Making Sensor Networks Practical with Robots

Anthony LaMarca, David Koizumi, Matthew Lease, Stefan Sigurdsson and Gaetano Borriello; Intel Research
Waylon Brunette, Kevin Sikorski, Dieter Fox; University of Washington

IRS-TR-02-004

2002

Making Sensor Networks Practical with Robots*

Anthony LaMarca[1], Waylon Brunette[2], David Koizumi[1], Matthew Lease[1], Stefan B. Sigurdsson[1], Kevin Sikorski[2], Dieter Fox[2], Gaetano Borriello[1, 2]

[1]Intel Research Laboratory @ Seattle
[2]Department of Computer Science & Engineering, University of Washington

Abstract. While wireless sensor networks offer new capabilities, there are a number of issues that hinder their deployment in practice. We argue that robotics can solve or greatly reduce the impact of many of these issues. Our hypothesis has been tested in the context of an autonomous system to care for houseplants that we have deployed in our office environment. This paper describes what we believe is needed to make sensor networks practical, the role robots can play in accomplishing this, and the results we have obtained in developing our application.

1 Introduction

Wireless sensor networks offer new ways to monitor our environment and do so continuously and invisibly. These networks have wide applicability including medical, industrial, scientific, military, and consumer applications. Estrin et al. have described applications to automotive telematics, precision agriculture, and defense systems [9]. Rabaey et al. have considered management of environmental control systems in large office buildings [28]. Schwiebert et al. have been working to develop wireless biomedical sensors [31]. Byers and Nasser suggest using wireless sensor networks to monitor toxicity levels in hazardous areas [7]. Krishnamurthy and Conner have used sensor networks to implement basic office information services such as monitoring the use of highly-coveted conference rooms [8]. From this list, it is clear that wireless sensor networks can provide important data and context information for a very wide range of ubiquitous computing applications.

While their potential benefits are clear, a number of open problems must be solved in order for wireless sensor networks to become viable in practice. These problems include issues related to deployment, security, calibration, failure detection and power management. In the last decade, significant advances have been made in the field of service robotics [11], and robots have become increasingly more feasible in practical system design. Therefore, we suggest that a number of the problems with wireless sensor networks can be solved or diminished by including a mobile robot as an integral part of the system. Specifically, the robot can be used to deploy and calibrate sensors, detect and react to sensor failure, deliver power to sensors, and otherwise maintain the overall health of the wireless sensor network. The ideal is for the robot

---

* A version of this paper appears in the *2002 International Conference on Pervasive Computing.*

to do all this while only engaging the user as a last resort (e.g. when new sensors are needed to replace non-functioning ones).

While this application may be new, robots have been performing these types of services for some time. Service robots are the class of mobile, autonomous robots that operate in human environments to assist and serve. The applications that employ these service robots vary greatly with respect to the level of human interaction. The scale ranges from robots that perform their tasks independently, such as janitorial robots [10] through hospital aides [21], to robots whose chief function is to interact with people, such as entertainment robots [32], museum tour guides [6, 34, 25], and robots that aid the blind [22] and the elderly [29]. Recently Howard et al. used sensors placed in the environment to help mobile robots build navigation maps [18]. However, robots have not been applied to the problem of maintaining a distributed, ubiquitous computing system, such as a sensor network, and keeping such a system running and well-calibrated over an extended period of time.

We have tested our hypothesis in the context of the PlantCare project at the Intel Research laboratory in Seattle. PlantCare is part of an experimental proactive computing platform that serves as an autonomous system to care for houseplants. While caring for houseplants may not be the most promising use of wireless sensor networks, it is a well-defined, practical problem that both encompasses many of the open problems associated with wireless sensor networks and is at the same time conducive to the use of mobile robots.

The PlantCare system consists of a wireless sensor network to measure and report environmental conditions impacting the plants, application logic to monitor these conditions and determine appropriate responsive behavior, and a mobile robot to provide system actuation. Initial results from our system suggest that this combination of robots and wireless sensors has the potential to achieve unprecedented levels of system independence, including the ability to (1) sustain the energy resources of both the robot and the sensor network indefinitely, (2) automate sensor calibration, including both configuration and dynamic response to changing hardware behavior and/or environmental conditions, and (3) detect sensor failure and inappropriate deployment.

The organization of this paper is as follows: In Section 2, we discuss what we believe are three major issues hindering the adoption of wireless sensor networks in practice. We describe the problems, as well as how robots can be used to address them. In Section 3, we discuss the PlantCare system including the sensors, robotics and overall system architecture. In Section 4, we describe our results from a simple experiment with robotic sensor calibration. Finally in Section 5, we discuss future work and conclude.

