Market-Based Resource Allocation in a Wirelessly Integrated Naval Engineering Plant
Office of Naval Research (ONR) Contract Number: N00014-05-1-0596

FINAL REPORT

Jerome P. Lynch, Ph.D., University of Michigan (PI)
Andrew T. Zimmerman, Ph.D., University of Michigan
Raymond A. Swartz, Ph.D., University of Michigan

Date of Report:
December 1, 2009

Point of Contact:
Jerome P. Lynch, Ph.D.
Assistant Professor
Department of Civil and Environmental Engineering
Department of Electrical Engineering and Computer Science
2380 G. G. Brown Building
Ann Arbor, MI 48109-2125
(734) 615-5290
jerlynch @umich.edu

DTIC® DEFENSE TECHNICAL INFORMATION CENTER

Information for the Defense Community

DTIC® has determined on 3 / 19 / 12 that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

☒ **DISTRIBUTION STATEMENT A.** Approved for public release; distribution is unlimited.

☐ **© COPYRIGHTED.** U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

☐ **DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT C.** Distribution authorized to U.S. Government Agencies and their contractors (fill in reason) (date determination). Other requests for this document shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT E.** Distribution authorized to DoD Components only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT F.** Further dissemination only as directed by (insert controlling DoD office) (date of determination) or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

☐ **DISTRIBUTION STATEMENT X.** Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25; (date of determination). DoD Controlling Office is (insert controlling DoD office).

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 12-01-2009	**2. REPORT TYPE** Final Technical Report	**3. DATES COVERED** *(From - To)* June 1, 2005 - December 31, 2008

4. TITLE AND SUBTITLE

Market-Based Resource Allocation in a Wirelessly Integrated Naval Engineering Plant

5a. CONTRACT NUMBER

N00014-05-1-0596

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jerome P. Lynch, Ph.D.
Andrew Zimmerman, Ph.D.
Raymond A. Swartz, Ph.D.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Michigan
2380 G G Brown Building
Ann Arbor, MI 48109-2125

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Attn: Anthony J. Seman III
875 North Randolph Street
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)

ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

20120305030

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The focus of this research project is the application of wireless sensor networks in naval plant systems for monitoring and control purposes. During the first quarter of the project, research efforts have been largely focused upon the design and construction of a wireless sensor prototype explicitly designed for embedment in the complex environments posed by navy vessels. The role of the wireless sensor is to: 1) collect real-time information from sensors, 2) wirelessly communicate plant data to other sensors and/or upward in the integrated engineering plant (IEP) architecture, 3) automatically calculate plant actions, and 4) issue commands to actuators. These functional requirements are considered and appropriate off-the-shelf components selected for the wireless sensor design. To integrate all of the selected hardware components, a four-layer printed circuit board with a small footprint is designed and fabricated. In parallel to the hardware design of the wireless sensor, software is concurrently written to achieve reliable wireless communication between sensors. The IEEE802.15.4 wireless protocol standard is adopted and current efforts are coding it in C.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*

Table of Contents

Executive Summary:

The United States Navy has recently embraced the revolutionary electric warship concept for the design of future vessels. A core element of this concept is the integrated engineered plant (IEP). IEP removes traditional system-level barriers between the distinct ship plants (electrical, propulsion, cooling, ventilation) so they can leverage a shared set of resources and information management systems. The benefits of the IEP framework are naval vessels with reduced costs, smaller crews, enhanced efficiency and improved survivability. As part of the IEP framework, plant components are becoming empowered with computing capabilities for self-control and management. As a result, the IEP is a complex and highly decentralized system that is still responsible for achieving real-time control objectives. To achieve optimal control within the challenging environment that the IEP's architecture poses, an integrated hardware-software framework is proposed. At the core of the proposal is the design of a modular wireless node prototype. The wireless node integrates wireless radios and embedded computing technologies with existing plant sensors and actuators. The units, termed Narada, are low-cost (costing less than $200 each), compact, hardened, and are capable of collecting sensor data and issuing actuation commands.

Wireless sensor/actuator networks forego the high data transfer rates associated with cabled sensors in exchange for low-cost and low-power communication between a large number of sensing devices, each of which features embedded data processing capabilities. As such, a new paradigm in large-scale data processing has emerged; one where communication bandwidth is somewhat limited but distributed data processing centers are abundant. By taking advantage of this grid of computational resources, data processing tasks once performed independently by a central processing unit can now be parallelized, automated, and carried out within a wireless sensor network. By utilizing the intelligent organization and self-healing properties of many wireless networks, an extremely scalable multi-processor computational framework can be developed to perform advanced engineering analyses. In this project, a novel parallelization of the simulated annealing stochastic search algorithm is presented and used to update models of ship plants by comparing model predictions to experimental results. The resulting distributed model updating algorithm is capable of being used as a computational tool to perform an assessment of the condition of a plant. Should damage occur to a plant, it can be identified in a timely fashion and reported to a supervisory element in the plant that can reconfigure the system to ensure the plant can continue to meet its mission objective. The project focused on the monitoring and condition assessment of a chilled water plant. Using a demonstrator located at the Naval Surface Warefare Center (NSWC) – Philadelphia, the Narada network proves effective in identifying rupture conditions in the plant.

A total of 24 high quality publications were published based on the work conducted in this project. In addition, a variety of technology transfer activities were undertaken including a long-term deployment of the wireless nodes developed on this project to monitor the hull of the FSF-1 SeaFighter, a high-speed aluminum littoral combat ship. During summer 2009, a 31-channel wireless sensor network was installed to record hull strains and accelerations with the ship monitored as it travelled from Panama City to Portland, Oregon.

1. Introduction:

To maintain its role as the preeminent global naval power, the United States Navy invests heavily in fundamental research aimed at modernizing its fleet. As part of these efforts, the Navy has begun to explore revolutionary ship design concepts intended to enhance mission functionality, improve damage tolerance, and reduce manning requirements. In recent years, the electric warship design paradigm has emerged as one such revolutionary concept (ONR 2003). The electronic naval warship seeks to employ excess power generated in the shipboard propulsion system to power auxiliary ship systems including power-pulsed weapons, high energy embedded sensors, among other systems. Benefits associated with Integrated Power Systems (IPS) include reduced cost, enhanced naval design flexibility, and increased mission efficiency. In tandem with IPS, the Navy has also pursued enhancement (namely improved ship reliability and survivability) of future naval vessels through the Integrated Engineering Plant (IEP) architecture.

1.1. Integrated Engineering Plant Concept:

The IEP architecture seeks further integration of all ship plants (*e.g.,* propulsion, cooling, ventilation, fire suppression, electrical) so that they share common resources and information management systems (Dunnington, *et al.* 2003). IEP capitalizes on the continuing distribution of computing elements throughout the ship. As microprocessor electronics become increasingly capable (from a computing standpoint) and inexpensive, computing resources are moving away from centralized computing systems (as was typical of older naval plant designs) and moving towards integration with the sensor and actuators embedded within the distinct shipboard plants. The trend of integrating computing resources with component-level sensor systems allows the IEP architecture to adopt automated decision making in a decentralized fashion.

At the core of IEP is a three tiered decision and control architecture divided between the component, process and mission control layers, as shown in Figure 1.1 (Dunnington, *et al.* 2003). The component layer consists of the various machines and modules in which sensors and actuators reside. Some computing technology resides within the component layer to carry out tasks associated with local closed-loop control and component-level health monitoring. The process layer resides above the component layer and contains greater computing resources. Given the responsibility of optimizing the performance of the different ship objectives, the process layer continuously monitors the component layer to be aware of the need to reconfigure ship's resources to better attain the ship's mission. The highest layer of the three tier hierarchy is the mission control layer. The mission control layer manages the global performance of the ship by making major resource allocations when the operating mode and physical condition of the ship changes. At this layer, the IEP system provides ship personnel with critical real-time information collected from the lower component and process layers.

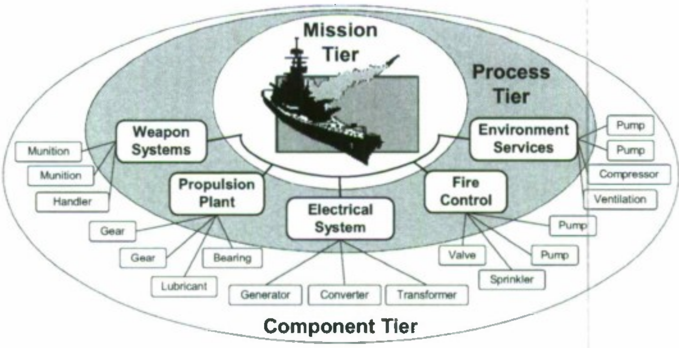


Figure 1.1: Integrated engineering plant (IEP) concept.

The mission efficiency of the ship is drastically improved because of the shared and interoperable management system provided by the IEP framework. For example, as sensor and plant systems continue grow in complexity and become increasingly interdependent, automated decision making can alleviate the problem of inundating the ship crew with too much information to be effective in their respective missions. Hence, automated decision making can lead to: 1) substantial reductions in the size of ship crews, 2) improvement of the mission responsiveness of the ship, and 3) rapid reallocation of plant resourced for ship survivability. A highly decentralized system also derives benefits by avoidance of single-point-of-failure problems common in centralized control system architectures. Some of the other benefits offered by the IEP framework include system scalability, cost-effectiveness, health monitoring of shipboard equipment and increased payload fractions.

To render the IEP framework feasible, a reliable ship-wide area network (SWAN) is needed to provide a communication infrastructure within and between the three IEP layers. The required SWAN must be immune to physical damage that is possible during battle. Current ship designs call for triple redundant serial buses for communication between components. In contrast to this approach, various military vendors have proposed different communication architectures that enjoy better fault tolerance. For example, Adept Systems has proposed a complex hybrid communication architecture comprising of multi-point ring networks and a LonTalk communication protocol (ANSI 709.1) (Smith 1994). This approach is scalable, cost-effective and enjoys self-healing capabilities as recently illustrated on the ONR Afloat Laboratory test-bed ship and the ex-USS Peterson (Adept 2004). Highly reliable network architectures are also being explored for the distributed power systems of future naval vessels. For example, ring topology load buses sharing strong similarity to the multi-point ring communication networks proposed by Adept, are proposed as part of the reconfigurable electrical systems (RES) program (Dunnington, *et al.* 2003).

1.2. Wireless Communications in Naval Ships:

As the United States Navy progresses towards the Integrated Engineering Plant design concept, the extensive length of wires that are needed to connect sensors and actuators with control units distributed throughout a vessel remains a technological challenge. In addition to the high cost of routing wires during construction, naval vessels represent a complex and harsh environment in which extensive lengths of wires are vulnerable to detriments such as heat, moisture and toxic agents (MacGillivray and Goddard 1997). With wires vulnerable to failure when exposed to these harsh conditions, reduction or outright elimination of communication wires would greatly enhance the reliability of on-board engineering control systems in addition to reducing installation and maintenance costs.

Ships pose a challenging setting for the propagation of wireless signals. Estes, *et al.* (2001) has explored the feasibility of wireless radios for both intra- and inter-compartment shipboard communications within various naval vessels (ex-USS America, USS Ross, and USS Carr). Their study considered radio frequencies between 800 MHz and 2.5 GHz which are typical radio frequencies for commercial off-the-shelf (COTS) radios. Based on the high radio frequency (RF) reflectivity of steel, multi-path effects were discovered to dominate received radio signals during inter-compartment wireless communication. To overcome multi-path influences, only frequency hopping spread spectrum (FHSS) wireless radios were discovered to effectively work. When assessing the feasibility of inter-compartment communication, ship bulkheads severely attenuated wireless signals (on the order of magnitude of 20-30 dB) but communication through two or three bulkheads was still found to be possible. Steel is a near perfect conductor that reflects electromagnetic waves thereby limiting radio signal penetration. However, on modern ships, a number of non-steel elements are present in the bulkheads (e.g. hatch seals, ducts, cable transits) that allow wireless signals to penetrate. Mokole, *et al.* (2000) has undertaken a similar wireless communication feasibility study on the ex-USS Shadwell using COTS wireless modems that communicate on the 800 MHz to 3 GHz radio frequencies. Their study found that radio communication is possible for inter-compartment communication using commercial wireless radios, even when bulkhead closures were fastened.

In 1999, the Office of Naval Research (ONR) initiated the Reduced Ship's Crew by Virtual Presence (RSVP) program with the aim of developing a proof-of-concept wireless monitoring system for shipboard sensor systems (Schwartz 2002; Seman, *et al.* 2003). Wireless communication between embedded sensors and existing shipboard local area networks (LANs) was aimed to reduce ship construction and maintenance costs in addition to enhancing the reliability and survivability of ships. The complete RSVP framework proposed dedicated sensor networks for monitoring the environmental parameters of ship spaces, the structural integrity of the hull, the health of machinery, and the real-time location of ship personnel. The framework sought to minimize the power consumption of sensing devices to ensure long operational lives. The machinery, environmental, structural and personnel wireless sensor networks had redundant access to various access points through 802.11 wireless communications. Data from the distributed wireless sensors are communicated to a dedicated watch station through a highly reliable wire-based shipboard LAN. The RSVP program validated different components of the RSVP architecture on numerous naval vessels including the USS Monterey and ex-USS Shadwell.

Other researchers have successfully explored the use of wireless communications integrated within various shipboard engineering plant systems to monitor component health. Ploeger, *et al.* (2003) describes a cost-effective wireless monitoring system that monitors the operational health of a shipboard ventilation system. The wireless system proposed is constructed from multiple wireless data acquisition nodes, termed the Intelligent Component Health Monitor (ICHM), that are capable of collecting sensor data from numerous analog sensors and communicating that data via Bluetooth wireless radios to a centralized data repository termed the Compartment Health Monitor (CHM). The system described is well suited for intra-compartment communication because of the short 20-30 m communication range associated with Bluetooth radios. Simple data processing of sensor data, including threshold detections and spectrum analysis, is performed at the compartment's CHM server. As validation, ventilation fans upon an operational aircraft carrier were monitored for overall health using the described system, resulting in an estimated shipboard crew labor reduction of 63% (reduction of 5000-8000 man hours per year per aircraft carrier) (Ploeger, *et al.* 2003).

Many other engineering domains have begun to explore the use of wireless sensors for reducing the high costs and technical difficulties often associated with wire-based data acquisition systems. For the past decade, academic and commercial research teams have proposed a variety of low-cost wireless sensor platforms. The most notable examples include the Smart Dust (University of California-Berkeley) and μAMPS (MIT) projects that have yielded low-cost wireless sensor nodes designed for deployment in wireless sensor networks defined by high nodal densities (Horton, *et al.* 2002; Min, *et al.* 2001). However, these generic wireless sensing systems, in their current state of development, do not address the unique demands of the harsh monitoring domain posed by a complex naval vessel, where low-power consumption, far-reaching communication ranges, penetration characteristics through ship materials and sufficient computational capabilities for autonomous data processing are all needed.

1.3. Decentralized Large-Scale Control and Artificial Intelligent Agents:

As industrial systems grow in size and complexity, the ability to control them in centralized fashion becomes increasingly difficult. In particular, a number of factors contribute to the breakdown of the centralized control system architecture. First, as the number of sensors and actuators linearly increase, the computational demands (aggregation of sensor data, calculation of control forces, and commanding actuators) placed upon a centralized controller grow exponentially (Lunze 1992). Furthermore, large plants are often complex systems that are difficult to precisely model using classical mathematical approaches. In response to these limitations, the field of control engineering has extensively explored the decentralization of the control system. By dividing a complex global system into simpler sub-systems, optimal closed-loop control can still be achieved by decentralized control (Siljak 1991).

Other researchers have pursued the use of agents for performing control of large-scale complex systems. Agents are software abstractions that are capable of perceiving their environment (through sensing) and acting upon that perception (through actuation) based on a rational decision process (through embedded intelligence) (Jennings, *et al.* 1997). As part of the rational decision process, the agent seeks to maximize a performance measure that represents the return on investment for a given action. Agents have found use in a number of complex systems including control systems, networks, software systems, manufacturing processes, among many others (Wooldridge and Jennings 1995). Autonomous intelligent agents, when assembled into a community, are often referred to as multi-agent systems (MAS). MAS are particularly attractive for performing control of complex systems because they represent a scalable problem solving approach for problems with incomplete information and decentralized data. For control problems where a centralized control solution is not feasible (due to the inability to exactly represent the system with mathematical models), multi-agent systems have proven to be reasonable approaches to performing feedback control.

The IEP framework proposed for future naval ships presents many technical challenges that renders centralized control an impractical approach. Alternatively, the use of multi-agent systems have been proposed as a means of controlling the IEP in a more effective and reliable manner. With high degrees of autonomy and the ability to rapidly adapt, intelligent agents will undoubtedly play a major role in the IEP framework. Agents provide a reconfigurable framework for plant control that can adapt with changes in the ship mission or ship operational conditions (an undamaged versus damaged ship). The agent framework proposed for the IEP is possible largely because computing power is migrating from upper tiers of the IEP framework down to the component level. Already, researchers are validating the feasibility of MAS for performing adaptive control within the IEP. Sun and Cartes (2004) propose the use of software agents for performing closed-loop control shipboard distributed power plant. Maturana, *et al.* (2004) describes the implementation of intelligent agents for controlling the flow of water within a chilled water system. Their study has implemented the agents using legacy control modules with abstracted Java-based software agents; validation of their multi-agent system is performed upon a small-scale chilled water plant.

2. Technical Objectives:

This project explores the use of wireless sensor and actuator networks within naval vessels as a means of reducing crew requirements and to enhance ship fight-through capabilities. Wireless sensors and wireless actuators are not sensors or actuators *per se*, but rather autonomous nodes of a data acquisition and feedback control to which sensors and actuators could be attached. The advantages of using wireless sensors for this purpose are to: 1) reduce the installation and maintenance costs associated with current tethered control systems, 2) to allow for ad-hoc network topologies between sensors and actuators offering communication robustness during battle, and 3) to collocate computing intelligence with plant components for plant assessment, control and reconfiguration. This represents a highly revolutionary approach to the implementation of the integrated engineering plant framework.

2.1. Objective 1: Low-Cost, Computationally-Rich Wireless Node Development:

The first technical objective of the project is the design of a low-cost wireless node designed explicitly for embedment in shipboard plant systems. Unlike generic commercial wireless sensors, the wireless node to be developed as part of this project will be designed to achieve functionality associated with the monitoring, control and reconfiguration of shipboard plants. Towards this end, the node will be designed to perform the following tasks: 1) collection of state data from sensors monitoring plant performance; 2) real-time wireless eommunieation of data to the upper ship mission tiers; 3) embedded data interrogation for intelligent in-network data proeessing; 4) ad-hoe eommunieation between wireless sensors to transfer real-time state data; 5) eollective deeision making between wireless sensor nodes; 5) automated aetuation of plant elements at the eomponent tier. In addition to these features, the wireless sensors must be low-eost (<$200) and be hardened to ensure they survive during battle and when exposed to water.

2.2. Objective 2: Embedded Firmware for Wireless Node Autonomy:

Onee the wireless sensors are designed and eonstrueted, the seeond project objeetive if foeused upon their autonomous operation. To automate their operation, embedded software is needed for eaeh wireless sensor. Embedded software, termed firmware, will automate the eollection of state data from sensors, wirelessly eommunieate data with other wireless sensors and to issue eommand signals to plant aetuators. A eore element of the firmware is the eoding of a flexible wireless eommunieation protoeol. Provided the low-power nature of the IEEE802.15.4 eommunieation standard, the IEEE802.15.4 physieal and medium aeeess eontrol protoeol layers will be written in C and eneoded in the wireless sensor eores. With a flexible eommunieation standard, the wireless sensor network will be able to form ad-hoe network topologies between plant eomponents.

2.3. Objective 3: Distributed Computing for State Awareness (Damage Detection):

The third objeetive of the project is to take full advantage of the embedded intelligenee eoupled with each wireless sensor through formulation of agent-behavioral models. Borrowing eoneepts from the field of artifieial intelligenee (AI), eaeh wireless sensor is modeled as an autonomous agent eapable of interaetion with other wireless sensors. The agent framework is espeeially powerful for aehieving deeentralized eontrol in a network of wireless sensors. Computationally, the projeet foeus will be on the development of eomputational tools that ean perform model updating. Using an analytieal model that deseribes the eonfiguration and operation of a plant system, model updating ean be performed to realize the best fit of the analytieal model to experimental data eolleets. If a plant experienee damage, this state will be identified by the model updating method. With model updating inherently an intraetable eombinatorial optimization problem, stoehastie methods that offer high probabilities of eonvergenee to global minima will be adopted. Furthermore, to benefit from a dense network of wireless sensors and aetuators installed in a ship plant, a means of elegantly distributing eomputational tasks to available wireless nodes will be developed. Using a multi-agent approaeh based on free market eeonomies (termed market based eontrol) will be explored for solving eomplex eomputational resouree alloeation problems eneountered in distributed in-network data proeessing.

