

Control systems usually deal with processes that evolve over time. The evolution of real systems is usually uncertain, and should also be modeled in terms of probability distributions by a process called *filtering* [13, 23]. We omit the details of this process due to lack of space.

2.2 Limited Sensing, Preference Acquisition and Action Dissemination

Collecting sensor values and user preferences, and disseminating action choices as dictated by the control law, requires significant energy expenditure in wireless sensor networks. Figure 2 (i) shows a wireless network we deployed at the Intel Lab in Berkeley. In this deployment, if we were to continuously collect data, the network would only survive for a few days due to battery capacity constraints. Thus, we strive to minimize the amount of data we must capture.

Although a few control systems take such sensing costs into account [23], most do not have such capabilities. Furthermore, in wireless sensor networks, the cost of sensing is non-local: since we need to traverse the network to collect sensor values, the cost of data gathering is non-linear in the number of data points. Though the cost of making a temperature measurement is independent of the sensor chosen, the cost of collecting the data can vary



(i) (ii)

Figure 2: (i) Layout of sensor network deployment with 54 nodes; (ii) Variance of the *average* temperature decreases rapidly in the beginning as observations are made sequentially.

widely with the actual subset collected. For the lab deployment above, the cost of collecting data from a sensor varies from a minimum of 5 mJ to a maximum of 191 mJ depending on the sensor chosen. Additionally, control systems rarely consider the cost of collecting user preferences and disseminating action choices which, again, are non-linear and possibly very high.

Fortunately, sensor measurements are highly correlated. Figure 2 (ii) illustrates this on real temperature data from our lab deployment, plotting the variance of the value of average temperature over the sensor network as we observe the values of the sensors sequentially. As we can see, though the initial variance in our estimate of the average temperature is quite high, it decreases very rapidly as observations are made, thus allowing us to estimate the correct average with very high accuracy with only a few observations.

We have recently developed a framework for exploiting such correlations between sensor values to significantly reduce the amount of communication and sensing required in the network [6]. By collecting only a subset of observations, the energy savings can be very significant. However, our estimate of the belief state, $\boldsymbol{b}^t$, at time t, may become less certain than when all sensor values are incorporated. We must be careful, as this additional uncertainty could potentially cause a significant decrease in the effectiveness of the control law, and bound the effect of approximate estimates of the belief state on the control law [2].

In addition to limiting the observations of sensor values, *SCS* also considers the potential to costs of disseminating action choices and collecting user preferences using models and uncertainty. Some research has been done in the context of uncertainty over user preferences, considering probability distributions over possible preferences [5]. We can build on this framework, using techniques analogous to those in [6] to minimize the amount of preference information communicated.

Similarly, the actuation setting may not change significantly from one time step to the next. In such cases, we can save energy and bandwidth by not updating the actuation values at some nodes. In particular, if the new actions do not change the reward function significantly from the optimal value, they should not be disseminated. Here, again, we can use control theory bounds to estimate the loss in rewards from these local perturbations [2].

2.3 Resource Management and Optimization

Given the setting outlined thus far, we can formalize a very significant challenge that we plan to address: **resource optimization in sensor networks for control systems**. In particular, our goal is the selection of the subset of sensor values and preferences to retrieve, and new action settings to disseminate that will utilize minimum power resources, while still provably maintaining bounds on the quality of the control law. This selection process includes three important challenges that we address in *SCS* :

1. **Cost estimation:** Once we select a set of nodes (actuators to disseminate actions, and sensors to collect data), we must determine the best protocol to visit these nodes, and the expected cost of running this

protocol on them. Though we have done some initial work in this direction [17], designing a efficient data collection protocol that can function in presence of arbitrary failures is one of our research goals. In Section 3, we present an active network monitoring component on which this protocol will be based.

2. **Decrease in control quality due to limited data collection and action dissemination:** Here, we are faced with a very interesting theoretical challenge. We have three sources of loss:
   - **Sensing loss:** By observing only a subset of the nodes, our estimate of $\boldsymbol{b}^t$ may be more uncertain. We can bound the effect of such uncertainty using the techniques in [2].
   - **Preference loss:** User preferences can change over time, if these changes are not monitored continuously, we may have uncertainty over user preferences. This issue can be tackled by a combination of probabilistic models of preferences [5] and sensitivity analysis techniques for control laws [27].
   - **Action dissemination loss:** If actuators are not informed of changes in the desired action, the quality of the control law may decrease. An initial solution for this problem can be based on the sensitivity analysis [2], using the difference between the optimal action setting and the action setting that the nodes are currently using. A novel, potentially more effective approach that we are developing as part of *SCS*, is to perform a constrained optimization of the action selection mechanism. Given that we are going to visit a subset of the nodes, we should compute the best action choice for this subset, assuming that the other nodes' action choices remain constant. This constrained optimization approach can potentially lead to significantly lower loss in the quality of the control strategy.