2 Practical Issues in Wireless Sensor Networks

In the context of the PlantCare project, we have had to deal with a number of practical issues relating to our wireless sensor network: How should new sensors be deployed? How do we know that we are not placing a sensor in an anomalous location that will return unrepresentative readings? How should sensors be calibrated? Should they be calibrated in a controlled environment and then deployed, running the risk of inaccuracy? Or instead, should sensors be calibrated in their deployed location, which would be more accurate but also more difficult? Since we decided to deploy sensors without a constant supply of power, how will we keep the sensors running indefinitely? How will we know when a sensor has failed? While any one of these problems can potentially be ignored or circumvented, collectively they represent an important set of issues that affect the accuracy and robustness of our overall system. We have condensed these issues into three areas: context-aware deployment, continuous calibration, and power delivery.

2.1 Context Aware Deployment of Sensors

Before a sensor can provide useful data to the system it must be deployed in a location that is contextually appropriate. While the physical placement of a sensor is often challenging, as in the case of a reactor or a water main, the issue of proper choice of location based on the application requirements is also difficult and time consuming. For example, consider the placement of a thermostat in a home. Since in most cases only one thermostat is installed, this single sensor will be used to guide all heating and cooling decisions. Thus the sensor should be placed where its readings closely correlate with the temperature of the home in general. Consequently, the placement of the sensor under a cooling vent, or on a section of wall that gets intense sunlight in the morning, will generate contextually inappropriate readings. Even if the sensor is properly calibrated, it will return data that is misleading to the system, and the residents of the home will be uncomfortable. In this example, perfect placement requires an isotherm map of the space that shows the temperature variation by location as well as time. Since this is rarely available, most sensor placement is done using simple heuristics. In the case of home thermostats, for example, the rule of thumb is to place it near the center of the house in a hallway. While this example is fairly simple, and an isotherm map is unlikely to be worth the effort, it illustrates the tradeoffs: The accuracy and relevance of the data collected are proportional to the level of understanding of the environment. Unfortunately, considerable human time is usually required to obtain this understanding.

In many cases, this problem can be solved using robots, as has already been demonstrated in the limited context of active vision [23]. Placing an accurately calibrated sensor on the robot turns it into a mobile wireless sensor. This "robo-sensor" can then move through the environment to take measurements. If an accurate map of the environment is available to the robot for navigation, then environment maps such as the temperature isotherm map mentioned above are straightforward to build. The robot simply needs to visit a series of locations determined by the required measurement density, taking a reading at each after waiting long enough for the sensor to acclimate. In many systems this is a natural extension of mapping tasks already performed by robots. For example, in PlantCare our physical environment is mapped using probabilistic techniques [12] in order to guide robotic navigation (see Figure 1). Taking readings multiple times at varying times of day will create the type of map that can strongly suggest good sensor placement to an application administrator. If robotic placement of the sensor were desired, this same data could

Figure 1: This map of our office was generated by the robot's mapping system

guide an algorithm that captures the traditional heuristic wisdom. This is an example of how a robot can be used to accomplish a time-consuming, repetitive task that is likely to be unappealing to humans.

2.2 Continuous Calibration of Sensors

Calibration is the process of deriving the function that translates the raw readings provided by the sensor into data in the correct units, while taking into account the characteristics of the particular sensor and the environment in which it has been placed. Calibration of sensors is critical to obtaining accurate measurements. Without calibration, readings produced by a sensor could range anywhere from having subtle inaccuracies to being completely meaningless. Our temperature sensors, for example, produce a voltage varying from 0 to 3 volts. The only way to know that a reading of 1.5v corresponds to 32 degrees Celsius is to obtain a number of readings from the sensor when the true temperature is known and use these calibration points to translate the voltage readings to the Celsius scale. Unfortunately, this mapping varies from sensor to sensor (especially as manufacturers endeavor to make sensors as low-cost as possible) and may even change over the lifetime of a deployed sensor (due to changes in environmental conditions or wear on the sensor). The result is that for most applications, sensors require both initial calibration and periodic recalibration to ensure accurate readings.

Sensors can be calibrated either before or after deployment. Pre-deployment calibration is performed in a controlled space in which the environmental factor measured by the sensor can be carefully regulated. The factor is then cycled through a set of values intended to represent the range the sensor will encounter in practice, and readings are taken from the sensor during this process. These actual vs. sensed values form the set of calibration data that can be used to derive a function that maps one to the other. For example, linear regression is a simple yet accurate technique for performing this mapping. In the event that the sensor data cannot be mapped into linear space, more complex algorithms can be employed. The big advantage of pre-deployment calibration is that it can be inexpensive, as a large number of sensors can often be calibrated at the same time. The disadvantage is that sensors are often sensitive to environmental conditions (e.g., temperature) and the act of deployment can change their characteristics. The alternative is to calibrate the sensor in-place, after it has been installed. This is more time consuming and inconvenient. It also suffers from the problem that the environmental factors may change slowly and it may be difficult to collect sufficiently varied calibration data to get accurate measurements.