2.4. Objective 4: Experimental Validation:

The final objeetive is to perform various validation experiments of the integrated hardware/software framework proposed. Opportunities will be sought to validate the wirelessly integrated plant on testbed platforms that exist at NSWC Carderoek Division - Philadelphia. Specifically, a scaled down chilled water plant demonstrator will be utilized.

3. Design of a Low-Cost Wireless Sensor/Actuator Node for Ship Automation:

3.1. Design of a Wireless Node:

While wireless sensor solutions are eommereially available, these solutions are too generie for use in naval plants. A more effeetive approaeh is to design a wireless node whose performanee has been optimized for operation in a naval vessel. A wireless sensor will be designed for real-time eontrol of naval plants using eommereial off-the-shelf (COTS) components. The wireless sensor design will be broken down to four funetional subsystems: sensing interfaee, aetuation interfaee, eomputational eore and wireless communication channel. The sensing interface is responsible for data collection from plant sensors, the computational core is intended to aggregate data and locally process it, the actuation interface will be used to command plant actuators, and the wireless channel will establish connectivity between sensors, actuators and the ship information management system. An architectural schematic of the proposed wireless node, termed Narada, is shown in Figure 3.1 and Figure 3.2.

3.1.1. Sensing Interface:

The sensing interface is the wireless node component to which plant sensors are attached. The wireless node should be able to accommodate multiple sensors at one time. Furthermore, the interface should be independent of the sensor type to ensure versatility in deployment in naval systems. The primary electrical component selected for the sensing interface is a four-channel, 16-bit, 100 kHz analog-to-digital converter (ADC); in particular, the Texas Instruments ADS8341 is selected. The interface is capable of accepting analog sensor signals that are between 0 and 5 V.

3.1.2. Computational Core:

The wireless node has at its core a low voltage microcontroller which is used to operate all of the electrical components, in addition to performing data interrogation tasks. The microcontroller selected is the 8-bit Atmel ATmega128 microcontroller. The Atmega128 provides 128 kB of flash memory which is adequate for most embedded software algorithms required for monitoring and control applications. To augment the built-in 4kB of SRAM, external memory of 128kB is provided using a Cypress CY62128B SRAM chip. This is adequate for data storage and analysis requirements of the sensing unit. The ATmega128 is operated at 8 MHz which consumes less than 100 mW of power from portable power sources (*e.g.,* batteries). Power is a consideration because it is anticipated the wireless nodes might be required to run on batteries during some battle scenarios.

3.1.3. Wireless Communication Channel:

A low-power, single-chip, IEEE 802.15.4 compliant wireless radio is adopted in the wireless node design. The newly created IEEE 802.15.4 standard is especially suitable for distributed computation, as it was developed for true ad-hoc peer-to-peer networking among battery powered wireless devices. The Chipcon CC2420 wireless transceiver is selected. The CC2420 is a single chip IEEE 802.15.4 compliant radio capable of ranges adequate for intra-compartment communication in ships. For example, its operational communication range is validated up to 20 m. Furthermore, the radio operates on the 2.4 GHz radio band with data rates as high as 250kpbs. Because the CC2420 is designed specifically to be IEEE 802.15.4 compliant, it does not require the use of a central coordinator to facilitate communication; the sensors may communicate directly with each other making them ideal for peer-to-peer communications.

3.1.4. Actuation Interface:

The Texas Instruments DAC7612 has been integrated in the design of the wireless node to service actuators. The DAC allows two actuation channels to operate simultaneously. The output voltage range of the DAC is 0 to 4 V.

3.2. Assessment of Data Collection Quality :

With a set of wireless nodes fully assembled, various diagnostic tests were conducted to validate assumptions made during the design process of the node. Specifically, the ADC resolution was measured using a constant-voltage battery source. The sensing interface resolution is an important property that describes the data collection accuracy of the wireless sensor. For example, a high-resolution ADC (16-bits or higher) suggests the wireless sensor is capable of reading sensors with low amplitude outputs. In contrast, a low-resolution ADC (less than 16-bits) represents a cruder data collection tool where the quantization process of the ADC ignores small varying sensor outputs. Particularly for naval plant systems, sensors defined by high accuracy must be complemented by a high resolution ADC.

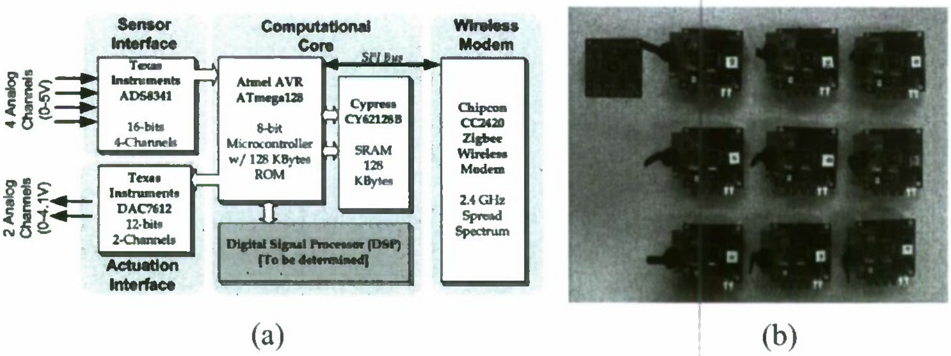


Figure 3.1 Design of a wireless node for ship systems: (a) hardware design; (b) nine fully assembled wireless sensors (a blank printed circuit board is shown in upper left corner).

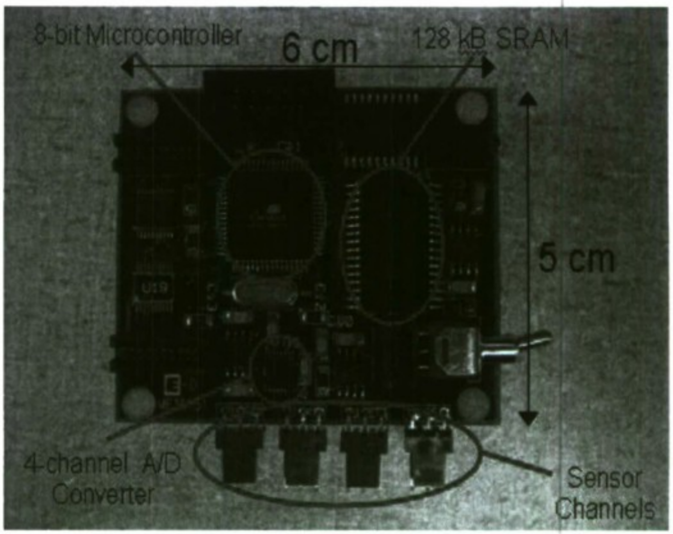


Figure 3.2. Final fully assembled wireless sensor (Chipcon transceiver not shown)

While the ADC (Texas Instruments ADS8341) has a nominal resolution of 16-bits, the presence of digital circuit elements (*e.g.*, microcontroller, radio) on the same circuit board can result in a reduction of the ADC resolution. Digital circuit elements internally contain electrical gates that open and close during their operation. As these electrical gates open, electrical charge floods into the wireless sensor ground plane. If the analog and digital circuits share a common ground plane, excess charge flooding can result in the ground exhibiting an apparent change in voltage relative to the power source of the system. The ADC employs two references that are assumed to be fixed to convert sensor outputs to digital formats: a high voltage reference (5 V power plane) and a low voltage reference (ground plane). Should those references change in any way, the ratiometric conversion of the ADC analog input (*i.e.*, sensor output) would exhibit quantization noise (random toggling of the least significant bits of the ADC digital output). The toggling of more than the 16^{th} least significant bit of a 16-bit ADC signifies a reduction of the true ADC resolution. The primary motivation of pursuing a 4-layer circuit board design was to employ separate power and ground planes for analog and digital circuit elements. Even with a 4-layer board, some resolution reduction is anticipated.

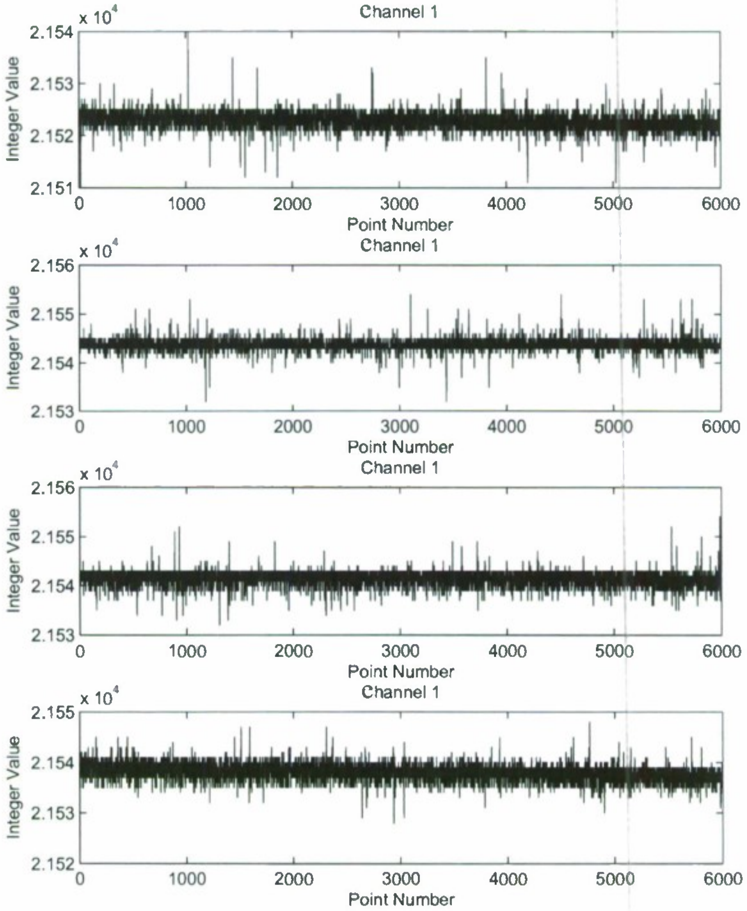


Figure 3.3. Integer output of each wireless sensor channel when a 1.5 V battery is attached.

The true resolution of the wireless sensor prototype was measured. An alkaline battery (1.5 V) was interfaced to each sensor channel of the wireless sensor (4 channels in total). Batteries are ideal power sources for such tests because they exhibit noise-free constant voltages. The wireless sensor recorded the battery analog voltage as a 16-bit digital number; the number was also wirelessly transmitted to a remote data repository where time-histories of the sensor channel voltages were stored. The time-history plots of the measured battery voltage are shown in Figure 3.3 for each of the 4 sensor channels. The time-history plots are in terms of the ADC integer values; 6000 data points collected at 200 Hz on each sensor channel.

To quantify the true resolution of the ADC, the time-history plots of the ADC integer outputs were used to generate histograms of the ADC integer outputs. To enhance the clarity of the histograms, the mean integer value is subtracted from each time history record. As shown in Figure 3.4, the zero-mean integer histograms for each sensor channel exhibited Gaussian distributions as is typical of ADC quantization error processes. The majority (>98%) of the ADC integer values fell within -4 to 4. This means the integer outputs vary by 8 integer values which is equivalent to 3-bits ($8 = 2^3$). If the resolution of the ADC is 16-bits, then only the last least significant bit is anticipated to be metastable (toggling between 0 and 1). In the case of the wireless sensor, the last three bits were toggling. As such, the wireless sensor's true ADC resolution was assessed to be 14-bits.

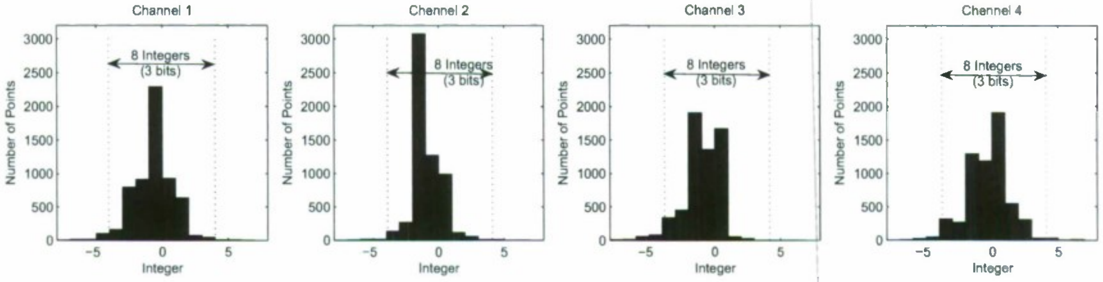
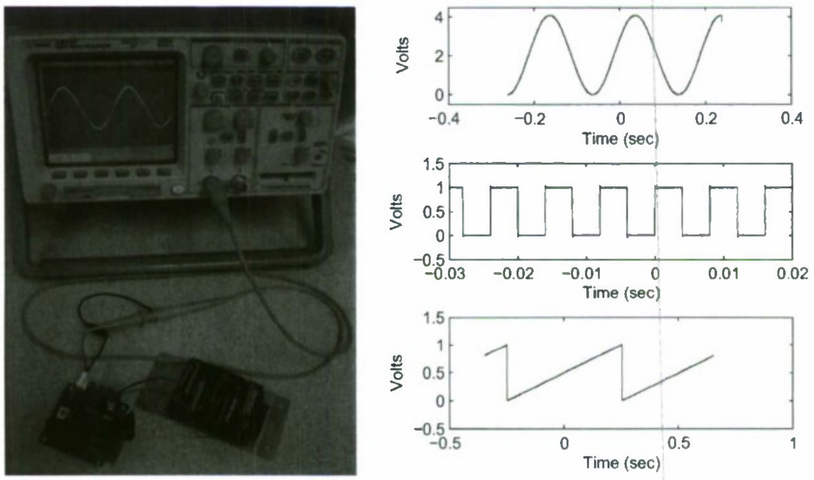


Figure 3.4. Performance of the on-board analog-to-digital converter (ADC) - histogram of ADC integer output to assess true conversion resolution.



(a)

(b)

Figure 3.5. (a) Wireless sensor prototype commanded to output a high-frequency sinusoidal wave on its actuation interface (wireless sensor, 7.5V battery source, and Agilent oscilloscope shown in this laboratory demonstration); (b) sample actuation signals generated by the wireless sensor and recorded by the oscilloscope.

3.3. Illustration of Actuation Interface Operation:

The wireless sensor is capable of outputting 0 to 4 V analog voltage signals using its on-board 2-channel DAC. To illustrate the operation of the actuation interface, a standard laboratory oscilloscope is interfaced to the actuation interface to measure the output signal generated by the wireless sensor. Figure 3.5 presents the laboratory set-up of the wireless sensor interfaced to the oscilloscope. The wireless sensor is commanded to output a variety of output signals. In particular, sinusoidal, square and triangular wave signals are generated by the wireless sensor. The signals measured by the oscilloscope are presented in Figure 3.5. As can be seen, the actuation interface was accurate (absent of noise) and could be operated at high sample rates.

3.5. Application of IEEE 802.15.4 Protocol in an Agent-Based Framework

In order to simplify the IEEE 802.15.4 standard, the LR-WPAN architecture can be broken down into a number of layers. Each layer corresponds to a specific part of the standard and provides services to higher layers. The IEEE 802.15.4 standard concentrates only on the bottom two layers of the communication protocol stack: the physical (PHY) and medium access control (MAC) layers. The PHY layer interacts directly with the physical radio in order to modulate and demodulate data and the MAC layer offers standardized packet structures for data transmission as well as a variety of methods for autonomous nodes to share common bandwidth. The upper layers, consisting of a network layer and an application layer, are application dependent and not defined explicitly by the standard.

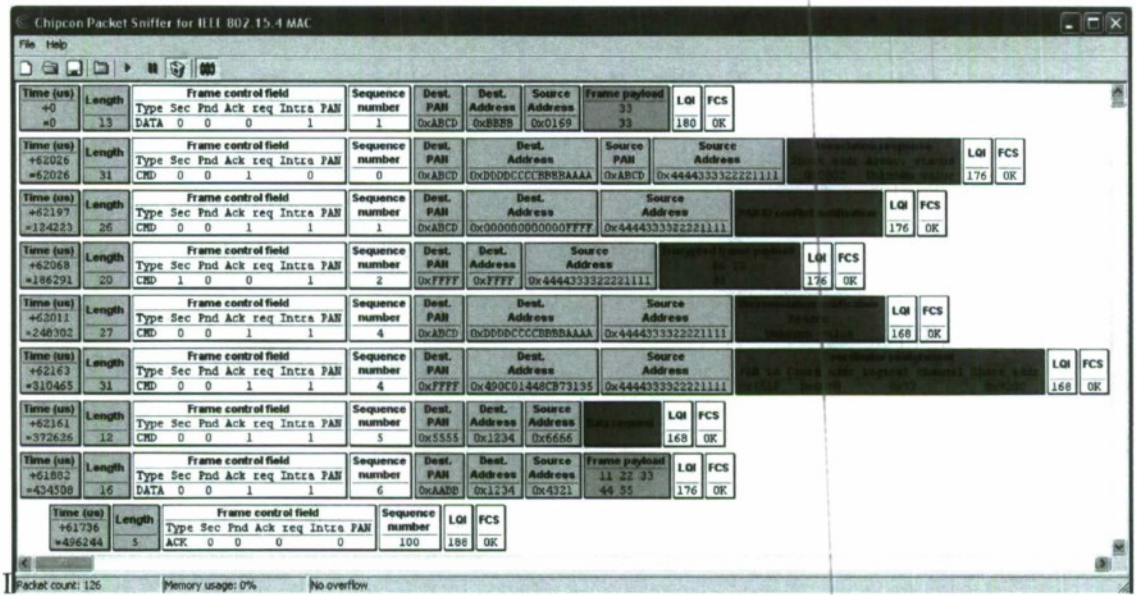
When embedded within the operating system of a wireless node, the upper layers represent any application which requires communication and data transfer services between devices. In this project, an example application may be control algorithms for naval systems. These upper layers are used to access the MAC layer for the purpose of beacon management, channel access, guaranteed time slot (GTS) management, frame validation, and acknowledged frame delivery, association, and disassociation. In turn, the MAC layer utilizes the PHY layer in order to control activation and deactivation of the radio transceiver, energy detection, link quality indication, channel selection, and clear channel assessment. The PHY layer is also used to transmit and receive packets across the physical medium.

As modern naval systems become increasingly complex, the number of sensors and actuators in a given control system and the subsequent computational demands placed on a centralized controller will grow exponentially. As a result, the ability to control such a system in a centralized manner is becoming ever more difficult. In particular, when considering a wireless control system where centralized activity across a high density wireless network is not at all feasible, an agent-based framework has been proposed in which each sensing unit behaves in an autonomous manner. Agents are software abstractions capable of sensing their environment and reacting to changes in that environment through rational embedded intelligence. Multiple agent systems are very attractive for controlling complex systems because they offer a scalable approach for solving problems using decentralized data.

In order to accommodate such a framework in a wireless setting, a layer of embedded firmware is needed to serve as a real-time operating system that allows upper software layers to collect and store sensor data, issue actuation commands, and share state data with other units within the system. The IEEE 802.15.4 communication protocol has been effectively leveraged for this exact purpose. In addition to providing a well defined application program interface (API) to the upper layers, these embedded layers of the communication protocol stack facilitate communication between multiple agents acting autonomously.

The IEEE 802.15.4 communication protocol has many advantages that cater to an autonomous network. To ensure that the network is time synchronized, each unit is designed to act in the MAC-provided beacon-enabled mode. Additionally, channel access is controlled by a slotted carrier sense multiple access collision avoidance scheme (CSMA/CA). In this mechanism, each node must sense a free channel at least twice before transmitting. The first sense is delayed by a random delay, which serves to reduce the probability of collision when two nodes simultaneously sense a free channel. When a channel is busy, transmission does not occur and an additional random delay is computed. When a packet does not reach its destination, it can be retransmitted during a new contention period.