3. **Optimization of subset of nodes:** Finally, given the cost function and the estimate for the quality loss of the control law, we must solve the computationally challenging problem of *optimizing the subset of visited nodes* given a maximum allowable decrease in the quality of the control law. In our previous work [6], we have proposed a simple, but highly effective, heuristic for the (narrower) problem of selecting a subset of nodes to visit to most efficiently increase the confidence in the estimate of the global system state. Even though the general problem of selecting which nodes to visit, which attributes to acquire, and which actuations to perform at every time step seems extremely challenging at first, we believe that it is possible to develop efficient approximation algorithms for this problem. In addition, for many practical cases, we believe that the use of machine learning techniques can allow us to obtain provably near-optimal solutions.

3 Active Network Monitoring as a Resource Optimization Problem

Effective monitoring of the sensor network system itself is a central part *SCS*, since our algorithms depend on up-to-date topology and correlation information to function most efficiently. Hence, a core piece of *SCS* is a tool that monitors the network topology (including link quality and network dynamics), the status of individual nodes (*e.g.,*, remaining battery life, whether they are failed or producing noisy readings, etc.), as well as the overall quality of the probabilistic models used to drive the control system. This last condition is particularly important as it allows us to decide when our models need to be re-learned, *e.g.,*, from a new set of historical data.

In addition to enabling our optimization algorithms, and allowing us to perform maintenance decisions, a monitoring tool will also enable us to address another interesting challenge: making decisions about where to place sensors. The basic idea is as follows: if our state estimates $\boldsymbol{b}^t$ are too uncertain to satisfy user-specified certainty goals, our monitoring system should also allow us to determine the positions of new sensors that will most likely decrease uncertainty. Similarly, if communication quality is too low, our monitoring tool should recommend placements that will help improve the communication topology.

The network topology, the status of a sensor, or the quality of a model are all descriptions of the current state of the system analogous to $\boldsymbol{b}(\mathbf{x})$ in Section 2.1. We can thus view the monitoring problem in the same resource management light described in Section 2. However, there are some interesting research challenges specific to the monitoring problem:

1. **Failure models:** Models for outlier detection and failures modes are essential for effective monitoring. Our goal is to scale existing machine learning techniques for this purpose (*e.g.,*, [8, 14]) to work with the limited resources and distributed, dynamic regime of wireless networks.
2. **Model selection and evaluation:** Again, building on statistical techniques (*e.g.,*, [10]), we can evaluate the quality of the fit of our control and fault models to the observed set of sensor readings and actuator responses. This evaluation step allows us to detect inappropriate models and retrain the system as needed.
3. **Resource management:** The optimization algorithms in Section 2.1 also need to be extended to incorporate these outlier detection and fault models. A general, unified framework of this sort can significantly decrease the resource load on the sensor network by allowing data used for outlier detection to also be used in our control-oriented planning algorithms.
4. **In-network component:** A sensor node, or nearby nodes, should be able to detect failures more rapidly locally than is possible from a central base station. On the other hand, the base station has a "global" view that can detect widespread correlated failures. Thus, we are developing an approach that integrates an in-network detection component with our centralized model-based tool to balance this tradeoff appropriately.

Although learning failure models and model selection are relatively well-studied tasks in the machine learning literature, the resource management and in-network components are novel, and require significant new research. Others in the systems community have looked at tools for sensor network deployment and management [16, 9]. We believe our approach, however, can detect failures and recommend placements for new nodes in ways that other tools cannot by exploiting statistical correlations and models of network behavior.

4 A Low-Level System Interface for Sensornet Control

To hide the low-level functioning of the sensors and actuators from the higher level control and monitoring facilities, we utilize a *system abstraction layer* that provides the following basic API:

1. *Estimate the current network topology and link qualities*: from networking data collected over time.
2. *Observe one or more sensors or preferences:* Using the current network topology, or a user specified path, collect readings from a specified set of sensors or preferences and notify the user as these arrive.
3. *Set the states of one or more actuators*: Using the current network topology, route action choices to a subset of the actuators.
4. *Continuously monitor the state of a subset of the nodes*: Using a user-specified predicate, continuously monitor those devices for sensor readings or new preferences that satisfy the predicate.