A robot equipped with an accurate, calibrated sensor offers a novel solution: adaptive in-place calibration. Given robotic support, a sensor can be deployed uncalibrated and a robot can be used to visit the sensor periodically to take calibration readings. As long as sensor readings remain in the range of these initial calibration readings, no additional visits are required. However, if environmental factors deviate outside the calibrated range, the robot can be dispatched to collect additional readings. For example, consider a wireless temperature sensor in an unheated warehouse. The sensor is deployed uncalibrated during the summer. Over a period of five days, the robot collects a set of calibration readings. These readings combined with simple regression provide sufficient accuracy for the next three months. When fall sets in, the temperature drops, and the system determines that the summer calibration data is not sufficient to predict the characteristics of the sensor. The robot is then re-deployed to collect new calibration samples to extend the range of the regression model. Again we see the robot filling a role that a human could but would never want to: namely, as an agent on call ready to visit any sensor whenever needed.

Recalibration is another area in which a robotic solution offers advantages over the standard technique. Currently, the most common method used to ensure sensors are operating correctly is periodic manual calibration [17]. This approach has a number of drawbacks. For instance, manual labor is expensive, so sensor recalibration is performed infrequently resulting in overall inaccuracy. Complementing our robotic solution, we assign a limited lifetime to calibration data as it is collected. Rather than a task performed once or twice a year, recalibration becomes an ever-present background task of the system. Once the lifetime of a piece of calibration data has expired, it is considered inaccurate and is replaced by fresh data. The result is an adaptive mechanism in which the tradeoff of sensor accuracy vs. robotic work can be adjusted at will.

The issue of failure detection is tied closely to calibration. Detecting a failed sensor is the pathological case of calibration when it is decided that the sensor readings no longer correlate with the environmental factor in question. This detection is fairly straightforward in the case of a gross fault, in which a sensor begins reporting drastically different conditions. Regular periodic calibration by our robotic solution also removes the traditional need to detect drift errors, in which a sensor's accuracy slowly decays over time [37]. Without sophisticated models, however, it is very difficult to detect when a faulty sensor returns plausible, possibly varying, yet uncorrelated data. Because sensors are only checked periodically, faulty sensors can remain in service for periods up to the calibration frequency. Depending on the scenario in which the sensors are used, this could pose severe consequences. To mitigate this, the calibration cycle can be made more frequent by using a smaller calibration data lifetime.

Inferential sensing is an emerging technique that continuously verifies sensor accuracy via on-line, real-time modeling [17]. This technique uses historical data rather than sensor redundancy [19] to construct predictive models of sensor behavior and can significantly enhance system fault detection and handling. Whenever a sensor provides a reading, the data is compared to estimated values produced by the model, generating differences known as residuals. A decision logic module then statistically evaluates each residual to generate a health metric assigned to the corresponding sensor. Instead of performing periodic recalibration, a human operator monitors these health metrics in order to schedule maintenance proactively or provide it when necessary. Unlike simpler methods, this approach can detect sensors that fail in non-trivial ways. In addition, these predictive models can be used to generate estimated sensor readings while a sensor is offline or awaiting recalibration or replacement. These predictive modeling techniques complement our robotic calibration approach. In the case of sensor failure, the predictive model can serve as warning system and the robot can verify the failure. When the predictive model suspects a sensor of drifting, it can prematurely age the sensor's calibration data, forcing a more timely robotic recalibration.

While robotic calibration solves many problems, it has a set of issues and considerations of its own. For accurate calibration data to be collected by the robot, the system must know precisely where the sensors are located. In the case of statically positioned sensors, this requires that a detailed sensor map be provided to the system. In the case of mobile sensors, a sophisticated localization system is required. Current localization systems use ultrasound [36], RF signal strength [2], radio time-of-flight [15], or some combination of technologies to pinpoint the physical location of objects in space. Second, the sensor needs to be accessible to the robot, and not all deployment environments allow this. In some situations the robot cannot reach the sensor (e.g. a sensor on the ceiling of a room), while other environments are physically hostile to robots. Finally, some sensors require physical integration with the environment in order to obtain accurate measurement, and as a result, corroboration cannot be achieved by placing a similar sensor nearby. For example, a stress sensor installed within a concrete wall cannot be verified by a mobile sensor no matter how close it gets.

2.3 Renewable Energy Provided by Mobile Service Robots

An obvious limitation of wireless sensor networks is the lack of a continuous energy supply. To make matters worse, the additional freedom of wireless networks allows one to envision deployment environments lacking any nearby infrastructure for supplying power. Given these constraints, it is generally accepted that the majority of wireless sensor networks will face a world of severe energy restrictions [27]. We review current approaches to dealing with the power problem and describe how robots both complement and extend these techniques.