The PHY and MAC layers have been encoded in the C programming language and loaded into the flash memory of the wireless sensing unit microcontroller. In the laboratory, a variety of MAC layer command and data frames are transmitted wirelessly from unit to unit. These command and data frames are then intercepted and interpreted by an IEEE 802.15.4 packet sniffer which is used to record all packets sent within the network. These tests display the functionality of the embedded MAC protocol as well as the wireless connectivity of the system. A screen shot of the packet sniffer results can be seen in Figure 3.6. In order to further demonstrate the ability of the PHY layer to command the physical radio to modulate and demodulate data, a testbed is created which utilizes this layer in conjunction with the sensing capabilities of the wireless sensing units. In the laboratory, a unidirectional accelerometer is attached to a



```
Chipcon Packet Sniffer for IEEE 802.15.4 MAC
File  Help

Time (us) | Length | Frame control field                    | Sequence | Dest. | Dest.   | Source  | Frame payload | LQI | FCS
          |        | Type Sec Pnd Ack req Intra PAN         | number   | PAN   | Address | Address |               |     |
+0        |        |                                        |          |       |         |         | 33            |     |
=0        | 13     | DATA  0   0     0        1             | 1        | 0xABCD| 0xBBBB  | 0x0169  | 33            | 180 | OK

Time (us) | Length | Frame control field            | Sequence | Dest. | Dest.              | Source | Source             | [illegible] | LQI | FCS
+62026    |        | Type Sec Pnd Ack req Intra PAN | number   | PAN   | Address            | PAN    | Address            |             |     |
=62026    | 31     | CMD   0   0     1        0     | 0        | 0xABCD| 0xDDDDCCCCBBBBAAAA | 0xABCD | 0x4444333322221111 |             | 176 | OK

Time (us) | Length | Frame control field            | Sequence | Dest. | Dest.              | Source             | PAN ID conflict notification | LQI | FCS
+62197    |        | Type Sec Pnd Ack req Intra PAN | number   | PAN   | Address            | Address            |                              |     |
=124223   | 26     | CMD   0   0     1        1     | 1        | 0xABCD| 0x000000000000FFFF | 0x4444333322221111 |                              | 176 | OK

Time (us) | Length | Frame control field            | Sequence | Dest. | Dest.   | Source             | [illegible] | LQI | FCS
+62068    |        | Type Sec Pnd Ack req Intra PAN | number   | PAN   | Address | Address            |             |     |
=186291   | 20     | CMD   1   0     0        1     | 2        | 0xFFFF| 0xFFFF  | 0x4444333322221111 |             | 176 | OK

Time (us) | Length | Frame control field            | Sequence | Dest. | Dest.              | Source             | [illegible] | LQI | FCS
+62011    |        | Type Sec Pnd Ack req Intra PAN | number   | PAN   | Address            | Address            |             |     |
=248302   | 27     | CMD   0   0     1        1     | 4        | 0xABCD| 0xDDDDCCCCBBBBAAAA | 0x4444333322221111 |             | 168 | OK

Time (us) | Length | Frame control field            | Sequence | Dest. | Dest.              | Source             | Coordinator realignment | LQI | FCS
+62163    |        | Type Sec Pnd Ack req Intra PAN | number   | PAN   | Address            | Address            | [illegible]             |     |
=310465   | 31     | CMD   0   0     1        1     | 4        | 0xFFFF| 0x490C01448CB73135 | 0x4444333322221111 | [illegible]             | 168 | OK

Time (us) | Length | Frame control field            | Sequence | Dest. | Dest.   | Source  | Data request | LQI | FCS
+62161    |        | Type Sec Pnd Ack req Intra PAN | number   | PAN   | Address | Address |              |     |
=372626   | 12     | CMD   0   0     1        1     | 5        | 0x5555| 0x1234  | 0x6666  |              | 168 | OK

Time (us) | Length | Frame control field            | Sequence | Dest. | Dest.   | Source  | Frame payload | LQI | FCS
+61882    |        | Type Sec Pnd Ack req Intra PAN | number   | PAN   | Address | Address | 11 22 33      |     |
=434508   | 16     | DATA  0   0     1        1     | 6        | 0xAADD| 0x1234  | 0x4321  | 44 55         | 176 | OK

Time (us) | Length | Frame control field            | Sequence | LQI | FCS
+61736    |        | Type Sec Pnd Ack req Intra PAN | number   |     |
=496244   | 5      | ACK   0   0     0        0     | 100      | 188 | OK

Packet count: 126    Memory usage: 0%    No overflow
```

Figure 3.6. Laboratory validation of full implementation of IEEE 802.15.4 MAC and PHY layers

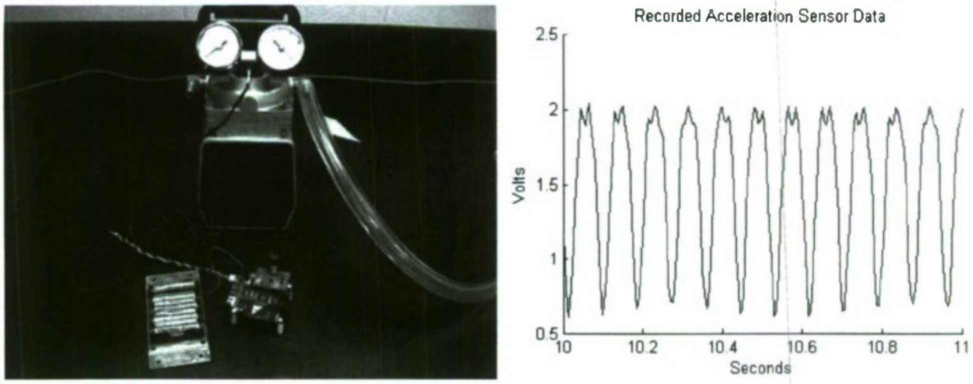


Figure 3.7. Experimental setup and acceleration time history plot for laboratory validation of PHY layer for sensing applications.

vacuum pump. The pump is turned on and acceleration data is collected with a wireless sensing unit at a sampling rate of 500 Hz. This data is then transmitted back to a centralized server and stored in a local data repository. Figure 3.7 shows the experimental setup and the time history acceleration record of the pump vibration.

4. Model Updating for System Assessment and Damage Detection:

In a naval environment, it is imperative that distinct changes in the operational condition of naval plants be detected, as the detection of such state changes will lead to the initiation of market-based control solutions adopted to optimally allocate scarce system resources. As such, this section outlines the creation of a novel data processing architecture used to continuously update system models representing a fully-functional ship plant (*e.g.*, chilled water supply system). Changes to such a plant are identified through model updating methods conducted on the wireless sensor network, and can be indicative of damage. Model updating methods are typically computationally intense and are often reserved for data servers where significant computational resources are located. In our approach, a parallel implementation of the simulated annealing model update method (a stochastic search algorithm) is embedded upon a network of Narada wireless sensors to identify bifurcations in the system, perhaps due to battle damage. Emphasis is made to keep the update time to near real-time (less than ten minutes) and to ensure that the full computational capabilities of the wireless sensor network (albeit, highly distributed) are maximized.

By viewing a wireless network as a parallel computer with an unknown and possibly changing number of processing nodes, this distributed data processing architecture is capable of performing complicated types of data analysis while creating a scalable environment that is not only resistant to communication and sensor failure, but that also becomes increasingly efficient at higher nodal densities. This novel architecture functions by allowing a network of sensors to autonomously detect and utilize the computing resources of any available wireless node on the fly. This “ad-hoc” capability allows for increases in the parallelism and efficiency of the architecture in real-time, and can be used to reform or “self-heal” the network in the wake of any communication and/or sensor failures.

In order to examine the data processing capabilities of this novel architecture from an damage detection perspective, a parallelized version of the simulated annealing (SA) stochastic optimization method is designed for implementation within a distributed WSN. One of the reasons that the SA algorithm is chosen for parallelization is that it can be applied to many of the optimization problems that may arise in the naval environment. In this study, the wireless parallel SA (WPSA) method is used within a WSN for the updating of an analytical model of a shipboard chilled water system.

4.1. Background on combinatorial optimization by simulated annealing:

One of the most studied areas in computational engineering is that of combinatorial optimization (CO). This field involves developing efficient methods for finding the maximum or minimum value of any function with a large number of independent variables. CO problems are typically very difficult to solve computationally, as an exact solution often requires a number of computational steps that grows faster than any finite power of the size of the problem. As such, it is often desirable in engineering applications to quickly find good approximations to the optimal solution instead of expending the time and resources required to find an absolute global optimum. Unfortunately, even approximate solutions can sometimes be difficult to find, as most relevant search strategies involve iterative improvement, and as such, have a tendency to get stuck in local (not global) optima. However, in the 1980’s, several algorithms derived from physical and biological systems were developed for finding near-global optima in functions containing many local optima (Bounds 1987). One of these methods is the simulated annealing (SA) optimization technique, first presented by Kirkpatrick *et al.* (1983).

SA was developed out of the observation that a connection could be made between CO and the behavior of physical material systems in thermal equilibrium at a finite temperature. In material physics, experiments that determine the low-temperature state of a material are performed by first melting the substance, and then slowly lowering the substance’s temperature, eventually spending a long time at temperatures near freezing. This annealing procedure allows the substance to eventually obtain an optimal thermal energy state amongst an almost infinite number of possible atomistic configurations. Assuming that a method exists for determining the energy of a physical system in a specific atomistic configuration, this physical annealing procedure can be viewed as a CO problem where the objective is to find the globally minimal energy state of the material's atoms.

As such, by borrowing ideas from the natural annealing process, a "simulated" version of the annealing method can be developed to quickly obtain good approximate solutions to CO problems where the objective is to find a globally minimal value of some optimization function. This is done by viewing the value of the function to be optimized as the physical system "energy", introducing an "effective" annealing temperature which will simulate the material cooling process, and utilizing the Metropolis procedure (explained below) to avoid premature convergence on local optima, which is the key to the effectiveness of the generalized annealing process.

In 1953, Metropolis *et al.* created an algorithm that can probabilistically simulate a collection of atoms converging on thermal equilibrium at a set temperature. At each step in this algorithm, a randomly selected atom is displaced a small, random distance, and the resulting change in system energy (ΔE) is computed. If $\Delta E \leq 0$, this disturbance is accepted. Otherwise, if $\Delta E > 0$, the new configuration will be accepted with the following probability:

$$\Pr(accept) = e^{\frac{-\Delta E}{k_B \cdot T}} \tag{4.1}$$

where T is the temperature of the system and k_B is Boltzmann's constant. If the new configuration is accepted, the next step of the search continues with that atom displaced. Otherwise, if the new configuration is not accepted, the next step in the search continues using the original atomistic configuration. By repeating this procedure many times, Metropolis simulates the thermal motion of atoms subjected to a constant temperature, and mimics the probabilistic process by which nature avoids premature convergence on suboptimal configurations.

As proposed by Kirkpatrick *et al.* (1983), "simulated" annealing can be used in the context of CO by representing each possible configuration of optimization function parameters as a distinct state, s. The objective of the annealing process is to find a system state that minimizes the value of an optimization function, $E(s)$. In order to help avoid convergence on a sub-optimal minimum, the Metropolis framework can be applied to the SA procedure by generating a new state, s_{new}, by altering the value of one function parameter at random. The objective function value of this new state, $E(s_{new})$, is then compared with the objective function value of the old state, $E(s_{old})$, and the new state is probabilistically accepted or rejected based on the criterion presented in Equation 4.1. When SA is implemented within a computing machine, the probability of a new system state being accepted at a given temperature can be stated as follows: accept a new state, s_{new}, if and only if:

$$E(s_{new}) \leq E(s_{old}) + T \cdot |\ln(U)| \tag{4.2}$$

where U is a uniformly distributed random variable between 0 and 1. The addition of the $T \cdot |\ln(U)|$ term allows the system to periodically accept a sub-optimal state in hopes of avoiding premature convergence on a local optima. A standard SA cooling schedule begins the optimization process by assigning a high initial temperature T_0 and then letting the Metropolis algorithm run for N iterations. During each iteration, a new psuedorandomly generated state is created by modifying one of the optimization parameters, and the newly generated state is either rejected or accepted based on the Metropolis criterion (Equation 4.2). After N iterations, the temperature of the system is reduced by a factor of ρ, such that $T_{new} = \rho \cdot T_{old}$, and N additional iterations will be run at the new, lower temperature (T_{new}). This process continues until the temperature is sufficiently low that very few new states are accepted, meaning that a globally optimal state has likely been found and the system has, in essence, frozen.

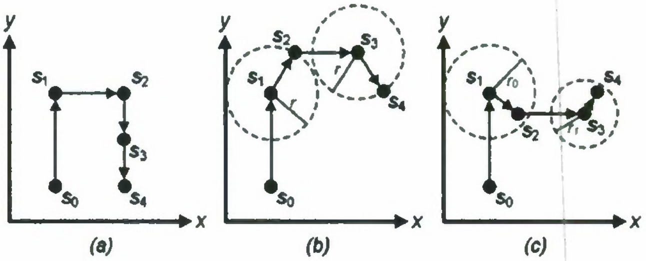


Figure 4.1: Random state generation for a two-dimensional search problem using (a) standard SA, (b) BSA, and (c) WPSA (Zimmerman and Lynch *in press*).

Since Kirkpatrick *et al.* first published the SA methodology in 1983, countless variations on the original algorithm have been seen in the literature. For each specific optimization problem, it seems, a different variant on the traditional SA method provides the quickest convergence and the most accurate results. As such, it is important to note that the WPSA methodology proposed herein for use in wireless sensor networks can be effectively utilized in conjunction with almost any variant on the SA method. However, for the model updating problem studied in this paper, a modification on the blended simulated annealing (BSA) algorithm proposed by Levin and Lieven (1998) is exclusively utilized. The BSA algorithm deviates from the standard SA methodology in the way in which it creates randomly generated states. In standard SA, new states are generated by randomly choosing one annealing parameter and assigning it a new value chosen uniformly from within the parameter's valid range (Figure 4.1(a)). In the BSA algorithm, however, this standard type of state generation is alternated every other step with a "radius adjustment" approach, where all annealing parameters are changed by choosing a random point *on* a hypersphere that is a fixed radius away from the previous annealing state (Figure 4.1(b)). This method requires two separate annealing temperatures, one for the standard SA adjustment and one for the radius adjustment. For this study, the BSA algorithm is modified slightly such that instead of choosing a point that lies on a fixed radius from a previous annealing state, all annealing parameters are randomly assigned new values that reside *within* a given radius from the individual parameter's current assignment (Figure 4.1(c)). Then, the radius itself is treated as a variable in the SA process much like the annealing temperature. It starts with a high value near 1.0 (such that the entirety of each parameter's valid range can be searched), and as time progresses, the searchable radius is reduced such that the SA search focuses increasingly on values that are close to the currently optimal state. This improves upon the BSA algorithm by eliminating the wasteful interrogation of search states far away from the currently optimal, especially later in the search as a final, optimal solution is converged upon.

4.2. Wireless parallel simulated annealing:

When considering performing CO tasks on a wireless sensor network, SA may at first appear to be an excellent candidate for a stochastic search procedure. Because a search using SA requires only a negligible two or possibly three states to be stored in memory at any one time, SA is extremely attractive in the wireless setting where memory capacity within most prototypes is limited. However, the computational costs of implementing SA, which may require a value of E to be determined at hundreds of thousands of randomly generated states in order to converge on an optimal solution, can be staggering. When implemented within a single wireless sensing device, where processing speed is usually only a fraction of that of an ordinary personal computer, this is a potentially debilitating problem.

In order to mitigate the computational demands imposed by SA, many researchers have developed parallel SA techniques that, when run on a large number of processors, can successfully increase the speed with which a solution to a CO problem can be obtained (Greening 1990). However, most of these methods require communication between processors both before and after each random state is generated. In the wireless setting, where battery preservation is a high priority and communication bandwidth is limited, this type of constant communication negates the advantages of parallelism and represents a poor use of battery power. In this study, a parallel SA procedure is created that utilizes the computational resources distributed across large wireless sensing networks while minimizing the communication demands of the parallel algorithm. This is done by taking advantage of the fact that the SA process typically rejects more states than it accepts, especially as the annealing temperature is lowered and the algorithm converges on a solution. Specifically, the traditionally serial SA search problem (which is continuous across all temperature steps) can be broken into a set of smaller search trees, each of which corresponds to a given temperature step and begins with the globally optimal state assignment so far detected at the preceding temperature step. Each smaller search problem can then be assigned individually to any available sensor in the network, and thus multiple temperature steps can be searched concurrently. This concept is displayed graphically in Figure 4.2. One of the great advantages of this methodology is that, given the ad-hoc communication capabilities of many wireless sensing devices, these individualized search trees can be distributed in real-time to any available processor within the sensing network. Because the ad-hoc assignment and reassignment of search problems can allow for individual nodes to drop from,

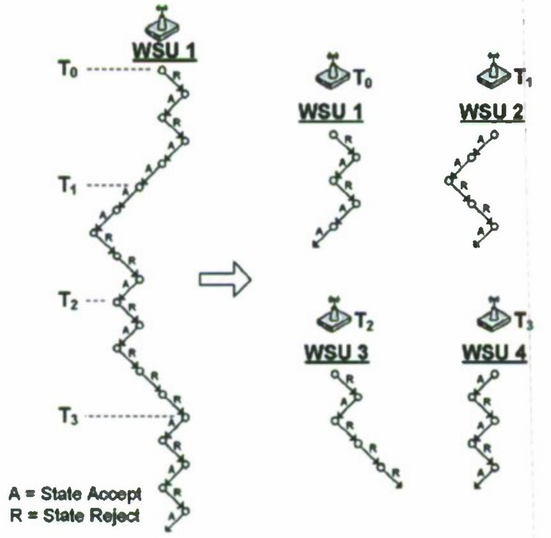


Figure 4.2: A simple serial SA search tree, shown up to the fourth temperature step and its corresponding WPSA search trees, assigned to wireless sensors (Zimmerman and Lynch *in press*).

or appear in the network mid-search, this parallelized updating method is incredibly valuable in systems where sensor or communication reliability may be in question.

4.2.1. Wireless implementation of the WPSA algorithm:

In the parallel SA implementation used in this study (WPSA), a computational task requiring SA optimization is first assigned to any one available sensing unit, along with a user-defined initial temperature, T_0. This first wireless sensor, n_0, then beacons the network, searching for other sensors available for data processing. If a second sensing node, n_1, is found, the first sensor, n_0, will assign the SA search tree starting at the next temperature step, T_1, to the second sensor, n_1, passing along its current information regarding the most optimal system state yet visited. This process continues until no sensors remain available for data processing.

If a given sensor, n_i, detects an optimal solution, (*i.e.*, no new states are accepted at the temperature step sensor n_i is investigating), it will order the rest of the network to discontinue the SA search, and will alert the network end-user of the discovered results. However, if sensor n_i finishes its part of the SA search without having converged on a solution (*i.e.*, new states are still being accepted), it will alert its successor, n_{i+1}, that no solution was found at temperature step T_i, and sensor n_i will again make itself available to the network for computation on a lower temperature step. While WPSA functions autonomously without need for a centralized controller, the WSU assigned to the highest temperature step at any given time keeps track of search progress and alerts the user when the search has been completed. Because of the self-healing capabilities of many WSNs, this parallel algorithm will always adapt in order to utilize the maximum number of processing nodes available at any one time, even if some sensors drop in and out of the network during computation.

As the WPSA search continues, information regarding newly found, increasingly optimal states is disseminated downwards through the network, such that all sensors are cognizant of any search progress that has been made at higher temperature steps. This allows all sensors to maximize the effectiveness of their search at a given temperature step, and maintains the continuity of the serial SA process. Specifically, when a sensor detects a state, s, with a lower optimization function value than that of any other known state, it will immediately propagate this information downward to the sensor directly below it in the search tree (its child). If the propagated state information also represents the minimal value of the objective function that the child has found so far, the child will then restart its N search iterations from the newly found minimum state and inform the sensor directly below it of this newly discovered state. However, if a child receives a state, s_p, from a parent, and the child has already randomly generated a state, s_c, that yields a lower objective function value than s_p, ($E(s_c) < E(s_p)$), that child will merely restart its SA iterations given its current search state, s_c, without passing any information on to its successor. In this way, it is assured that each temperature step is thoroughly searched given the complete information obtained at the preceding temperature step. While this does result in an increase in the total number of SA iterations required to reach a solution over the serial SA procedure, the additional randomly generated states at many (if not all) temperature steps slightly increases the probability that a "better" solution will be found than otherwise possible.

4.2.2. Illustrative example of the WPSA algorithm:

Figure 4.3 illustrates the distribution of one example parallelized simulated annealing task over a network of four wireless sensing units. This task has an initial global minimum objective function value of $E = 10$, and is assigned by the user to wireless sensing unit 1 (WSU 1). Simultaneous to this assignment, the user also alerts all other sensors that they should make themselves available for computation. After receipt of this task assignment, WSU 1 recognizes that WSU 2 is an available computational node, and orders this unit to perform N search iterations at the second temperature step, starting with WSU 1's current global state (with a global minimum of $E = 10$). In a similar way, WSU 2 assigns the third temperature step to WSU 3, and WSU 3 assigns the fourth temperature step to WSU 4.