Though this API may appear to be missing many of the features one would expect to find in a "general purpose" interface to WSNs, we believe that this API will address a wide range of applications, and its simplicity will allow us to provide effective response times and power efficiency. One of the long-term research challenges is to refine this API such that it strikes the appropriate balance between simplicity, expressiveness and efficiency.

5 Conclusions

In this paper, we described the design of our planned sensor control system, *SCS*, that is aimed towards enabling rapid development of software control and monitoring systems using low-power wireless networks. The design of our system is motivated by both the needs of real-world monitoring and control systems, such as heating, ventilation, and air conditioning problems (HVAC), process control systems (PCS), and health monitoring of industrial equipment (HM), and the peculiarities of power-constrained wireless sensor networks not observed in more prevalent wired sensor networks. Our platform works by combining techniques from machine learning and statistics with high-level abstractions inspired by work in software systems and databases, allowing for mathematically-sound decision making despite the loss and uncertainty that is inherent in these networks. *SCS* also insulates users from the low-level sensor network details by allowing them to express control system application requirements at the granularity of the entire network, and enabling them to focus on the novel requirements of their deployments.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38, 2002.

[2] D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Massachusetts, 1996.

[3] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, August 1988.

[4] A. E. Bryson and Y.-C. Ho. *Applied Optimal Control*. Blaisdell, New York, 1969.

[5] U. Chajewska and D. Koller. Utilities as random variables: Density estimation and structure discovery. In *UAI*, 2000.

[6] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, 2004.

[7] Amol Deshpande, Carlos Guestrin, and Samuel Madden. Resource-aware wireless sensor-actuator networks. 2005.

[8] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2001.

[9] L. Girod, T. Stathopoulos abd N. Ramanathan, J. Elson, D. Estrin, E. Osterweil, and T. Schoellhammer. A system for simulation, emulation, and deployment of heterogeneous sensor networks. In *SenSys*, 2004.

[10] D. Heckerman, D. Geiger, and M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft Research, Redmond, Washington, 1994.

[11] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, and David Cullerand Kristofer Pister. System architecture directions for networked sensors. In *Proceedings of ASPLOS*, November 2000.

[12] R. A. Howard and J. E. Matheson. Influence diagrams. In *Readings on the Principles and Applications of Decision Analysis*. 1984.

[13] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960.

[14] U. Lerner, B. Moses, M. Scott, S. McIlraith, and D. Koller. Monitoring a complex physical system using a hybrid dynamic bayes net. In *UAI*, 2002.

[15] M. Littman. *Algorithms for Sequential Decision Making*. PhD thesis, Brown University, 1996.

[16] S. Madden. *The Design and Evaluation of a Query Processing Architecture for Sensor Networks*. PhD thesis, UC Berkeley, 2003.

[17] A. Meliou, C. Guestrin, and J. Hellerstein. Routing queries in sensor networks. Tech. report, Intel Research, 2004.

[18] Tom Mitchell. *Machine Learning*. McGraw-Hill, Boston, Massachusetts, 1997.

[19] M. C. Mozer. Lessons from an adaptive house. *Smart environments: Technologies, protocols, and applications*, 2004.

[20] K. Ogata. *Modern Control Engineering*. Prentice Hall, Upper Saddle River, NJ, fourth edition, 2001.

[21] Greg Pottie and William Kaiser. Wireless integrated network sensors. *Comm. of the ACM*, 43(5), May 2000.

[22] C. Rios. WLANs and power over ethernet, 1999. `www.ieee802.org/3/power_study/public/july99/rios_1_0799.pdf`.

[23] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1994.

[24] R. H. Shumway and D. S. Stoffer. Dynamic linear models with switching. *Journal of the American Statistical Association*, 86(415):763–769, September 1991.

[25] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.

[26] R. Szewczyk, A. Mainwaring, J. Polastre, D. Culler. An analysis of a large scale habitat monitoring application. In *SenSys*, 2004.

[27] C. C. White and H. K. Eldeib. Markov decision processes with imprecise transition probabilities. *Operations Research*, 42(4):739–749, 1994.

[28] A. Woo, T. Tong, D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys*, 2003.

[29] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *SenSys*, 2003.

[30] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, 2004.