The most popular approach assumes a sensor network is deployed with batteries that will not be recharged or replaced. In this scenario, power conservation is paramount and the network is considered disposable with the rate of energy consumption determining its operational lifetime. Many alternatives for conserving power have been considered [3, 7]. For example, Heidemann et al. have proposed application-specific network topologies in order to reduce costly communications in ad-hoc wireless sensor networks. [14]. Bhadwaj et al. have even built an abstract model of such wireless sensor networks in order to derive an upper bound on their lifetimes. [4]

For some applications, however, this model of wireless sensor networks is not appropriate: The sensor network cannot be treated as disposable and it is possible to sustain the sensors by recharging or replacing batteries when needed. For example, Schwiebert et al. are considering biomedical monitoring via wireless sensors implanted in the body. Because sensors are intended for long-term use, they have proposed using radio frequencies (RF) or infrared (IR) signals to inductively charge the implanted sensors from an external power source [31]. While recharging is well-suited to this application because the patient can wear a compact, portable power supply, not all applications can depend on this level of user interaction.

Some have taken this a step further to propose extracting energy directly from the deployment environment. These "scavenging" techniques propose powering sensors via solar power [35, 20], kinetic energy [26], floor vibration, and acoustic noise [28]. However, scavenging techniques are only now becoming capable of generating the level of power required to sustain current wireless sensor applications, and not all deployment environments are conducive to such techniques.

An alternative approach is to use passive sensors. Unlike the active sensors considered thus far, passive sensors require no local power source. For example, a surface-acoustic-wave (SAW) passive sensor is powered entirely by the RF field used to read it. A broad range of SAW sensors are available for measuring temperature, pressure, magnetic field strength, torque, etc. [13, 30]. In general, passive sensors offer the advantage of lower cost and higher robustness than active sensors but tend to require more expensive infrastructure. Also, because passive sensors depend on external power, they measure their environment only when polled; an event-driven application would be required to perform such polling regularly and generate appropriate events on behalf of the sensors.

As an alternative to these approaches, we propose using mobile service robots to sustain a deployed, active sensor network. Because we require our PlantCare system to run unassisted for long periods of time, we can neither treat the sensor network as disposable nor have it depend on an administrator who can perform recharging or battery replacement. Further, passive sensors do not yet exist to measure all of the environmental conditions pertinent to plant care, such as soil moisture content. On the other hand, Michaud et al. have already demonstrated that mobile service robots are capable of recharging themselves [24], thus it appears an appropriate task to have a robot of this kind deliver power to deployed active sensors as the need arises.

The power could be delivered in a variety of forms. A sufficiently agile robot could replace weak batteries in a sensor with fresh ones. An easier approach is to outfit the robot with equipment to recharge sensors using inductance or direct electrical connection. While this idea is simple to understand and straightforward to engineer, it has the potential to greatly increase the flexibility of wireless sensor networks. With a robot integrated in the system, wireless sensor nodes can be placed in locations in which no power is available at all. Infrequent visits by the robot enable the wireless node to perpetually participate in the application without any human intervention.

In addition to delivering power to active sensors, robots can potentially improve the efficacy of passive sensor networks as well. A mobile service robot equipped to perform inductive charging via RF could read a SAW sensor using the same equipment. This makes it easier to support a mixed environment of both active and passive sensors as well as reducing the infrastructure required to read a physically disparate collection of passive sensors.

3 PlantCare

We have been exploring the relationship between wireless sensor networks and robots in the context of the PlantCare project. The goal of the project is to understand and develop solutions to the challenges facing proactive computing. Proactive computing differs from pervasive computing in the sense that proactive applications have a component that anticipates needs and provides for dealing with them without the user's attention being called to the problem unless absolutely necessary, and then only at an appropriate level of abstraction. In general, the goal of proactive computing is to develop solutions to real problems that may involve hundred or thousands of devices, but present little distraction or cognitive load on their users. Specifically, PlantCare is trying to build a zero-configuration and distraction-free system for the automatic care of houseplants. Our plan was to instrument each plant with a wireless sensor placed in its pot and employ a robot to deliver water to the plants. Soon after conceptualizing the project we realized that due to power constraints, the sensor network and robot needed as much care as the plants themselves. This realization led to the idea of robots as a worthy caretaker for wireless sensor networks, in general. In order to provide background for the work in this paper, we present a brief description of the sensors, robots, and software employed by the PlantCare project.