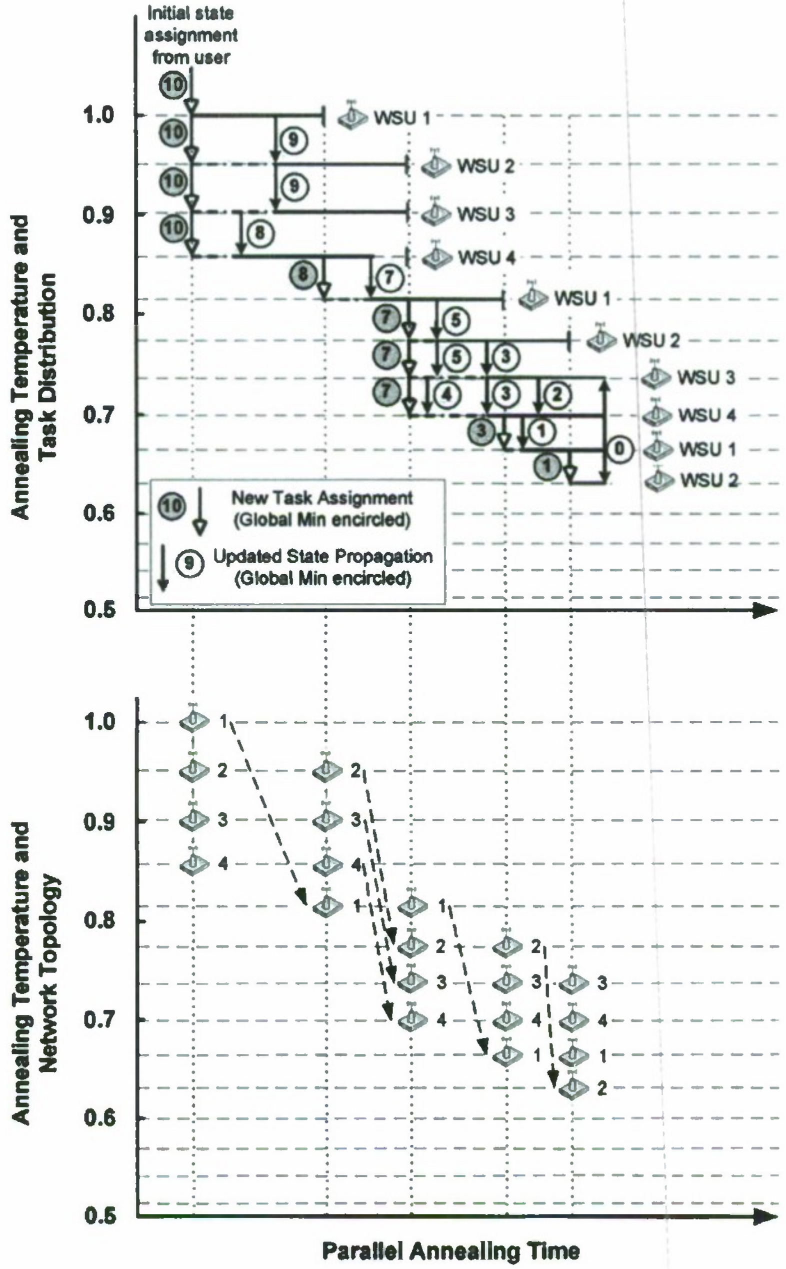
After searching approximately *N*/4 SA-generated search states, WSU 3 detects a state with an objective function value of 8. It immediately passes this state information along to its child, WSU 4. Because WSU 4 has a current global minimum value of $E = 10$, WSU 4 restarts its search of *N* SA-generated states at the fourth temperature step with this updated information. WSU 4 has no children, so the propagation of this new state stops when it reaches WSU 4.

Soon thereafter, WSU 1 detects a SA-generated state with an objective function value of $E = 9$. This is lower than its current minimum value of $E = 10$, so it informs its child, WSU 2 of the newly found state. WSU 2 recognizes $E = 9$ as a new global minimum, so it restarts its search at the second temperature step with this information and passes the updated values along to its child (WSU 3). WSU 3, however, has already detected a global minimum of $E = 8$, and thus it simply restarts its search of *N* SA-generated states with its current state information (without needing to send any updated information to its child).

When WSU 1 finishes its search of *N* SA-generated states, it alerts its child (WSU 2) that it has not found a globally optimal solution, and it disengages from the search process. At this point, however, WSU 1 broadcasts its availability to the other nodes in the network. WSU 4, which is in need of a child node, assigns the SA task at the fifth temperature step to WSU 1, given its current state (with a global minimum value of $E = 8$).
This process continues for several more temperature steps until WSU 1 detects a globally optimal state (*i.e.* it finds a state with an objective function value of $E = 0$). At this point, WSU 1 broadcasts its find to the network, thereby stopping all other computation, and it alerts the user that a globally optimal state has been found.

4.3. Experimental Validation of the WPSA algorithm within a Naval Chilled Water System:
To validate the performance of the proposed distributed WPSA process for damage detection, a set of experiments were run on simulated data from a very simple Navy shipboard chilled water system (shown in Figure 4.4). The pipe network in this system contains sixteen pipes of equal diameter but varying length. The system is driven by two identical pumps, located along pipes three and four, and there are four flowmeters measuring flow within the network. Each of these flowmeters is monitored by a single wireless sensing unit.

An analytical model of the pipe network described above is designed and embedded within each sensing node. Given a particular system configuration, this software can determine the expected flowrate and head loss in each pipe. By conserving mass at each pipe junction and using the linearized Darcy-Weisback equation to relate head loss in loops, a set of linear equations describing the piping network can be automatically formulated. These equations can then be solved using Gauss-Jordan elimination, and values for flowrate can be found. However, because head losses in this environment are dependent on flowrates and flowrates are dependent on head losses, this process must iterate until a solution is converged upon. Once embedded within the Narada's Atmega128 processor, each execution of the Gauss-Jordan process is timed and found to take roughly 0.73 seconds.



Initial state assignment from user
Annealing Temperature and Task Distribution
1.0
0.9
0.8
0.7
0.6
0.5
WSU 1
WSU 2
WSU 3
WSU 4
WSU 1
WSU 2
WSU 3
WSU 4
WSU 1
WSU 2
10
9
8
7
5
4
3
2
1
0
New Task Assignment (Global Min encircled)
Updated State Propagation (Global Min encircled)
Annealing Temperature and Network Topology
1.0
0.9
0.8
0.7
0.6
0.5
1
2
3
4
Parallel Annealing Time

Figure 4.3: One parallel simulated annealing task running on four wireless sensors

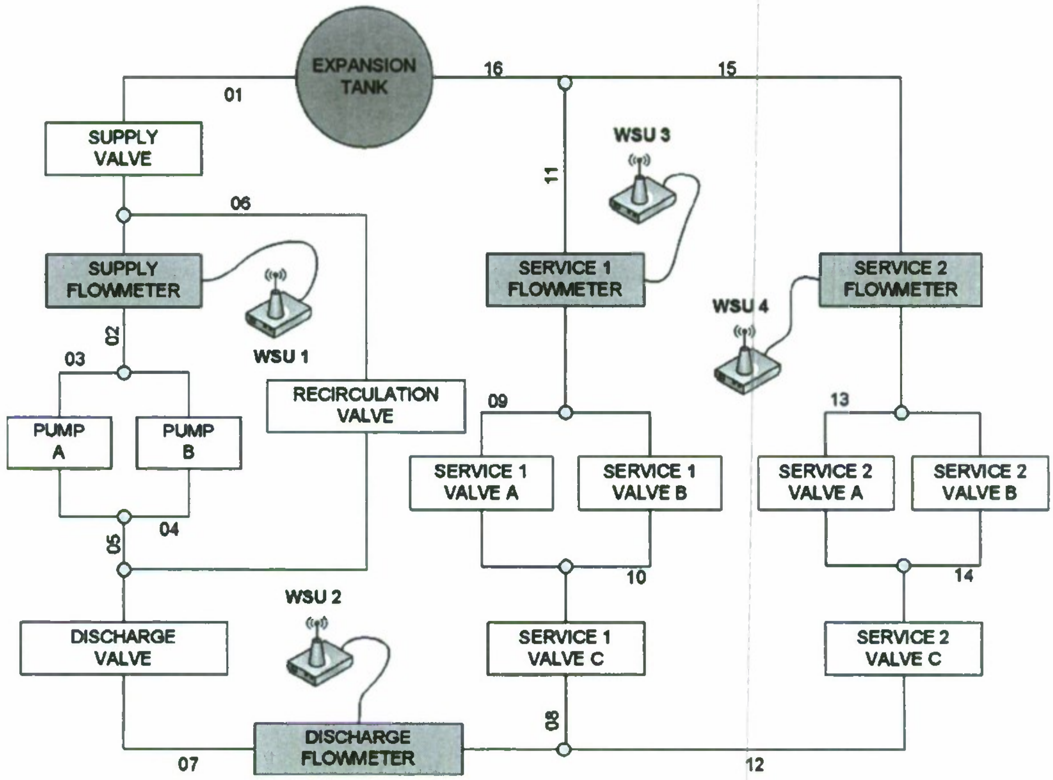


Figure 4.4: A simple Navy shipboard chilled water system.

Because flowrate data is constantly available at several locations throughout the pipe network, an objective function can easily be created to compare experimental observations with analytical model projections:

$$E = \sum_i \sqrt{\left(Q^i_{anal} - Q^i_{sense}\right)} \tag{4.3}$$

where E is the objective difference between analytical and experimental results and Q^i_{anal} and Q^i_{sense} are the analytical and sensed flowrates, respectively, in pipe i.

Using the SA method, this objective function can be repeatedly evaluated given different values for a set of model properties until an "updated" configuration of parameters is found that minimizes the objective difference between expected and observed flowrates. Because changes in this objective function can be interpreted as a direct indicator of damage, the choice of updating parameters is extremely important in terms of the ability to detect and localize system bifurcations. In the case of a chilled water system, it is advantageous to choose updating parameters that strongly influence flowrate, such as pipe diameter. By doing so, it becomes possible to look for changes in pipe properties as an indication of suboptimal performance.

For the sake of simplicity, it is assumed that all damage in this system is local to sector three, which contains pipes 12,13,14, and 15. In order to account for a reduction in flow rate that may result from a blocked or leaking pipe, pipe diameters for each of these four pipes are designated as the updating parameters for this problem. All pipes in the system are initialized with a diameter of 0.5 inches. In order to introduce simulated damage to the system, sensor results are simulated with pipe 13 having an 80% reduction in pipe diameter.

Having obtained sensor readings for the damaged state, a central server sets up the model updating problem and chooses one sensing unit to serve as the initial network coordinator. This unit then searches for an available node to assign additional tasks to, and the SA process continues in an ad-hoc fashion until a set of updated parameters are converged upon. A time series representation of pipe diameter convergence can be seen in Figure 4.5.

In order to evaluate the effectiveness of the parallelized updating algorithm, an entire model update is completed several times with varying numbers of Narada nodes made available for computation. It can be seen from Figure 4.6 that this type of parallelization can drastically decrease the time required to find updated model properties and detect damage within a system. It can also be seen from this figure that increasing the number of computing sites also improves the accuracy of the final solution. Figure 4.7 shows the ability of this algorithm to accurately detect and locate system bifurcations. It can be seen that the SA method rightly identified a significant diameter reduction in pipe 13, while pipes 12,14, and 15 only observe minor variations.

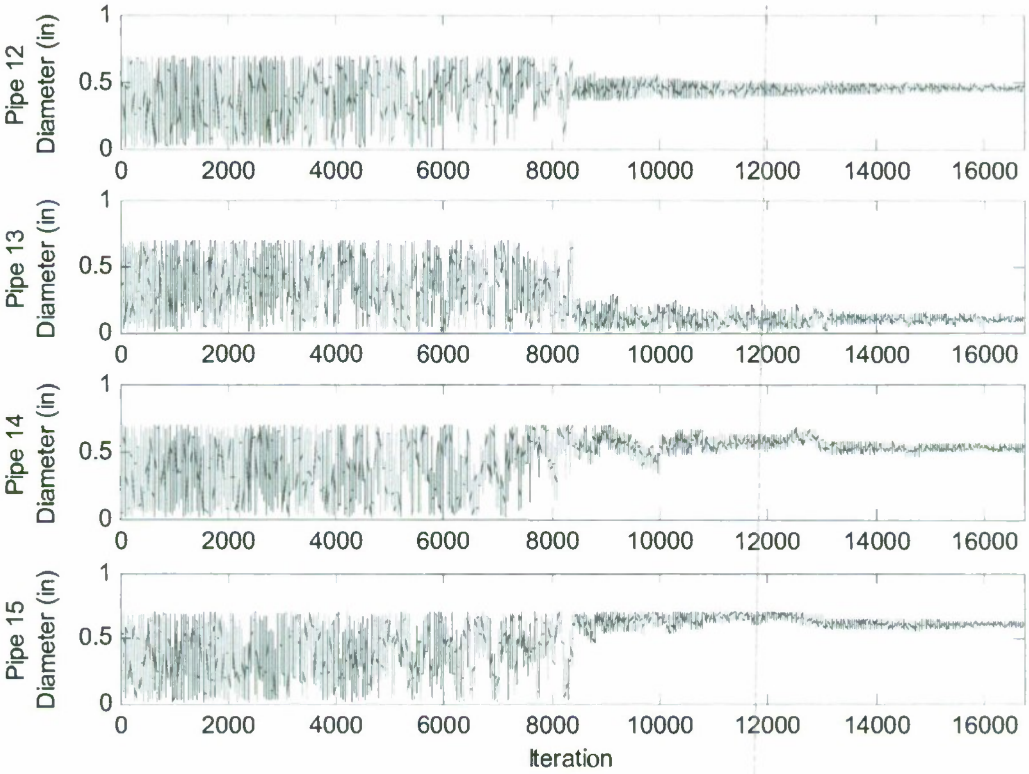


Figure 4.5: SA pipe diameter convergence.

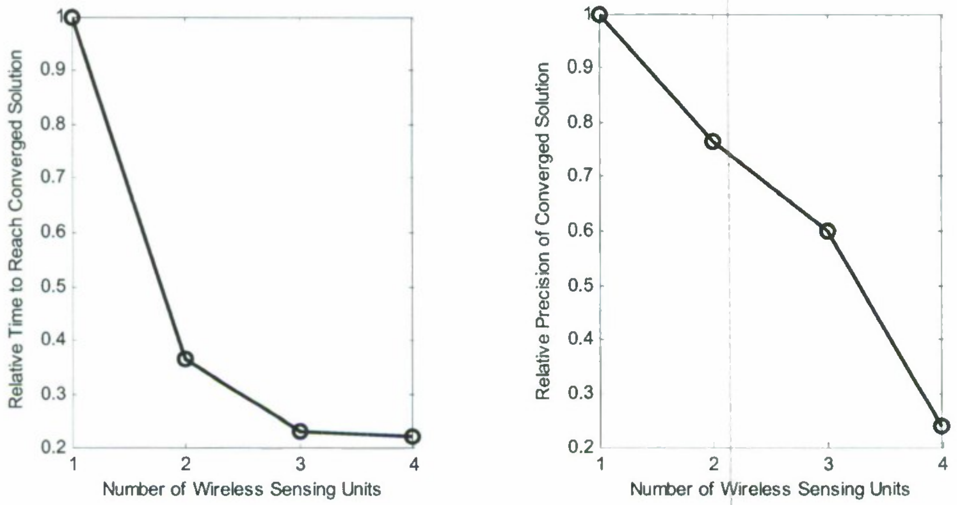


Figure 4.6: Speed and accuracy improvements using WPSA.

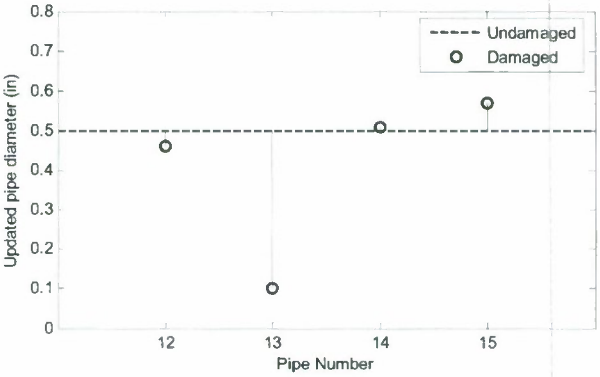


Figure 4.7: Speed and accuracy improvements using WPSA.

5. Experimental Testing of Narada Wireless Sensors in a Chilled Water Plant:

In order to experimentally verify the ability of the Narada wireless sensor to perform sensing, actuation and control tasks relevant to the Naval environment, the ONR tabletop chilled water demonstration system was chosen as an appropriate testbed. Field validation of the Narada sensing platform was performed in two phases, with Phase I focusing on the integration of the wireless sensing environment within the chilled water system and Phase II investigating the ability of a network of Narada wireless sensors to detect and locate plant damage using the model updating methodology presented in Section 4.

5.1. Phase I (Proof of Concept) Field Work Statement:

This section of the report details the results of field testing performed on June 25th and 26th, 2008, at the Naval Surface Warfare Center in Philadelphia, PA. On these dates, the basic interface capabilities of the University of Michigan's Narada wireless sensing system were tested using the ONR tabletop chilled water demonstration system. The purpose of these field tests was to validate the ability of the Narada wireless system to record flow data within a chilled water system, to archive this data for later access, and to lay the groundwork for the use of embedded model updating techniques to detect and locate pipe or pump degradation within the chilled water system.

5.1.1. Phase I Objectives:

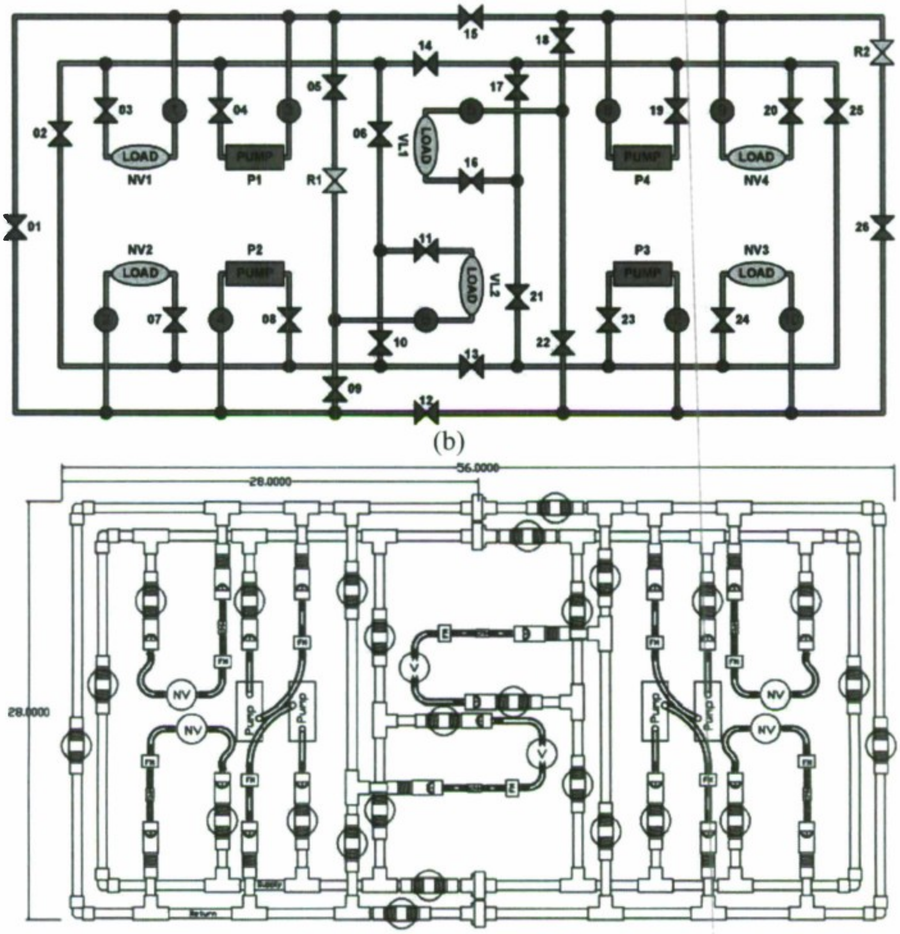
The objectives of the Phase I field work can be defined as follows:

1. Interface Narada wireless sensors with the flowmeters already installed on the ONR tabletop chilled water demonstration system.
2. Validate the ability of the Narada sensors to collect analog flowmeter data from the tabletop demonstrator and convert it into digital flow readings.
3. Collect full-system flow data from the tabletop demonstrator in an undamaged state.
4. Introduce simulated damage to the tabletop demonstrator through the use of rupture valves, pump shorts, and valve closures.
5. Collect full-system flow data from the tabletop demonstrator in a damaged state.

5.1.2. Integration of Narada wireless sensors with tabletop flowmeters:

The ONR tabletop chilled water demonstration system (shown in Figure 5.1) was developed as a testbed on which to validate new sensing and control technologies. This tabletop system, which emulates many of the electrical and mechanical aspects of a shipboard chilled water system, contains four pumps which provide chilled water through a series of 42 pipes to two vital and four non-vital "loads". In reality, these loads are things vital to shipboard functionality (such as electrical or mechanical equipment), but on the demonstrator they are simply visual flow indicators. Each pipe that provides flow to a pump or a load is instrumented with a flowmeter (for a total of 10 sensors), and a set of 26 valves are in place to control the flow to different sectors of the system. Two rupture valves are also installed in order to create a controllable way of simulating damage within the pipe network. The testbed also contains an electrical system built to emulate shipboard conditions, but for the purposes of this study only the mechanical pipe network is of interest. Details of the equipment used to interface with the Narada sensing system can be seen below:

- DigiFlow DFS-2W Flowmeter (Figure 5.2)
  - Excitation Voltage: 5V
  - Measurement Range with Narada Sensor: 0 mL/sec – 27.7 mL/sec
- Greylor PQ-12 DC Gear Pump (Figure 5.3)
  - Characteristic Equation: $h_p = -268.527Q^2 + 86.78Q + 40.272$
    - Where Q is in GPM and h_p is in ft
- STC 2W040-3/8-D Solenoid Valve (Figure 5.4)
  - Excitation Voltage: 12-24V



(b)

(a)

Figure 5.1. ONR tabletop chilled water demonstration system, (a) plan view and (b) schematic.