3.1 The Sensors

In the PlantCare system, wireless sensor nodes (see Figure 2) are placed both on the robot and in the plants being cared for. The sensors in the plants provide a continuous stream of data reflecting their state while the sensor node on the robot is used to calibrate the sensors. While the sensors in the plants and on the robot vary slightly, the wireless nodes are identical. PlantCare's sensors are built using the UC Berkeley "mote" sensor platform running TinyOS [16]. Motes operate at 3V and are assembled from off-the-shelf components that include an 8-bit microcontroller, a two-way 916MHz radio for communication, and an expansion connector that facilitates connection of environmental sensors. TinyOS is a small, real-time, modular operating system that supports ad-hoc networking to allow motes to communicate both with each other and with a base station. Our environmental sensing hardware consists of a photo-resistor for measuring light levels, a thermistor for measuring temperature, an irrometer for measuring soil moisture content, and a sensor that monitors the current charge of the power source. In addition, the sensor nodes in our plants have been augmented with a custom power system in which capacitors replace traditional batteries and can be recharged using an inductive coil to support power delivery.

Our wireless network contains a single base station mote, which by virtue of being attached to the serial port of an Internet-connected PC serves as the physical link between the wireless sensor network and the PlantCare services. The base station listens to the sensor network for messages containing sensor readings and forwards these messages to the serial port. Additional software infrastructure described in Section 3.3 handles the processing of these messages on the PC.

3.2 The Robot

The robot hardware platform (see Figure 3) consists of a Pioneer 2-DX mobile robot [1] augmented with custom hardware for watering plants, recharging the robot, recharging remote sensors, and sensing environmental conditions for calibration purposes. To deliver water to the plants, the robot has been fitted with a small water tank, dispensing spout, and pump. To deliver power to wireless sensors an inductive charging coil has been positioned near the watering spout. Similarly, another paddle-shaped inductive charge coil has been added to the robot to allow it to recharge itself at its "maintenance bay". In order to support calibration, the robot includes a sensor node that was human-calibrated. Finally, a small microcontroller board allows software on the robot to both control and read the state of this collection of custom hardware. Both this microcontroller and the laser scanner the robot uses for navigation are connected to a laptop that runs the robot's control and navigation algorithms and is in turn connected to the network via an IEEE 802.11b wireless card.

Lastly, the robot has a maintenance bay it uses to automatically charge its own batteries and refill its water reservoir. The bay has a water supply with a spout for dispensing water to the robot, and a charging system matched to the robot's induction coil. We envision that the bay would take the form of a kitchen cabinet in a more consumer-realistic deployment.

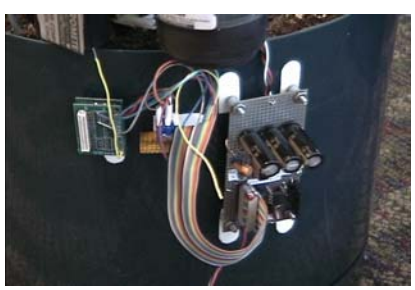

Figure 2: A deployed plant sensor

Figure 3: The robot hardware platform

We chose inductive charging for both the sensors and the robot in order to reduce the danger to people and equipment due to splashing water. The measured efficiency of inductively charging the sensors is around 70% of the baseline efficiency achieved with a shielded cable. This inefficiency reduces the amount of time the robot can function without recharging, thereby resulting in more frequent visits to the maintenance bay. This was deemed acceptable given the potential risk of the accidental meeting of water and electricity.

The main components of the robot navigation system consist of a reactive collision avoidance module, a module for map building and path planning, and a localization module. All components use probabilistic methods to deal with uncertain sensor information. The reliability of this approach has been demonstrated during the deployment of the robots Rhino and Minerva as autonomous museum tour guide robots [6, 34]. The high-level task ordering and dispatching software was custom-built for the PlantCare project.

3.3 The Software

As part of the PlantCare project we have also developed a software infrastructure called Rain to support proactive applications. The objective of Rain is to provide a framework in which to experiment with how to structure applications as a collection of cooperating services that communicate via asynchronous events. In the Rain environment, services register with a central discovery service and use this service to find other services they wish to interact with. This structure gives the application the opportunity to transparently support the highly dynamic environments envisioned for proactive systems. The asynchronous communication model allows applications to be highly responsive even in the face of widely distributed services running on hardware platforms with widely varying performance. Finally, to support heterogeneous computing environments, Rain messages are encoded in XML. Services in Rain are very similar to SOAP services that communicate by passing XML documents asynchronously [5]; The main difference between Rain and SOAP is that SOAP is as almost exclusively used as a synchronous RPC system and Rain is geared towards support of asynchronous event-based software architectures.

Our PlantCare application is composed of fifteen services that collectively provide both the high-level application logic as well as the low-level driver-like code that communicates with hardware and external software. Specific to our sensor network, there are services that independently receive data from the sensor base stations, unpack the data from its proprietary form, calibrate the data reading based on previously collected calibration data, and store the readings for future use by applications. The services pertaining to the robot consist of a low-level service that knows how to activate the robot's sensors and actuators, and a high-level service that encapsulates the understanding of our application-specific robotic tasks such as watering plants and delivering power to the motes.