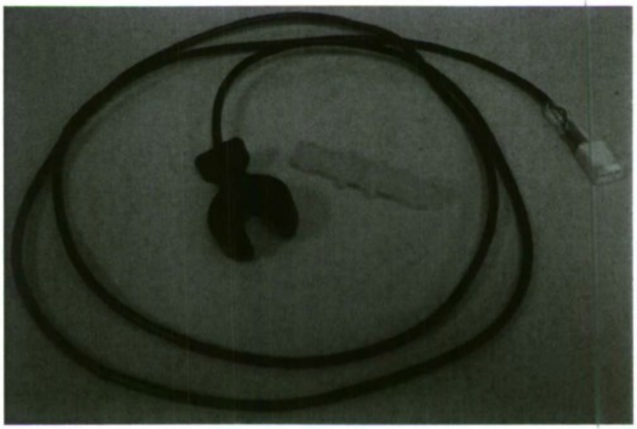


Figure 5.2. Digiflow DFS-2W Flowmeter, modified for connection to Narada wireless sensor.

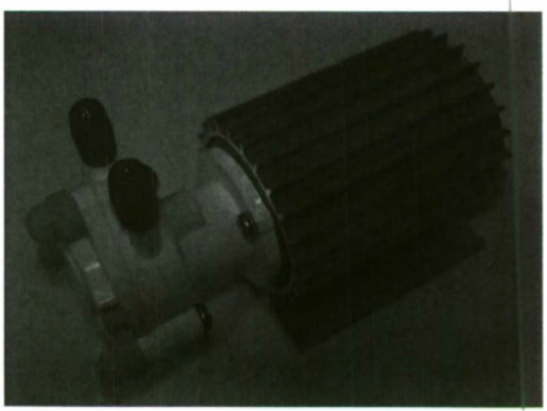


Figure 5.3. Greylor PQ-12 DC Gear Pump.

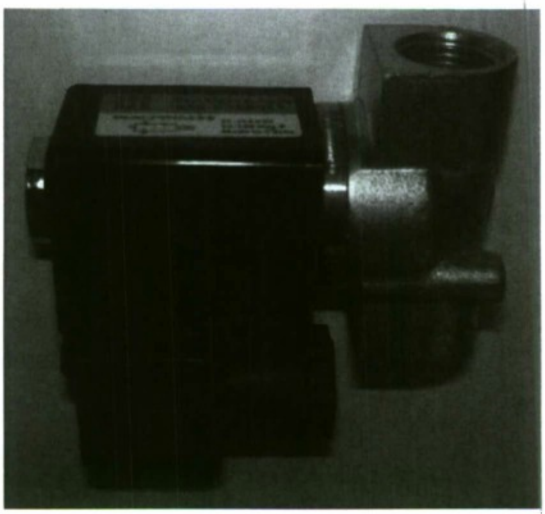


Figure 5.4. STC 2W040-3/8-D Solenoid Valve.

Each of the flowmeters in the tabletop system functions in the same way: as fluid flows through a meter, it first passes a fixed worm. This worm imparts a spiral flow which, in turn, spins a rotor on a virtually friction-free bearing. As they spin, the rotor blades interrupt an infrared beam coming from a fixed external electronics mounting assembly, thus generating a square wave output signal. Each flowmeter is powered by a 12V DC power supply. However, through the use of an external resistor, the nominal output voltage of the sensor is approximately 3.5V. Each time the rotor blades spin, an open circuit is created and this sensor voltage drops briefly to 0V. An example of this analog output data can be seen in Figure 5.5. One Narada wireless sensor is utilized to monitor each flowmeter (Figure 5.6). During the course of this field test, it was found that the voltage supply being utilized was providing an excessive amount of noise to the sensor readings. This noise was likely present because the original power supply was being utilized to power not only the flowmeters, but all of the demonstrator's valves and pumps. The resulting noise prevented the square wave signal from reaching the 0V level, and made it very difficult to extract digital flowrate information from the analog sensor. An example of this noisy output can be found in Figure 5.7. A different power supply was brought in for the sole purpose of powering the system's ten flowmeters, and this solution appeared to solve the problem and produce clean square wave output (Figure 5.5).

In order to convert this pulsing analog voltage data into flowrate, specialized data acquisition code is written for the Narada wireless sensors. When asked to record flow data, the sensors directly sample the flowmeter signal at a very high sampling rate (5000Hz), counting how frequently a pulse occurs by looking for "falling edges" or pairs of data points where the first is above 3V and the second is below 3V. The amount of time between pulses is determined and a conversion factor is used to store a "current flowrate" value in memory (Equation 5.1).

$$flowrate\left(\frac{mL}{\sec}\right) = \frac{\frac{1}{60}\left(\frac{mL}{pulse}\right)}{x\left(\frac{\sec}{pulse}\right)} \tag{5.1}$$

Then, this "current flowrate" value can be sampled much more slowly by the traditional data acquisition algorithm. This method allows the user to maximize the sampling speed of the Narada hardware while not needlessly wasting communication bandwidth and hard drive space by storing large amount of extremely high frequency data. Lastly, in order to account for variability and resolution error in the edge counting algorithm, the converted flowrate is smoothed using a rolling average. A visual example of the Narada's final flowrate output (sampled at 200 Hz) can be seen in Figure 5.8. The blue dots in this figure represent the converted and sampled flowrates and the red line represents the rolling average.

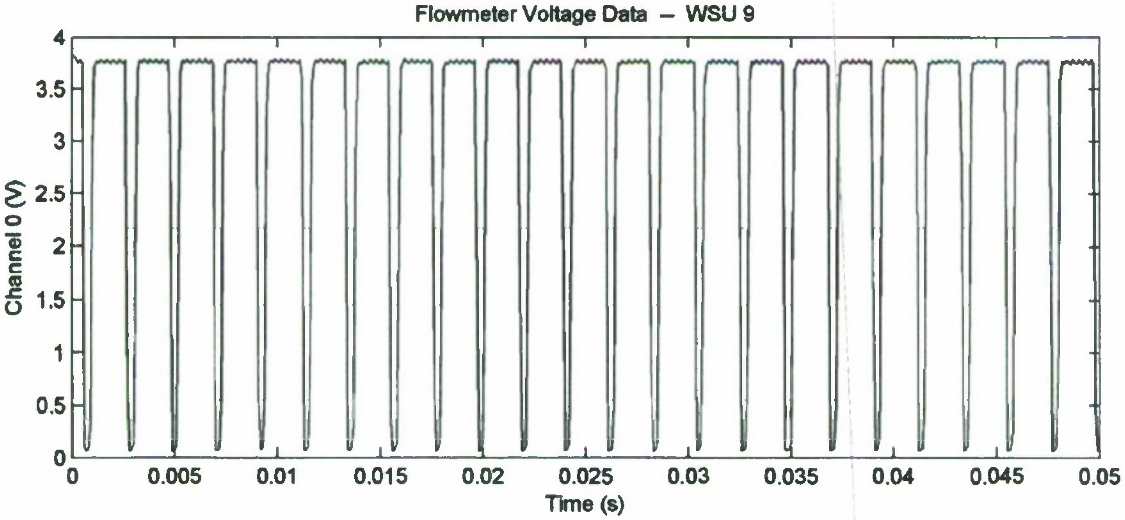


Figure 5.5. Sample flowmeter voltage data, collected with a Narada wireless sensor.

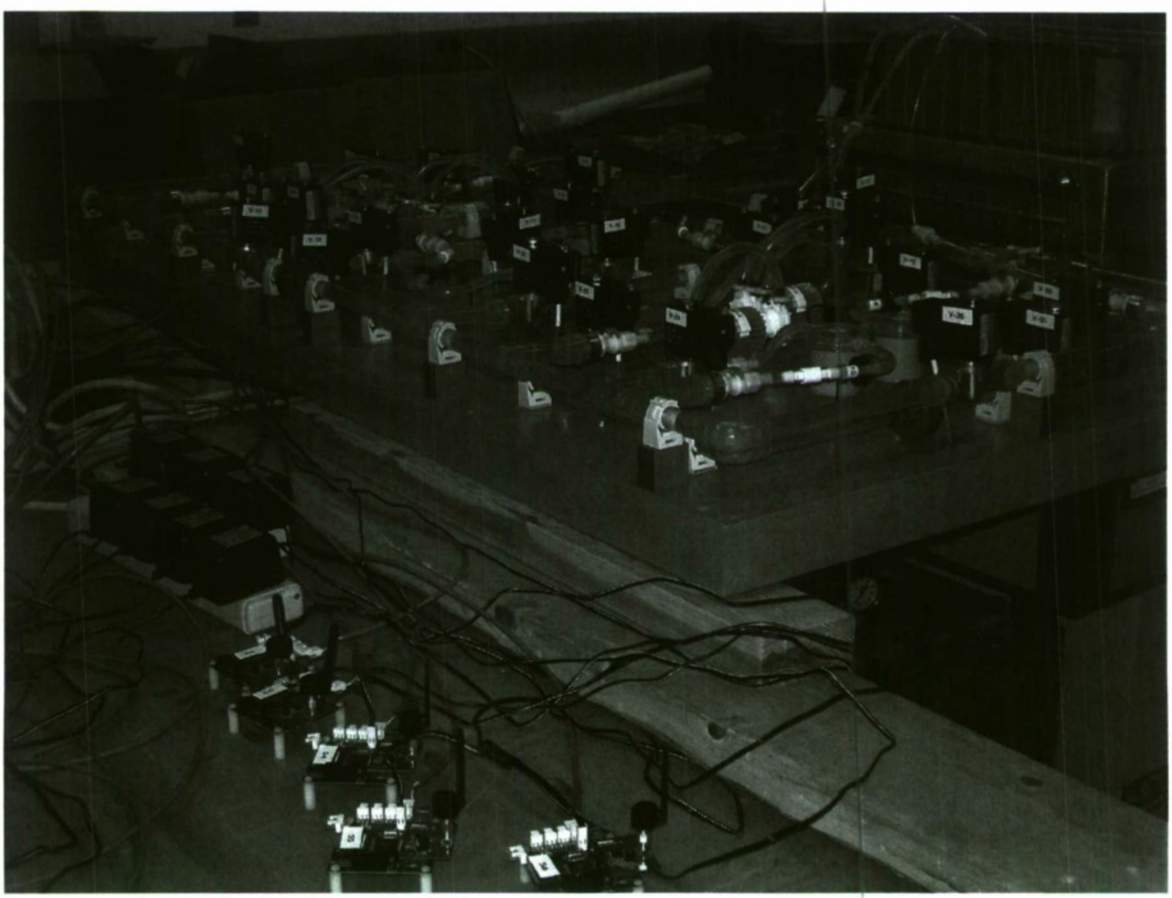


Figure 5.6. Narada wireless sensors set to monitor tabletop flowrates.

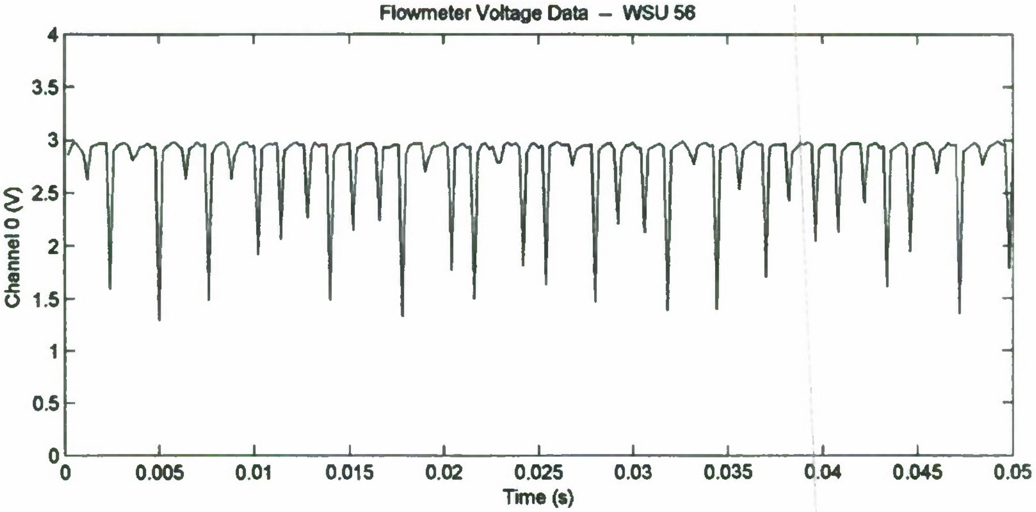


Figure 5.7. Sample noisy flowmeter voltage data, collected with a Narada wireless sensor.

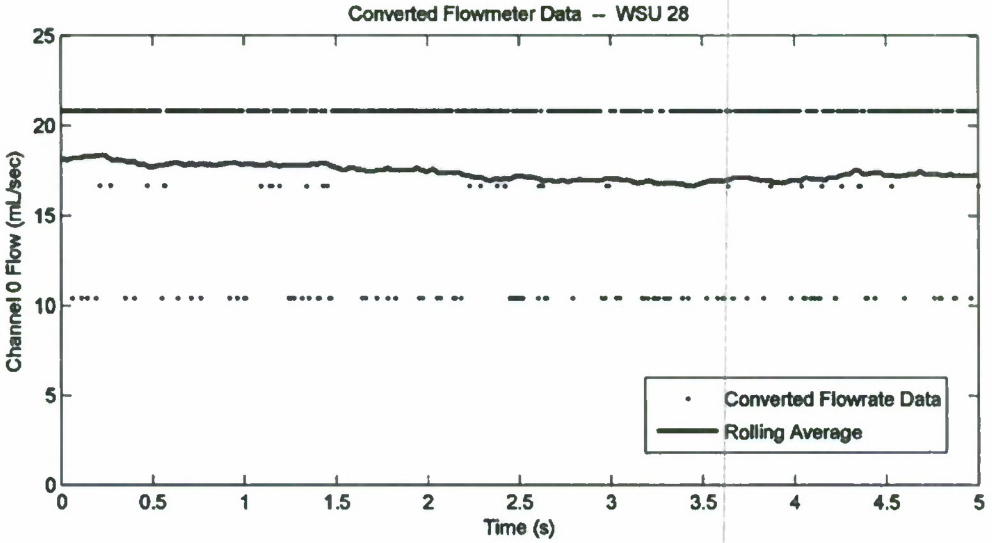


Figure 5.8. Sample converted flowmeter data, collected and converted with a Narada wireless sensor.

Beeause the rotor blades spin very fast (60,000 pulses per liter of passing water), a very high-frequeney square-wave output signal is ereated, even at the relatively low flowrates seen on the demonstrator (typieal 5-25 mL/see flowrates ereate square-wave frequeneies from 300-1500Hz). When these high-frequeney signals are eompounded with the faet that the square-wave has a very low duty eyele (<20%), it beeomes very diffieult to eapture the voltage drops that oeeur as the rotor blades spin. In faet, while the system is capable of measuring flowrates up to 27.7 mL/see, it was found that flow rates above 22 mL/see were often improperly reeorded when using the Narada system, even when sampled at the wireless sensor's maximum speed (10,000 Hz). An example Narada output from a high flowrate pipe is shown in Figure 5.9. It ean be seen that very few voltage drops are reeorded, when eontrasted with the lower flowrate output in Figure 5.5. It is unknown whether the underlying problem lies in insufficient sampling speeds, or whether the flowmeters themselves are simply not eapable of reading flowrates up to their doeumented maximum (75 mL/see). If it is a sampling issue, this problem eould be remedied in the future on the Narada system by utilizing a faster timing erystal and/or a faster mieroeontroller to eommand the high-speed analog to digital eonversion, by utilizing an additional mieroeontroller for the sole purpose of eonverting from analog to flow data, or by seleeting a different flowmeter.

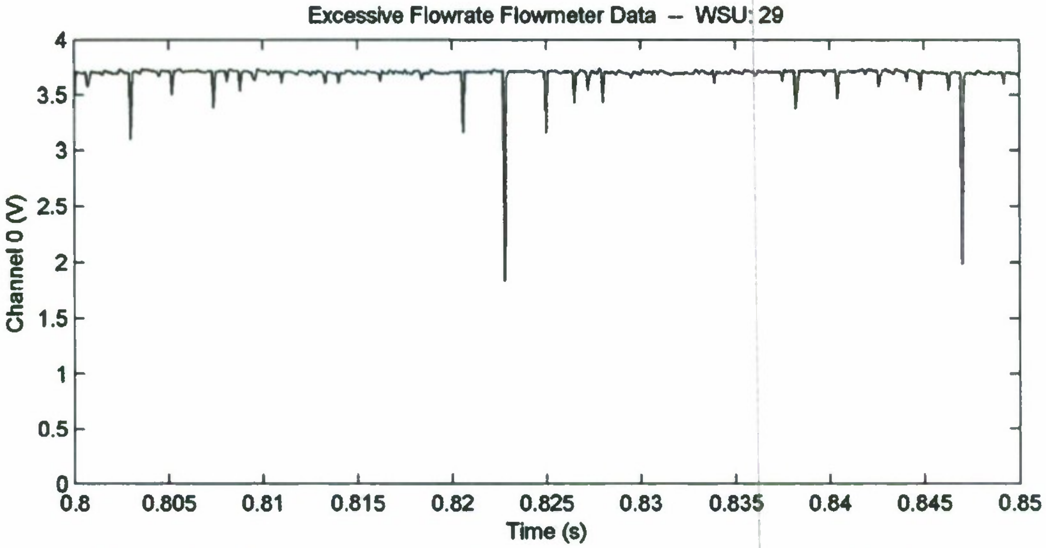


Figure 5.9. Excessive flowrate data (>25mL/sec) sampled at 10,000 Hz.

5.2 Phase II (Model Updating and Plant Damage) Field Work Statement:

This section of the report details the results of field testing performed on December 18th and 19th, 2008, at the Naval Surface Warfare Center in Philadelphia, PA. On these dates, the model updating and damage detection capabilities of the University of Michigan's Narada wireless sensing system were tested using the ONR tabletop chilled water demonstration system. The purpose of these field tests was to record a large amount of flow data from within a chilled water system when that system is subjected to several types of rupture and when that system is being driven by different pump configurations. This data was collected for the end purpose of developing effective model updating strategies for the detection of pipe rupture using networks of wireless sensing prototypes.

5.2.1. Phase II Objectives:

The objectives of the Phase II field work can be defined as follows:

1. Interface Narada wireless sensors with the flowmeters already installed on the ONR tabletop chilled water demonstration system using suggested improvements from Phase I. Also, interface Narada wireless sensors with new flowmeters installed at each rupture location.
2. Utilizing 9 different pump configurations, collect large amounts of full-system flow data from the tabletop demonstrator in varying states of damage (undamaged, rupture 1, rupture 2, rupture 1 and rupture 2).

5.2.2. Large Scale Data Collection for Model Updating and Plant Damage Detection:

When developing the model updating methodology presented in Section 4, it was found that in order to accurately match experimental results with analytical model projections, we need to collect data over several different input-output combinations. Because we have control over the pumps in the chilled water environment, it is decided to use different combinations of pumps to create multiple input-output scenarios. Each configuration is created by powering down a certain subset of pumps. The configurations used in this study are as follows (pump locations can be seen in Figure 5.1b):

Configuration #	Pumps
1	1,2,3,4
2	1,2,3
3	2,3,4

4	3,4
5	1,3
6	2,3
7	1
8	2
9	3

After implementing the changes suggested in Phase 1, a large amount of clean data was collected from the demonstrator. A typical example of flowmeter voltage data from a run in Pump Configuration #1 is shown in Figure 5.10. When this data is converted to flowrate using the technique developed in Phase I, a rolling average of flow data can be created. Over a sufficiently large sample of data, this average can be used as a representation of flow at that point in time. A visual example of the Narada's final flowrate output (sampled at 200 Hz) can be seen in Figure 5.11. The blue dots in this figure represent the converted and sampled flowrates and the red line represents the rolling average.

Table 5.1 contains the numerical representation of a subset the averaged flow data collected in Phase II. For each pump configuration, the flows recorded at each of the ten flowmeters (F1 through F10) can be seen. Flow meter locations can be found in Figure 5.1 (flowmeters are represented by pink circles).

Table 5.1. Example Phase II averaged flowrate data

	Pump Config	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
No Rupture	Configuration 1	13.37	14.14	21.2	20.35	13.73	13.88	18.67	18.89	12.41	11.41
	Configuration 2	10.53	11.28	24.23	23.13	10.5	11.09	21.68	0	9.54	8.89
	Configuration 3	9.8	10.88	0.58	23.37	10.5	10.47	21.66	22.37	9.63	8.73
	Configuration 5	6.94	7.51	27.59	0	7.09	7.53	24.43	0	6.58	5.62
	Configuration 7	4.06	4.27	30.34	0	3.75	4.22	0	0	3.46	2.4
	Configuration 8	3.41	4.67	0	26.96	3.67	4.09	0	0	3.36	2.34
Rupture 1	Configuration 1	13.48	14.34	21.52	20.23	13.85	14.06	18.93	19.36	12.45	11.48
	Configuration 2	10.61	11.42	24.52	23.21	10.59	11.27	22.05	0.17	9.51	8.9
	Configuration 3	9.94	11.1	0.53	23.33	10.75	10.81	21.88	22.75	9.74	8.6
	Configuration 5	7.09	7.55	27.89	0	7.13	7.66	24.92	0	6.59	5.74
	Configuration 7	4.03	4.18	30.41	0	3.67	4.11	0	0	3.37	2.38
	Configuration 8	3.5	4.78	0	26.83	3.76	4.16	0	0	3.4	2.45
Rupture 2	Configuration 1	13.43	14.39	21.18	19.89	13.73	13.92	18.93	19.57	12.35	11.33
	Configuration 2	10.49	11.34	24.42	22.93	10.56	11.06	21.99	0.2	9.44	8.74
	Configuration 3	9.88	11.03	0.51	23.07	10.63	10.6	21.8	22.75	9.63	8.66
	Configuration 5	7.01	7.58	27.66	0	7.31	7.76	24.82	0.04	6.73	5.26
	Configuration 7	4.12	4.33	30.24	0	3.84	4.33	0	0	3.55	2.33
	Configuration 8	3.33	4.61	0	27.24	3.65	4.09	0	0	3.36	2.16
Ruptures 1 & 2	Configuration 1	13.48	14.38	21.55	20.23	13.78	14.03	19.04	19.6	12.63	11.52
	Configuration 2	10.48	11.24	24.79	23.39	10.51	11.13	22.09	0.18	9.62	8.77
	Configuration 3	9.97	11.01	0.48	23.31	10.69	10.73	21.92	22.89	9.88	8.81
	Configuration 5	7.11	7.53	27.85	0	7.22	7.68	25.02	0	6.83	5.73
	Configuration 7	4.06	4.23	30.33	0	3.66	4.12	0.85	0	3.46	2.33
	Configuration 8	3.33	4.65	0	27.18	3.59	3.97	0	0	3.39	2.25

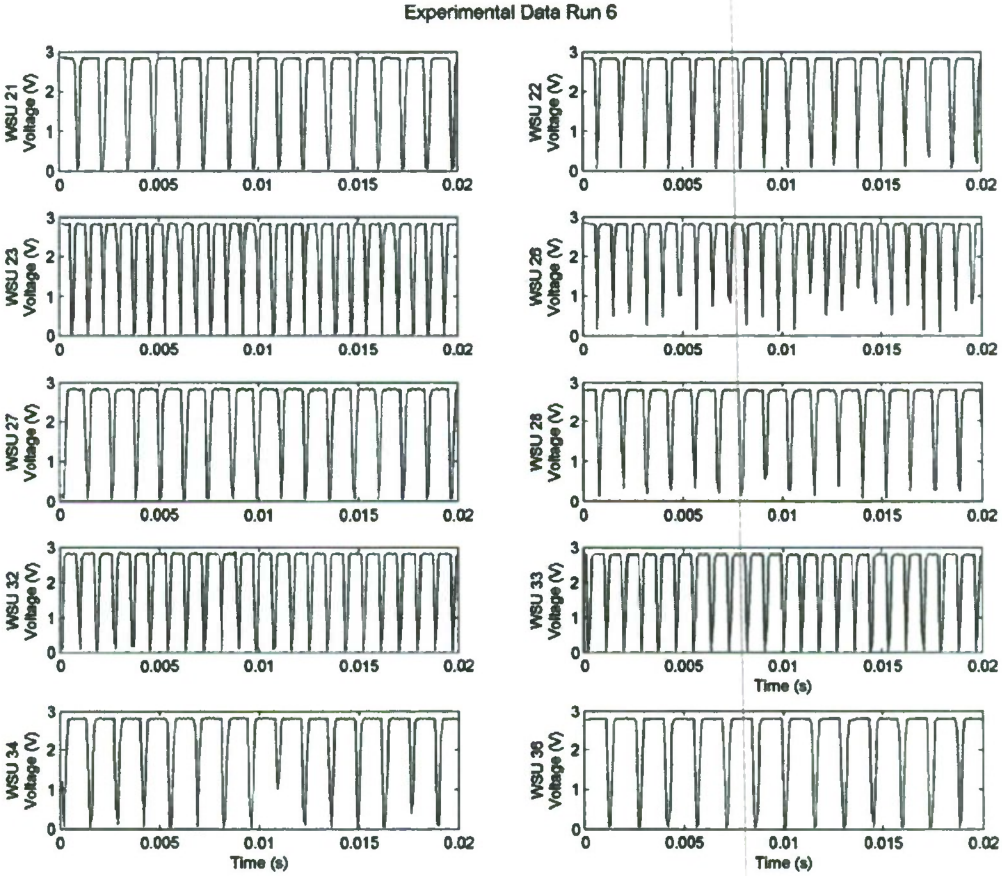


Figure 5.10. Typical Phase II flowmeter voltage data, sampled at 10,000 Hz.

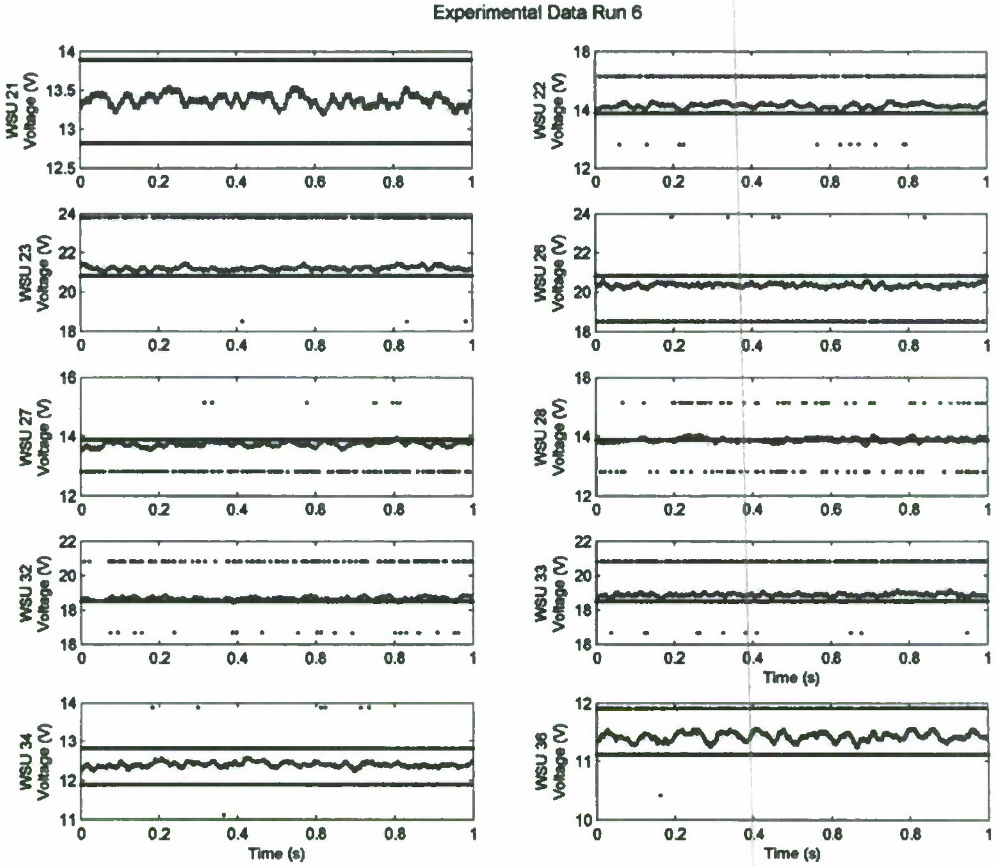


Figure 5.11. Typical Phase II converted flowrate data (blue) and rolling average (red), sampled at 200 Hz.

6. Model Updating of a Chilled Water Plant with Rupture Conditions:

In a naval environment, the ability to detect distinct changes in the operational condition of a engineering plant is extremely desirable. In this study, a network of wireless sensors is used in conjunction with an analytical model of a pipe network in order to detect damage and/or system bifurcations in a shipboard chilled water system. To this end, model updating methods have been developed and embedded within the computational core of the Narada wireless sensing unit. These methods function by "updating" unknown system parameters within an analytical system model such that they match experimental data collected by the wireless sensors. Over time, if and when these updated analytical paramcters change, they can serve as indicators of the severity and location of any acquired system damage.

For the purpose of this study, an analytical model of a pipe network is created that can be used to determine flowrate information in each pipe, given a certain pipe and pump configuration. This analytical model is designed using the linear theory method as proposed by Jeppson (1976). Essentially, this method is used to satisfy a system of equations where an independent continuity equation ($\Sigma Q_i = 0$) is written at each pipe junction and a head loss equation ($\Sigma K_i Q_i = \Delta H$) is written around each loop in the network. Additionally, equations relating pump flow rate and pump head loss are written for each pump. Using an iterative convergence process, a set of system flowrates can be settled upon that satisfy head loss, pump, and flowrate continuity requirements.

To validate the ability of this model updating system to detect and locate damage in a chilled water system, the ONR tabletop chilled water system demonstrator is chosen as a viable testbed. This system contains 42 pipes, 4 pumps, 6 loads, 26 valves, and 2 rupture valves. Each of the 10 pipes containing either a load or a pump is monitored with a flowmeter. Because of the ability of the demonstrator to simulate pipe rupture through the use of two rupture valves, pipe rupture is chosen as the primary damage mechanism for investigation. As such, a properly functioning model updating system will be capable of detecting when one (or both) of the rupture valves is open, and how much water is flowing through each of them given a set of flowmeter data. A schematic drawing of the ONR tabletop demonstrator can be found in Figure 6.1.

6.1. Examination of the Proposed Analytical Model:

In the initial analytical model used in this study to represent the flow distribution throughout an interconnected network of pipes, the Darcy-Weisbach equation is utilized to determine head loss in a pipe with a given length, diameter, flowrate, and friction factor. The Darcy-Weisbach equation is the most accurate known method for modeling flow through a pipe, but it relies on the calculation of friction factor from a set of equations that are dependent on the type of flow being seen in the pipe. As such, these equations are highly accurate when dealing with flows that are either fully laminar or fully turbulent. Unfortunately, because of the scale of the ONR tabletop chilled water system, most of the pipes in this network fall in the transition range between laminar and turbulent flow. Therefore, it is found that the Darcy-Weisbach equation does not produce consistent results across the board, and in many cases it may predict flowrates in some pipes that are orders of magnitude larger than what is seen in reality.

As a result, it is decided to utilize the Hazen-Williams equation in lieu of the Darcy-Weisbach equation for the model updating work performed in this study. The Hazen-Williams equation presents a much more simplistic (and therefore less accurate) representation of the relationship between flowrate and head loss in a given pipe, but it is linear throughout the transition region between laminar and turbulent flow, and thus allows us to look more confidently at systems where some pipes are laminar, some are turbulent, and some are in-between. In the model updating case, where we are much more interested in system changes over time than in the correlation between a given model and reality, the Hazen-Williams equation can be an excellent choice.

6.2. Model Updating using Pipe Diameters:

In the initial model updating method employed in this study, pipe damage is designed to be associated directly with pipe diameter. In this methodology, the diameter of each pipe is treated as an unknown, constrained by a range defined between 0% and 200% of the pipe's actual diameter. Then, these unknowns are utilized as the updating parameters in the model updating problem, and a search is launched to find the set of 42 pipe diameters which produce a set of analytical system flowrates that match most closely the 10 experimental flowrates measured in the tabletop demonstrator. The thought behind this methodology is that, in the undamaged system, the set of diameters that the model updating algorithm converges upon should be more or less identical to the actual pipe diameters in the physical system. Then, as rupture (or blockage) is introduced in a given pipe, a new model update should show an increase (or decrease) in updated pipe diameter for that given pipe.

It is quickly found that by using this methodology, the model updating algorithm is capable of finding a set of pipe diameters that produced analytical flowrates exactly equivalent to those measured experimentally. Unfortunately, the updated set of pipe diameters does not reflect the physical system in two very distinct ways:

1. There is no correlation found between updated pipe diameter and physical pipe diameter.
2. Successive model updates, while always effectively finding a system that predicts analytical flowrates equivalent to experimental flowrate measurements, returns vastly different sets of updated pipe diameters each time the algorithm is run.

As such, this methodology is found to be worthless from the damage detection perspective, as there (1) is no way to directly correlate the analytical model (pipe diameters) to the physical system, and (2) there is a seemingly infinite number of solutions to the pipe diameter model updating problem.

6.3. Model Updating using Hazen-Williams Coefficients:

In an attempt to improve upon the initial updating methodology, it is decided to try and correlate pipe damage not to a pipe's diameter, but to a pipe's Hazen-Williams coefficient (C_{HW}). The thought behind this methodology is that because the Hazen-Williams coefficient is tied directly to a pipe's frictional properties (and thus its influence on head loss), it may better reflect system changes in the case of a pipe rupture or blockage. Using this methodology, the Hazen-Williams coefficient of each pipe is treated as an unknown model updating parameter, constrained by a range defined between 1 (very rough pipe) and

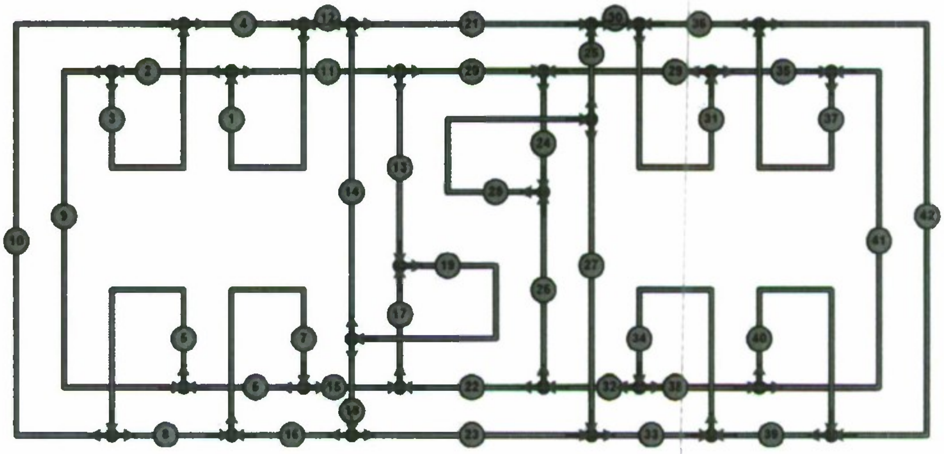


Figure 6.1. Pipe numbering and flow direction schematic for chilled water system with rupture valves closed.

1000 (very smooth pipe). Then, a search is launched to find the set of 42 coefficients which produce a set of analytical system flowrates that match most closely the 10 experimental flowrates measured in the tabletop demonstrator. The thought behind this methodology is that, in the undamaged system, the set of coefficients that the model updating algorithm converges upon should correlate to the actual frictional capacity of each pipe in the physical system. Then, as a rupture (or blockage) is introduced in a given pipe, a new model update should show a respective increase (or decrease) in the frictional coefficient for that pipe.

At first glance, it seems as if this methodology is preferable to the updating of pipe diameters, as Hazen-Williams coefficients are more intuitively tied to the physical actuality of friction change associated with pipe rupture or pipe blockage. Unfortunately, this methodology is hampered by the same problems found when using pipe diameters as an updating parameter. Specifically, there is a seemingly infinite number of solutions to the model updating problem, making damage detection very difficult, if not impossible.

6.4. Dimensionality Reduction of the Model Updating Problem:

One of the ways in which the problem of infinite solutions makes itself most obvious is that after a completed model update, two pipes which have the same length, diameter, and internal frictional losses (bends/tees/valves) may be found to have vastly different C_{HW} values. For example, pipes 9, 10, 41, and 42 (found in Figure 4) should have very similar frictional properties, and thus very similar C_{HW} values. However, it is found solutions obtained using the model updating method are not constrained in this way. As such, it is decided to pursue a model update containing only 12 different Hazen-Williams coefficients. One coefficient is assigned to each of the ten pipes containing either a pump or a load, one is assigned to each of the four pipes with two bends (9, 10, 41, and 42), and one is assigned to the remainder of the pipes, each of which has a three-way tee at each end and no bends.

Immediately after applying this dimensionality reduction principle to the model updating problem, a vast improvement can be seen in the quality of the final results. Not only does an exact match between analytical and experimental flowrates continue to be reached, but the resulting values begin to show a strong correlation with what would be expected from the actual physical system. For example, the C_{HW} value in the smaller pipes (1, 3, 5, 7, 19, 28, 31, 34, 37, 40), each of which contain several obstructions, is significantly lower (designating more friction) than the C_{HW} values found for the larger, unobstructed pipes. Additionally, a marked difference is seen between the C_{HW} values in the four pipes containing pumps (1, 7, 31, 34) than in those with loads. Overall, pipes that should have similar properties do have similar properties

Unfortunately, while this dimensionality reduction seems to improve the tie between the physical system and the model-updated results, variability in the results between model updating runs leads to the belief that the problem of infinite solutions is still present in the system.

6.5. Simultaneous Updating of Multiple Input-Output Relations:

Another of the ways in which the problem of infinite solutions manifests itself in the damage detection architecture is by introducing solutions (i.e. sets of Hazen-Williams coefficients) that may produce accurate model predictions for one particular input (i.e. all four pumps engaged), but vastly incorrect model predictions for another input (i.e. only three pumps engaged). As such, it is useful to use multiple sets of flowmeter data, collected using different input parameters (in this case, number of pumps engaged) to update several input-output relations simultaneously. In other words, by utilizing two or more different sets of input-output data from the same undamaged system, we can update an analytical model for each type of input using one unique set of the twelve updating parameters discussed in the previous section. Hopefully, this method will decrease significantly the number of possible C_{HW} combinations that produce accurate matches between analytical and experimental results.

By simultaneously updating multiple analytical models of different input scenarios, it is found that while each additional model slightly degrades the accuracy of the match between experimental measurements and analytical predictions (likely due to experimental error or model imperfection), the updated set of Hazen-Williams coefficients becomes largely unique. For example, when the sets of data generated with three pumps and four pumps are updated simultaneously, a unique set of 10 of the 12 updated Hazen-Williams coefficients can be found (those corresponding to the monitored pipes) that predicts flowrates to within 0.01 mL/sec of all recorded measurements. However, the two Hazen-Williams coefficients that are applied to the remainder of the pipes in the system continue to fluctuate, leaving a still infinite number of possible system configurations.

6.6. Addition of One Flowmeter to Ensure Asymmetric Monitoring:

Upon examining why only the Hazen-Williams coefficients associated with monitored pipes could be determined using the previous approach with multiple input-output relations, it became apparent that knowing full information about these ten pipes did not necessarily imply anything about the remainder of the pipe network. In other words, even if flow rates and head losses in these monitored pipes are fixed, the flow rates and head losses in the surrounding pipes can fluctuate at will, due to the symmetrical nature of the pipe network and the centralized positioning of the existing flowmeters. As such, it is proposed that an additional flowmeter be installed in one of main pipes used to connect parts of the system (i.e. pipe 20, 21, 22, or 23). Potentially, this will force the remainder of the system into one unique flow configuration, and as such, we should be able to detect one unique set of Hazen-Williams coefficients.

After updating the same set of models used in the previous section, it is found that the addition of one more flowmeter, installed in pipe 21, does in fact allow the model updating process to detect one unique set of Hazen-Williams coefficients for a given system state. The unique nature of this solution is conducive to damage detection applications, where a solid baseline value is necessary in order to effectively look for changes in system properties.

6.7. Model Updating for Pipe Rupture Detection and Quantification:

Having discovered a method of effectively determining system-wide properties through model updating (by adding an additional flowmeter, using dimensionality reduction and by simultaneous updating multiple input-output scenarios), it is now possible to look into the detection and location of damage (pipe rupture) within the tabletop demonstrator. In order to most accurately model the experimental situation present in the physical tabletop demonstrator, it is decided to model the ruptures exactly as they exist in reality. In other words, a pipe is added at the location of each rupture valve which joins the main demonstrator pipe network to the reservoir which provides water to the system. Thus, in order to accommodate the rupture valves and the reservoir, the number of pipes is increased from 42 to 47. This new analytical model can be seen in Figure 6.2. Ideally, an additional flowmeter would be placed in pipe 21, but due to demonstrator restrictions, this addition was physically impossible.