4 A Simple Experiment

To demonstrate the use of a service robot for the calibration of a wireless sensor network we performed a simple experiment. The experiment emulated the lifetime of a plant sensor from initial deployment through continuous, adaptive calibration with the help of the robot, and was controlled to eliminate the effects of power constraints and navigation errors.

An uncalibrated sensor was deployed and calibrated in-place. After a period of operation we changed the environmental characteristics the sensor was measuring. The subsequently reported readings then no longer fell within the range of the previously collected calibration data, which forced the system to gather additional calibration data. This mimics the previously mentioned example of a temperature sensor that is calibrated during summertime conditions but needs further calibration at the onset of winter. Finally, we simulated a gross change in sensor behavior by physically obscuring the sensor. Once the resulting measurement error was detected by the system during a simulated maintenance check performed by the robot, the old calibration data was discarded and new calibration data was collected.

4.1 Experiment Setup

The experiment setup consists of a darkroom with a single constant intensity light source, and a single wireless sensor node equipped with a light sensor. The light source is passed through two polarizing filters, one of which is rotated at set intervals to make the intensity of the emitted light approximate a sine wave. For the first half of the experiment the rotation of the filter is limited between 0 and 45 degrees. This creates a wave that is cut off at half the possible amplitude. After two periods of this smaller wave the rotation of the filter is extended to the full range of 0 to 90 degrees to create a change in environmental conditions. This full range of light is projected onto the sensor for two full periods. The sensor is then covered by a semi-transparent filter to emulate degradation of the sensor. Finally, the full range of light is again projected onto the partially obscured sensor for two periods.

During the entire experiment, the wireless node is reporting its light readings to a base station via its radio. This data is stored in a database and post-processed using linear regression to convert these raw sensor readings into luminance values. When the algorithm needed a new calibration point, the raw sensor voltages were paired with the actual light intensity. This simulates a robot with a perfect sensor. While sensors are never perfect, in general we expect that the sensors on the robot will be significantly more accurate than those in an inexpensive wireless sensor node.

4.2 Experiment Results

Figure 4 presents three time-series graphs representing different aspects of the data collected during the experiment. In all of these graphs the vertical axis shows either volts or luminance, while the horizontal axis always shows time. In the first graph, the solid line shows the voltage values measured by the sensor during the experiment, compared with the broken line showing the actual luminance of the light source. Note that until the sensor is obscured, the voltage readings reach higher on the graph than the luminance values. Once the sensor is obscured, the voltages measured are lower though the luminance has not changed. It is clear that without recalibration, the sensor readings cannot be translated accurately across the entire time line.

The second graph in this series compares the calibrated sensor readings to the actual luminance. The markers on the broken line represent times at which the system collected calibration samples. Moving from left to right, we see that the system initially collected eight calibration points. Once the measured luminance starts to fall, no additional calibration points are collected, as the system is comfortable with the range of data it had already collected. Note also that the measured line accurately