Given the analytical model proposed above, it is postulated that rupture can be detected and localized (given a previous knowledge of the two rupture locations) by using the following procedure:

1. Collect flowmeter data in the undamaged (i.e. rupture valves closed) case for two or more different input scenarios (created by varying the pump configuration).
2. Simultaneously update all 12 unknown C_{HW} values using the analytical models corresponding to the varying pump configurations, while forcing C_{HW} in pipes 44 and 46 to be equal to 0.01. (This simulates near-infinite friction, and thus a flow situation where both rupture valves are closed. A C_{HW} of 0 cannot be used because of divide-by-zero issues.) The resulting updated C_{HW} values represent the baseline, undamaged state of the demonstrator.
3. Collect flowmeter data in all three undamaged states (i.e. one or both rupture valves open) for each of the pump configurations.

4. For each damage state, simultaneously update each of the appropriate analytical models (corresponding to pump configuration) using the C_{HW} value in pipes 44 and 46 as the updating parameters. The baseline, undamaged C_{HW} values found in step 2 are utilized for all other pipes. The resulting updated C_{HW} values for pipes 44 and 46 represent the ruptured state of the demonstrator. The larger these values are relative to the baseline case (Step 2), the greater the degree of rupture in the associated location.

6.8 Experimental Model Updating Results for Pipe Rupture Detection and Quantification:
As described in Section 5, during Phase II experimental testing a large set of data is collected using a wide range of pump configurations to drive the system. While all nine pump configurations were tested, it was found that four of the configurations were better suited to driving water into the open rupture valves. As such, it was decided to only use these four pump configurations (1, 2, 3, and 5) for the purpose of model updating-based damage detection. In each of these pump configurations, data was collected for the undamaged system, the system with rupture valve 1 open, the system with rupture valve 2 open, and the system with both rupture valves open.

Before any experimental data is used to test the damage detection strategy proposed herein, it is decided to utilize a set of simulated data derived from the analytical model seen in Figure 6.2 and explained above. As such, the model is driven with the same set of four pump configurations, and simulated data is collected in the undamaged system, the system with rupture valve 1 open, the system with rupture valve 2 open, and the system with both rupture valves open. While several different objective functions were effectively utilized, it was found that the most effective objective function consisted of the average analytical difference between analytical model flowrates and experimental flowmeter data (essentially Equation 4.3).

Using the four different input cases generated by varying the pump configuration, it is found that the proposed model updating procedure is capable of not only detecting, but also quantifying (to an extent) rupture in the two locations targeted in this study. Cases are tested where rupture valve 1 is open, where rupture valve 2 is open, and where both rupture valves are open. In all cases, the model updating algorithm is capable of detecting damage. However, the degree to which that damage can be located and quantified varied from case to case.

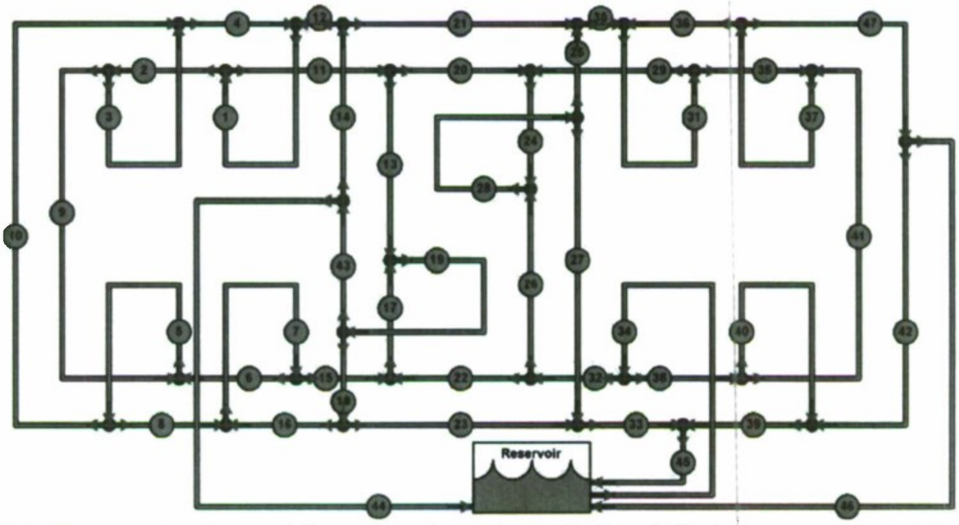


Figure 6.2. Pipe numbering and flow direction schematic for chilled water system with rupture valves open.

Beginning with the simulated set of data, the 12 unknown C_{HW} values are updated in an effort to find an analytical system that produces flowrate data that is, on average, as close as possible to the simulated data from the undamaged chilled water system. The C_{HW} values for pipes 44 and 46 are set to 0.01 as explained above. As seen in Figure 6.3, the model updating procedure is able to find a set of C_{HW} values that create analytical flowrates very close to those produced in the (undamaged) simulated system (within 1% error). These baseline values are then used to update the possible damage locations (rupture 1 and rupture 2) in each of the four possible damage scenarios. It can be seen in Figure 6.4, Figure 6.5, Figure 6.6, and Figure 6.7 that C_{HW} values for pipes 44 and 46 can be found that produce analytical flowrates very close to those found in the corresponding simulated systems (within 2% error in all cases). When looking at how the updated C_{HW} values for pipes 44 and 46 relate to their respective damage cases (Figure 6.8), it can be clearly seen that the damaged scenarios (where one or both of the rupture valves are open) show very distinctive signs of damage when compared with the undamaged scenario. However, while there does seem to be a correlation between which the physical and the updated rupture locations (C_{HW} values are higher for the physical rupture locations), the fact that the model updating procedure diagnoses some degree of rupture in both pipes for all damage scenarios (even when only one rupture location exists), this method appears questionable for locating and quantifying damage within a pipe network.

Moving to the set of experimental data, the 12 unknown C_{HW} values are again updated in an effort to find an analytical system that produces flowrate data that is, on average, as close as possible to the experimental data from the undamaged chilled water system. The C_{HW} values for pipes 44 and 46 are set to 0.01 as before. As seen in Figure 6.9, the model updating procedure is able to find a set of C_{HW} values that create analytical flowrates reasonably close to those produced in the (undamaged) experimental system (within 20% error in all cases and much closer in most cases). These baseline values are then used to update the possible damage locations (rupture 1 and rupture 2) in each of the four possible damage scenarios. It can be seen in Figure 6.10, Figure 6.11, Figure 6.12, and Figure 6.13 that C_{HW} values for pipes 44 and 46 can be found that produce analytical flowrates very close to those found in the corresponding simulated systems (again, within 20% error in all cases and much closer in most cases). When looking at how the updated C_{HW} values for pipes 44 and 46 relate to their respective damage cases (Figure 6.14), it can be seen (although somewhat less clearly than in the case of simulated data),that the damaged scenarios (where one or both of the rupture valves are open) show some signs of damage when compared with the undamaged scenario. There even seems to be some degree of correlation between which the physical and the updated rupture locations (C_{HW} values are higher for the physical rupture locations), especially in the case of rupture 1 and ruptures 1 & 2. However, the fact that the model updating procedure does not diagnose a greater degree of rupture at rupture location 2 for the case of rupture 2, this method (as with the simulated data) appears questionable for locating and quantifying damage within a pipe network.

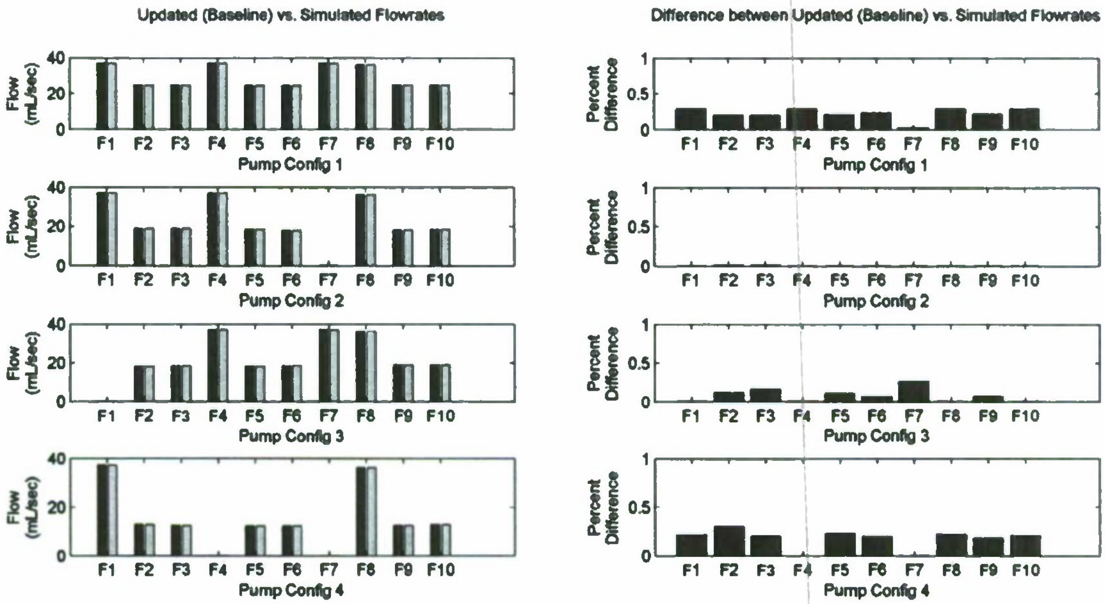


Figure 6.3. Model updating accuracy for a baseline update using simulated data.

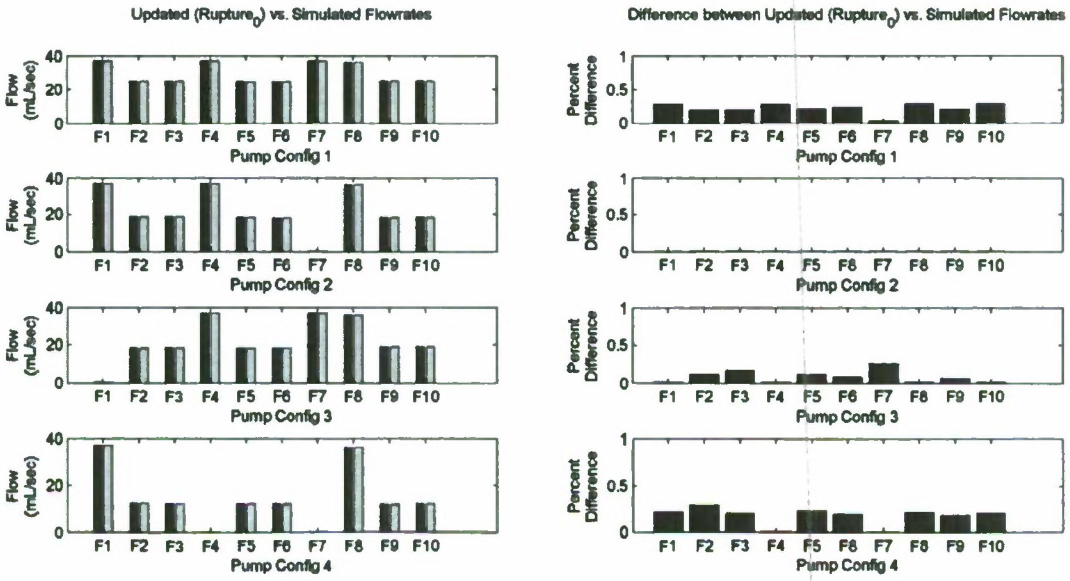


Figure 6.4. Model updating accuracy for an unruptured update using simulated data.

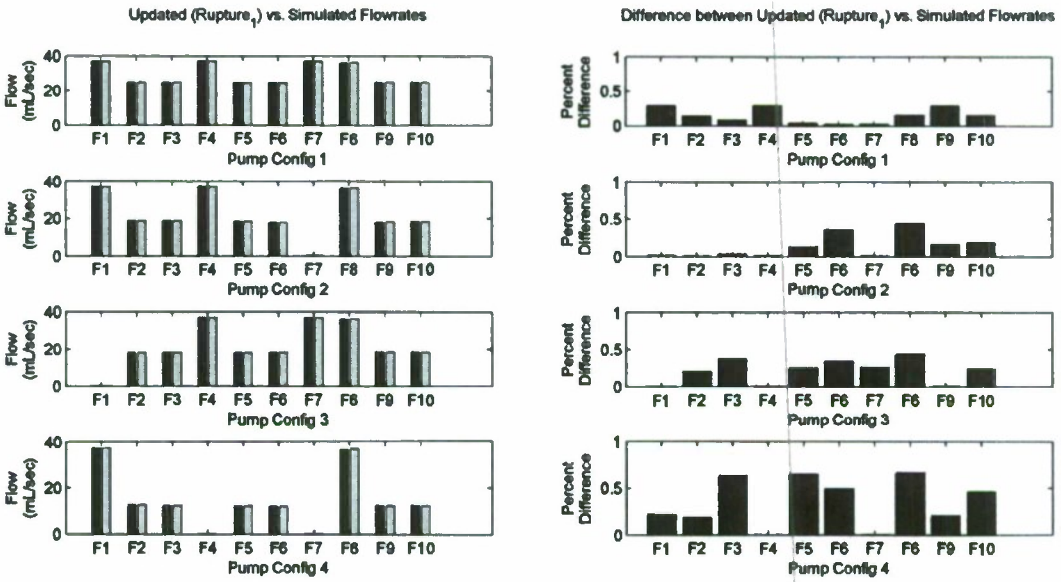


Figure 6.5. Model updating accuracy for a rupture case 1 update using simulated data.

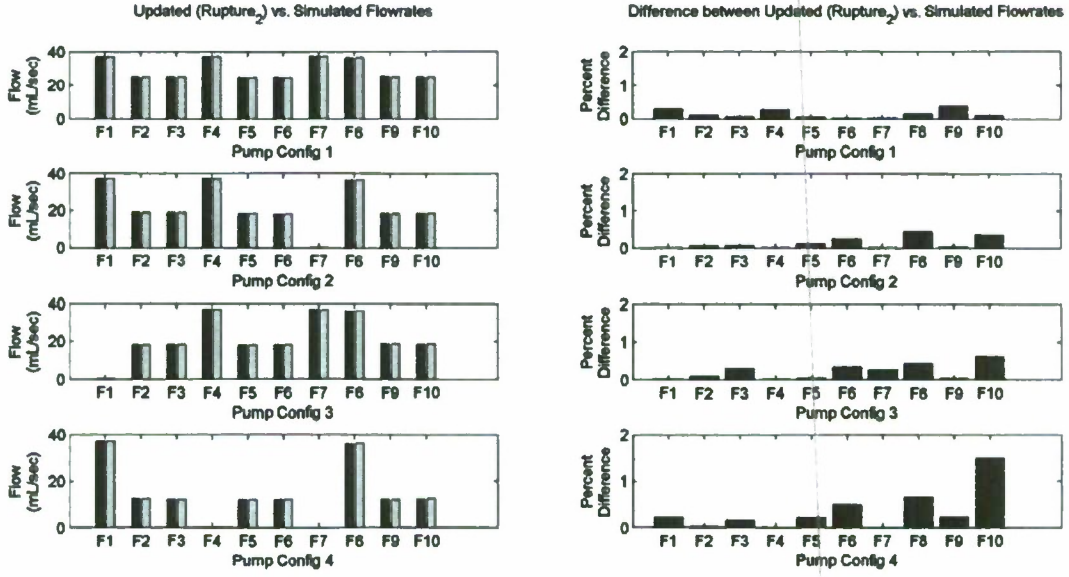


Figure 6.6. Model updating accuracy for a rupture case 2 update using simulated data.

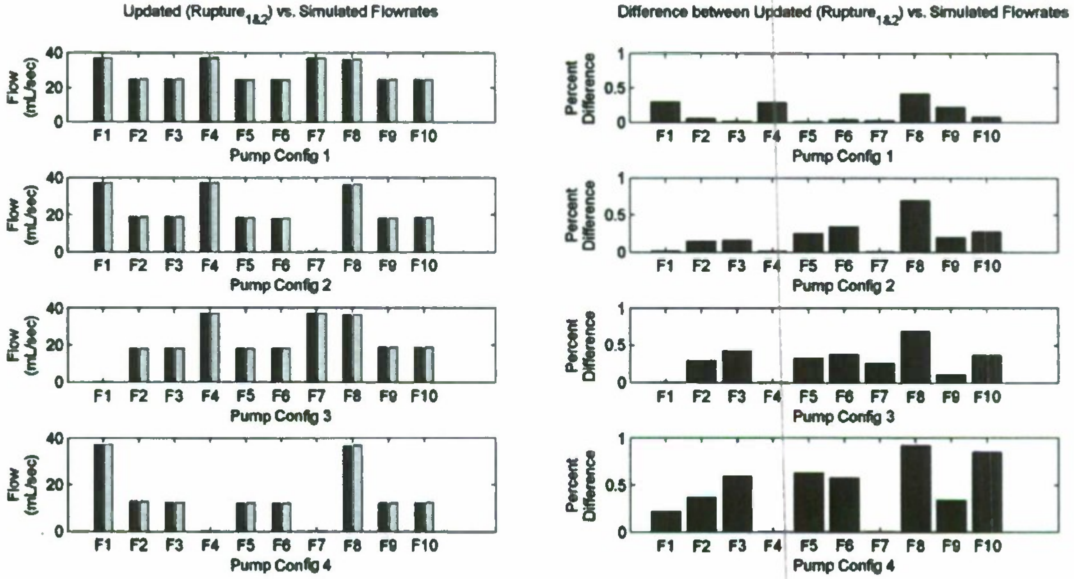


Figure 6.7. Model updating accuracy for a rupture case 1&2 update using simulated data.

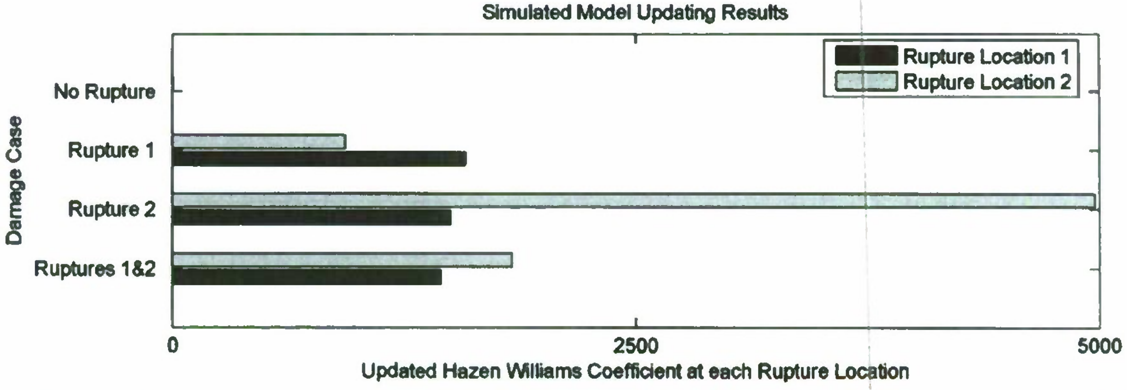


Figure 6.8. Model updating damage detection results for all damage cases using simulated data.

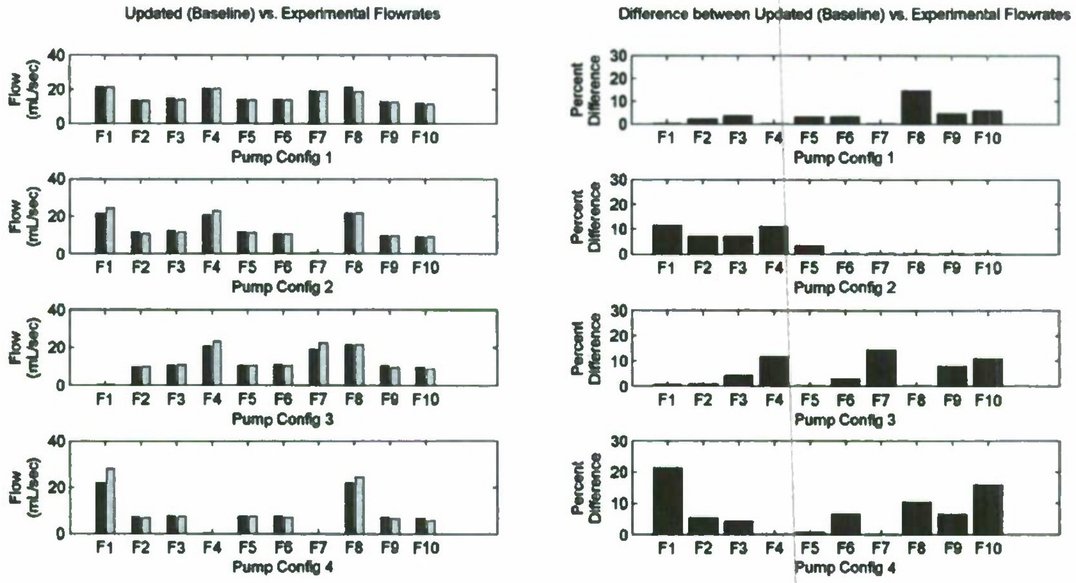


Figure 6.9. Model updating accuracy for a baseline update using experimental data.