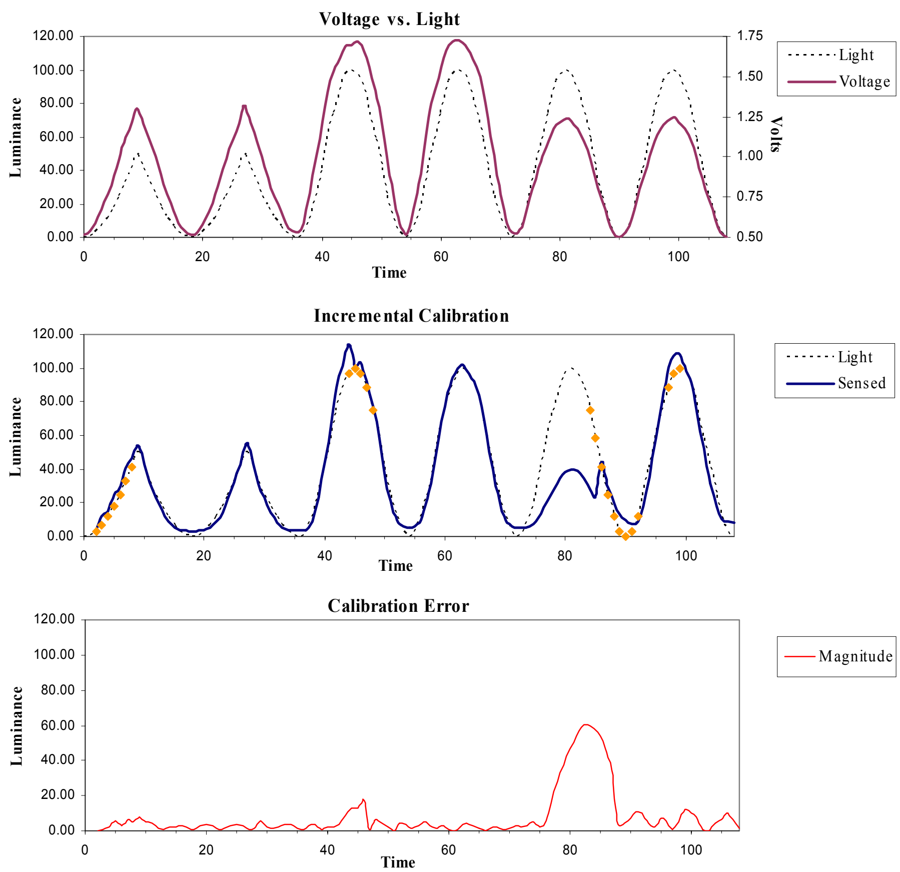

Figure 4: The top graph shows raw voltage vs. luminance over six periods of sinusoidal variation. The second graph compares the calibrated readings vs. the actual luminance for the same time span. The marks on the graph represent times at which calibration data was collected. The third graph shows the error in the calibrated readings

tracks the actual luminance except in the peaks and troughs of the graphs. We attribute these inaccuracies to limitations of our low-cost photo-resistor, pointing out one of the realities of using inexpensive sensors.

Midway through the third period, two things happen. The difference between the calibrated readings and the actual luminance increases. This is because the regression algorithm was being forced to make predictions outside its range of collected data. When the system realized this, it collected an additional four calibration points after a delay approximating the navigation latency of the robot. After incorporating the additional points, the measured values again track the actual luminance well.

Two thirds of the way through the experiment the sensor was covered with a filter to change its characteristics. Since the system was unaware of this, the converted measurements are in fact quite inaccurate. This could not be detected by the system and would have continued until a routine robotic check of the sensor revealed the inaccuracy. The delay of a few readings approximates the wait until the next check of the sensor. At this point, the system realized a significant change in the sensor had occurred, discarded all of its calibration data, and began collecting new data. After the recalibration period, the readings again track the actual luminance well.

Lastly, the third graph in this series shows the error between the calibrated sensor readings and the actual luminance. During the initial calibration period there is higher than average error. Another period of increased error occurs when the environmental conditions change and only settles down once additional calibration data is gathered. At the end of the measurement there is a very large spike of error while the system continues to use calibration data that has become inaccurate. Once the error is detected and new calibration data is gathered, the error again decreases. It is interesting to note that even after recalibration the error does not diminish to the extent it did following initial calibration. This is due to the fact that obscuring the sensor rendered it less sensitive to light. While our goal was to change the scale of the voltages returned, we also inadvertently compromised its accuracy.

5 Conclusions and Future Work

In this paper, we have introduced the idea that robots have the potential to greatly increase the feasibility of practical wireless sensor networks. While sensor networks and robotics are both quickly evolving fields, the union of the two fields seems inherently symbiotic. Sensor networks have data but lack actuation, while robots have actuation but limited sensing. We have explored this concept in the context of the PlantCare system, an autonomous system for managing the health of houseplants. We have presented data from a simple initial experiment showing how robots can be used to continuously calibrate deployed sensors. In the future, we intend to more deeply explore the relationship between robots and sensor networks. We plan to leverage techniques developed in the robotics community to build spatial models from noisy sensor information and to keep track of complex dynamic systems [33]. We also plan to explore the idea of treating localization data as just another aspect of the sensed environment, enabling localization to benefit from all of the advantages of continuous calibration.

References

1. ActivMedia Robotics, http://www.activrobots.com, visited Feb. 2002.
2. P. Bahl and V. Padmanabhan. *RADAR: An in-building RF-based user location and tracking system. In Proceedings of IEEE INFOCOM*, volume 2, pages 775-784, March 2000.
3. P. Bhagwat et al. System Design Issues for Low-Power, Low-Cost Short Range Wireless Networking. In IEEE International Conference on Personal Wireless Communications, 1999, pp. 264-268.
4. M. Bhardwaj, A. Chandrakasan, and T. Garnett. Upper Bounds on the Lifetime of Sensor Networks. IEEE International Conference on Communications, 2001, vol.3 pp. 785-790.
5. D. Box et al. Simple Object Access Protocol (SOAP) 1.1, World Wide Web Consortium (W3C), May 2000. http://www.w3.org/TR/2000/NOTE-SOAP-20000508, visited Feb. 2002.
6. Burgard, W., A. Cremers, D. Fox, D. Haehnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun. 1999. Experiences with an interactive museum tour-guide robot. Artificial Intelligence.
7. J. Byers and G. Nasser. Utility-Based Decision-Making in Wireless Sensor Networks. In IEEE MobiHoc, 2000.
8. W. S. Conner, L. Krishnamurthy, and R. Want, Making Everyday Life Easier Using Dense Sensor Networks. In Proceedings of the Third International Conference on Ubiquitous Computing (Ubicomp 2001), 49-55. 2001.
9. Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers. Computer Science and Telecommunications Board (CSTB) Report.
10. Endres, H., W. Feiten, and G. Lawitzky. 1998. Field test of a navigation system: Autonomous cleaning in supermarkets. In Proc. of the 1998 IEEE International Conference on Robotics & Automation (ICRA 98).
11. Engelberger, G. 1999. Services. In Nof, S. Y., ed., Handbook of Industrial Robotics. John Wiley and Sons, 2nd edition. Chapter 64, 1201-1212.
12. D. Fox, S. Thrun, F. Dellaert, and W. Burgard. *Particle filters for mobile robot localization*. Springer-Verlag, New York, 2001.
13. M. Hauser, L. Kraus, and P. Ripka, *Giant magnetoimpedance sensors*. IEEE Instrumentation & Measurement Magazine, 2001, Volume 4 Issue 2, pp. 28-32.
14. J. Hiedemann et al. Building Efficient Wireless Sensor Networks with Low-Level Naming, SOSP, 2001.
15. J. Hightower, R Want, and G Borriello. SpotON: An indoor 3d location sensing technology based on RF signal strength. UW-CSE Tech Report 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, February 2000.
16. J. Hill, R. Szewcyk, A. Woo, D. Culler, S. Hollar, K. Pister. 2000. System Architecture Directions for Networked Sensors. Architectural Support for Programming Languages and Operating Systems 2000.
17. J. W. Hines, A. Gribok, and B. Rasmussen. On-Line Sensor Calibration Verification: "A Survey". 14th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management, September 2001.
18. A. Howard, M.J. Mataric, and G.S. Sukhatme. Relaxation on a mesh: A formalism for generalized localization. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, 2001.
19. C. Jaikaeo, C. Srisathapornphat, and C.-C. Shen. Communications, 2001. ICC 2001. IEEE International Conference on , Volume: 5 , 2001, pp. 1627 -1632.

20. J. M. Kahn, R. H. Katz and K. S. J. Pister. Next Century Challenges: Mobile Networking for "Smart Dust". MobiCom, 1999, pp. 271-278.
21. S. King, and C. Weiman. 1990. Helpmate autonomous mobile robot navigation system. In Proceedings of the SPIE Conference on Mobile Robots. 190-198. Volume 2352.
22. G. Lacey, and K. Dawson-Howe. 1998. The application of robotics to a mobility aid for the elderly blind. Robotics and Autonomous Systems 23:245-252.
23. P. Lehel, E. Hemayed, and A. Farag. Sensor planning for a trinocular active vision system. IEEE Computer Society Conference on Computer Vision and Pattern Recognition , 1999 -312 Vol. 2.
24. F. Michaud et al. Experiences with with an autonomous robot attending AAAI. IEEE Intelligent Systems, 2001, Volume: 16 Issue: 5, pp. 23-29.
25. I. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto. An affective mobile educator with a full-time job. *Artificial Intelligence* , 114(1-2):95--124, 1999.
26. J. Paradiso and M. Feldmeier, A Compact, Wireless, Self-Powered Pushbutton Controller. In Proceedings of the Third International Conference on Ubiquitous Computing (Ubicomp 2001), 299-304. 2001.
27. G. Pottie. Wireless SensorNetworks In Information Theory Workshop, 1998, pp. 139-140.
28. J. Rabaey and et al. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. IEEE Computer, July 2000, Vol. 33, No. 7, pp. 42-48.
29. Roy, N., G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, J. Schulte and S. Thrun. 2000. Towards personal service robots for the elderly. In Workshop on Interactive Robots and Entertainment (WIRE 2000).
30. G. Schimetta, F. Dollinger, and R. Weigel. A wireless pressure-measurement system using a SAW hybrid sensor. IEEE Transactions on Microwave Theory and Techniques, 2000, Volume 48 Issue 12 , pp. 2730-2735.
31. L. Schwiebert et al. Research Challenges in wireless networks of biomedical sensors. In MobiCom, 2001.
32. Sony AIBO homepage. http://www.world.sony.com/Electronics/aibo. Visited in February 2002.
33. S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA 2000)* , 2000.
34. S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz. 1999. MINERVA: A second generation mobile tour-guide robot. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).
35. E. Tunstel, R. Welch, and B. Wilcox. Embedded Control of A Miniature Science Rover For Planetary Exploration. 7th International. Symposium. on Robotics with Applications, 1998.
36. Ward A. Ward, A. Jones, A. Hopper. A New Location Technique for the Active Office. IEEE Personal Communications, Vol. 4, No. 5, October 1997, pp. 42-47.
37. Xu *et al*. On-Line Sensor Calibration Monitoring And Fault Detection For Chemical Processes. *Maintenance and Reliability Conference (MARCON 98)*, Knoxville, TN, May 12-14, 1998.