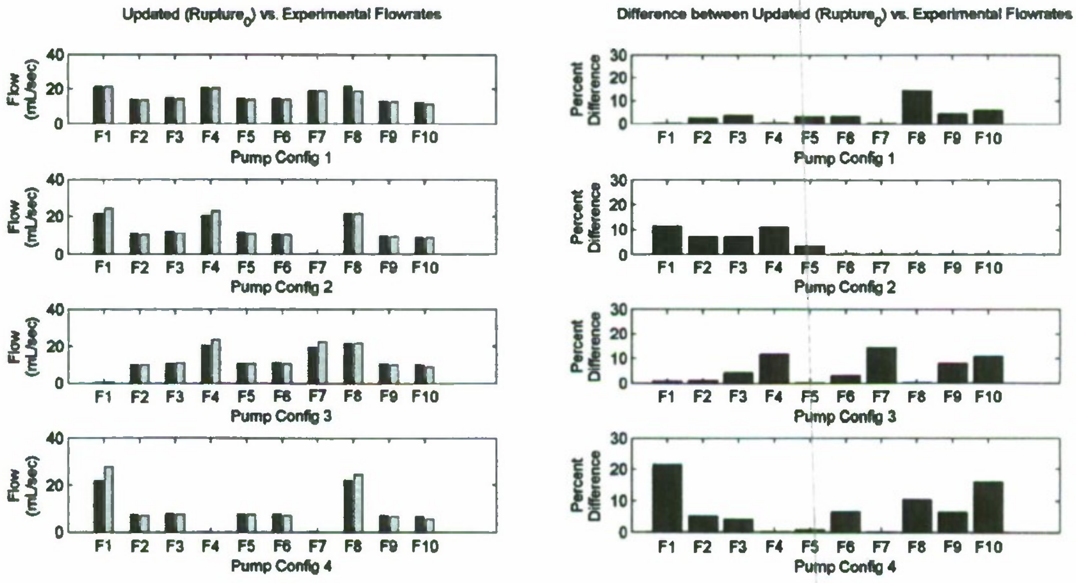
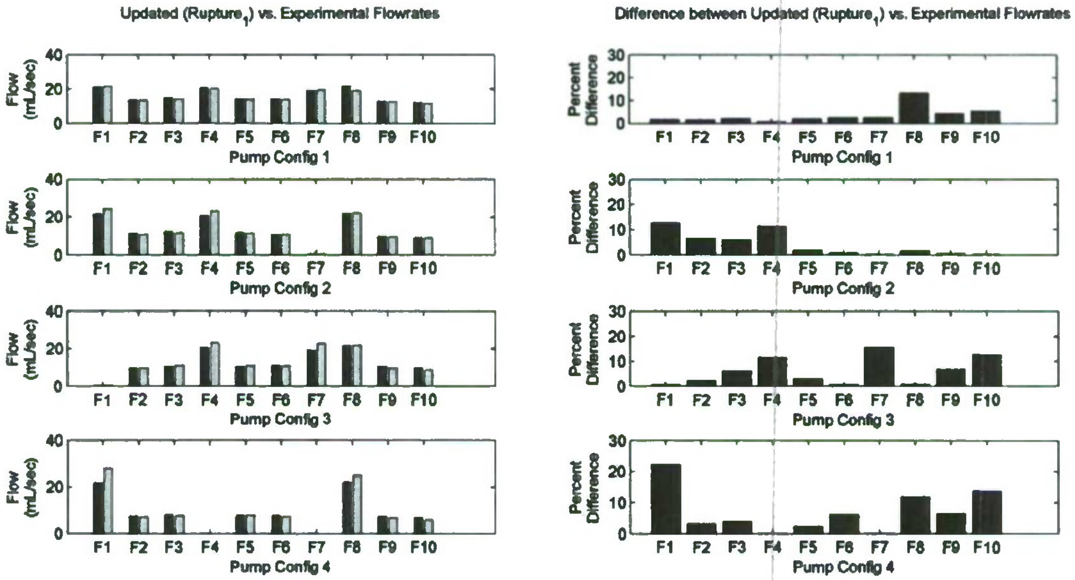
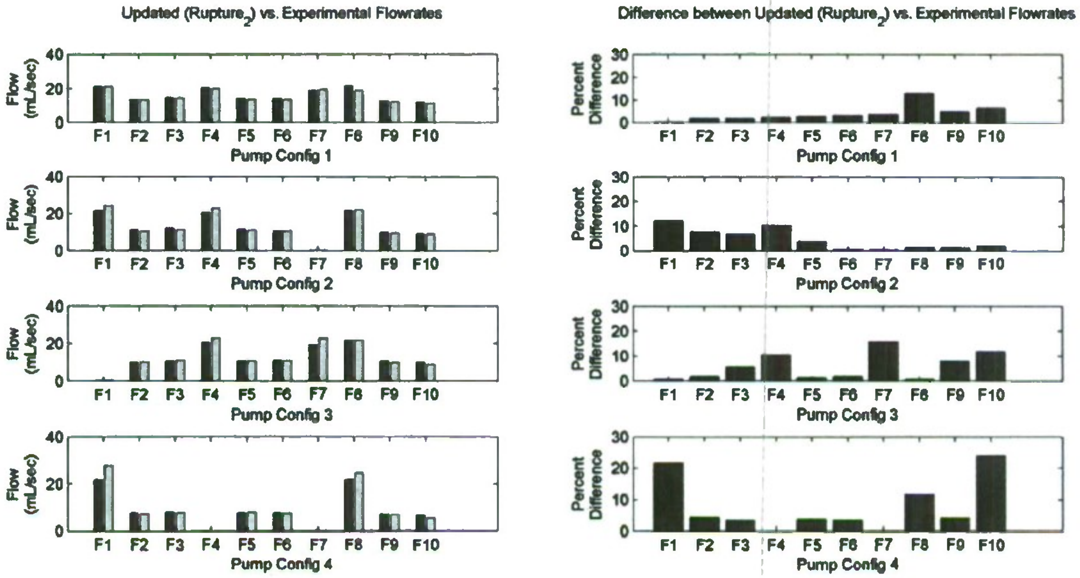


Figure 6.10. Model updating accuracy for an unruptured update using experimental data.



Figurc 6.11. Modcl updating accuracy for a rupture casc 1 updatc using cxpcrimcntal data.



Figurc 6.12. Modcl updating accuracy for a rupture casc 2 updatc using cxpcrimcntal data.

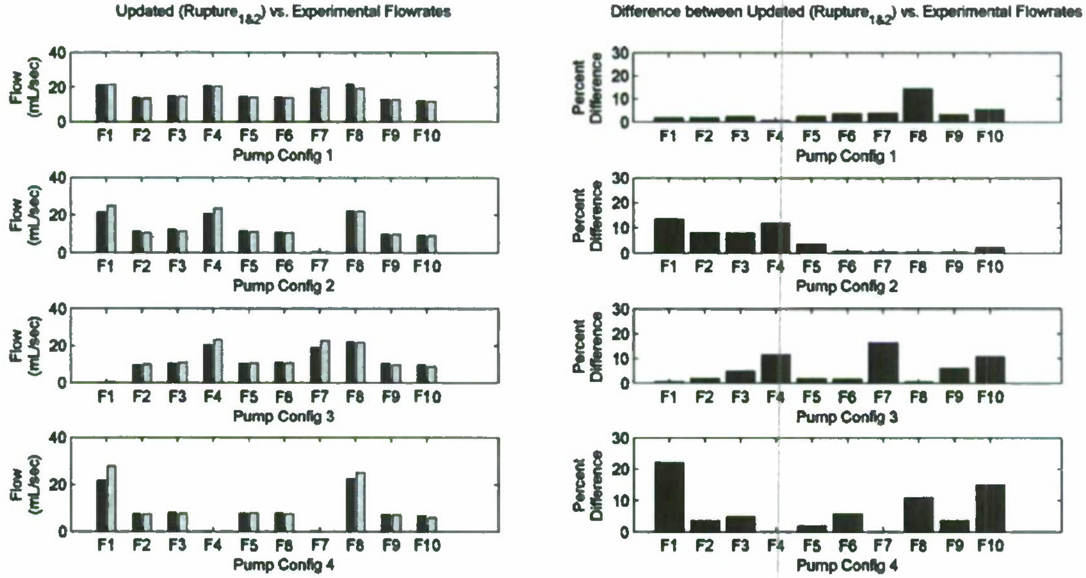


Figure 6.13. Model updating accuracy for a rupture case 1&2 update using experimental data.

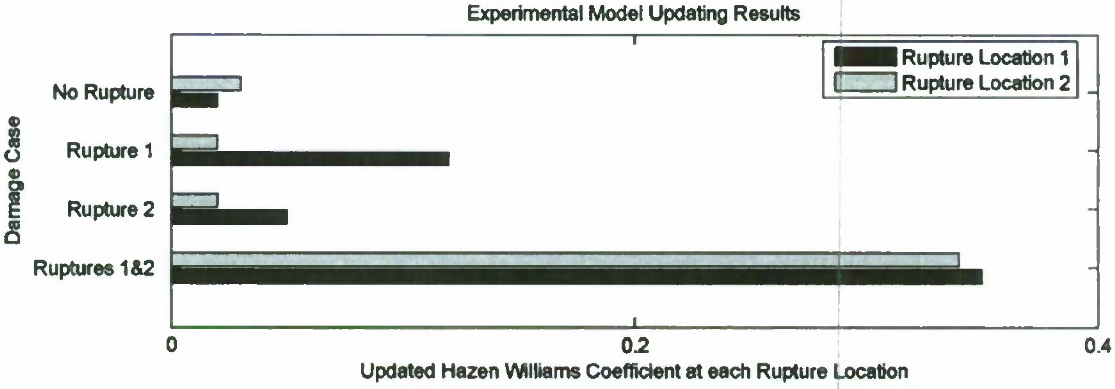


Figure 6.14. Model updating damage detection results for all damage cases using experimental data.

7. Market-based computational task assignment within a WSN:

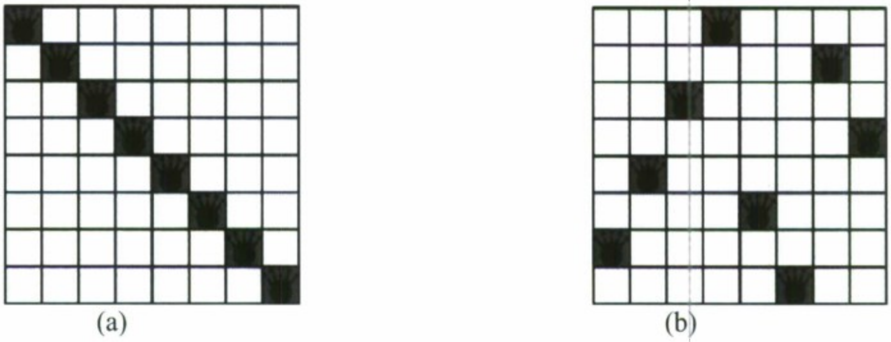
Having developed a framework in which a wireless sensing network can execute complex engineering analyses (such as model updating) in a distributed fashion, the next logical step is to expand this computing paradigm to allow a sensor network to perform multiple computational tasks simultaneously. However, one of the key challenges inherent in the simultaneous deployment of multiple distributed algorithms on a wireless sensing network is that in the wireless environment, many of the system resources (such as battery power, data storage capacity, MPU time, wireless bandwidth, etc.) required to perform complex computational tasks are available only in a limited manner. As such, it is important to devise a method of optimally distributing and consuming these scarce system resources throughout the network. One approach to this problem is to apply free-market economics to help allocate these resources. Free-market economies can be thought of as large collections of autonomous market agents (participants) such as producers (sellers) and consumers (buyers), among others. In this free-market framework, each agent is forced to compete against other agents in a competitive marketplace with scarce resources.

In the case of distributing computational tasks throughout wireless sensing networks, a market can be formed where the only service for sale is a block of CPU time which can be devoted to any computational task. For our purposes, we will deal with a network in which each computational task is a simulated annealing search problem of complexity C_{SA} that can be broken into separate blocks of N_{SA} SA search iterations. At any given instant in time, a certain number of processors, P_{SA}, are already associated with each task, and each task has already seen search progress made at SA temperature steps up to and including TS_{SA}. In order to provide a viable, robust testbed in which to work out and validate the market-based concepts proposed herein, the *n*-Queens problem is chosen as a simple, easy to implement combinatorial optimization problem that will fit nicely within the distributed model updating framework that was developed in Section 4 and applied to a chilled water system in Section 6.

7.1 The n-Queens problem:

The *n*-Queens problem is a well-known benchmark problem for evaluating the performance (*i.e.* speed and efficiency) of optimization or search algorithms developed in the computer science community. The objective of the *n*-Queens problem is to place *n* chess queens on an *n* x *n* chess board (where $n \geq 4$) such that no queen can attack another queen following basic chess rules. In other words, no queen can be placed on the same row, column, or diagonal as another queen. One of many solutions to the 8-Queens problem can be seen in Figure 7.1(b).

The *n*-Queens optimization problem proceeds by attempting to minimize an objective function, *E*, which sums the number of conflicts between queens in a given chess board configuration. In an analytical sense, if a queen is at position (I, J), it is in direct conflict with any queen at position (i, j), where $i = I$ (same column), or $j = J$ (same row), or $|i - I| = |j - J|$ (same diagonal). So, if we let q_{ij} represent each square on a chess board, and if we set q_{ij} equal to 1 if there is a queen at position (i, j), and 0 otherwise, we can create an appropriate objective function, *E*, as follows:



(a) (b)

Figure 7.1. (a) Initial board configuration for 8-Queens problem and (b) optimal solution for the 8-Queens problem

$$E=\sum_{I=0}^{n-1}\sum_{J=0}^{n-1}\begin{pmatrix} \displaystyle\sum_{k=J+1}^{n-1} q_{IJ}q_{Ik}+ \\ \displaystyle\sum_{k=I+1}^{n-1} q_{IJ}q_{kJ}+ \\ \displaystyle\sum_{k=1}^{\min(n-1-I,n-1-J)} q_{IJ}q_{(I+k)(J+k)}+ \\ \displaystyle\sum_{k=1}^{\min(n-1-I,J)} q_{IJ}q_{(I+k)(J-k)} \end{pmatrix} \qquad (7.1)$$

with the first term summing row conflicts, the second term summing column conflicts, the third term summing upper diagonal conflicts, and the fourth term summing lower diagonal conflicts. Each combination of squares q_{ij} and q_{IJ} returns 1 if there is a queen conflict and 0 if there is not, leading to a sum equal to the total number of conflicts. To minimize duplicate conflicts, each square on the chess board is evaluated only once against all other squares.

For the implementation of the n-Queens problem in this study, we choose to start with a board configuration such that a queen is placed on each diagonal square (i, j), *i.e.* where $i = j$, as seen in Figure 7.1(a). Clearly, in this initial state, each queen is in conflict with all other queens. New search states are generated by swapping the queens laying on two randomly selected rows, while retaining each queen's initial column. In this way, there is always one queen in each row and one queen in each column. This state generation method allows for significantly faster convergence of the optimization problem, as the first two terms of the objective function (Equation 7.1) can be ignored. The n-Queens problem is an ideal testbed in this study because it allows us to easily explore computational tasks of varying complexity by simply increasing the n-Queens problem size.

7.2 Market-based task assignment:

One incredibly complex system that is optimally controlled in a decentralized manner is that of the free-market economy, where scarce societal resources are distributed based on the local interactions of buyers and sellers who obey the laws of supply and demand. Recently, researchers have begun to utilize market-based concepts for the control or optimization of complex systems, most often in the realm of computer architecture where a market analogy is useful for modeling computer systems such as memory usage or network traffic. Perhaps the greatest benefit of market-based optimization is that can often yield a Pareto optimal solution. Pareto optimal can be defined by a market in a competitive equilibrium where no market

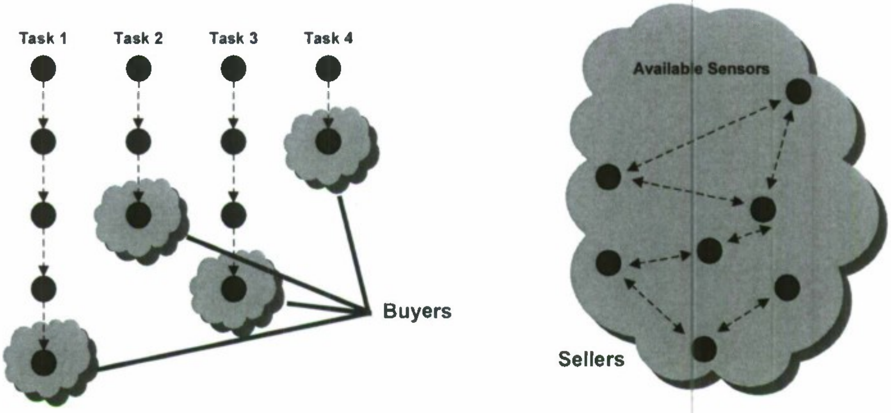


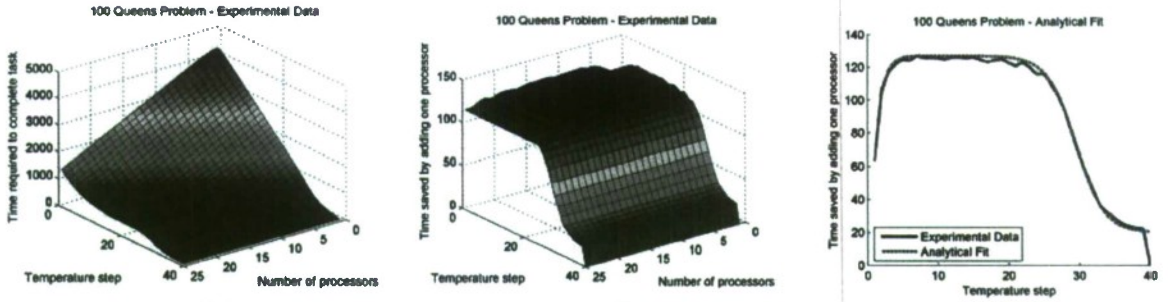
Figure 7.2. Buyer/seller distinction in the market-based task assignment model

participant can reap the benefits of higher utility or profits without causing harm to other participants when a resource allocation change is made.

In contrast to a simple auction-based system which could be used to crudely optimize the market based on only one goal, such as minimizing time to completion, a more robust market-based task assignment scheme can be developed that can optimally allocate resources in the midst of several market goals. This is accomplished through the use of buyer and seller "utilities". By embedding within each market agent the desire to optimize an individual utility function, competing goals can be settled through market means (supply, demand, price, etc). The result is a Pareto-optimal allocation of scare system resources. In this example, we are interested in three distinct (but possibly competing) performance objectives: (1) completing all required computational tasks as quickly as possible, (2) minimizing power consumed by the sensor network, and (3) functioning as robustly and as reliably as possible. In order to measure the ability of the market-based system to address these three competing performance objectives, several performance metrics can be created and utilized: (1) the time required to complete each task, (2) the number of wireless transmissions required to complete each task, and (3) the number of sensor and communication failures encountered during each task.

7.2.1 Buyer/seller framework:
As seen in Figure 7.2, the sellers in this market can be defined as the set of sensors in the wireless network not currently working on any computational task. These sensing units will be "selling" their computational abilities to a number of buyers, represented by the set of sensors most recently added to each existing computational task. For example, in Figure 7.2 there are four tasks, so there are four buyers. In order to address all three goals of this market in a streamlined manner, buyers and sellers are assigned different objectives. In this market, all buyers work both to minimize the overall time spent computing and to maximize network reliability by (1) minimizing CPU time required to complete each task, (2) minimizing CPU time lost due to sensor failure, and (3) minimizing CPU time lost due to communication failure. Sellers, on the other hand, work to minimize network power consumption. Because the wireless radio consumes significantly more power than any other sensor component, sellers aim to minimize power consumption by minimizing the number of wireless communications required to complete each task.

7.2.2 Utility function development:
In light of this framework, it is necessary to first outline the utility functions associated with both buyers and sellers. These utility functions will govern whether or not the buyer for each computational task will place a bid on the services of a given seller and which buyer, if any, a seller will sell its services to. On



(a) (b) (c)

Figure 7.3. (a) Experimental time data for each temperature step/processor combination, (b) graphical representation of experimentally determined T_{SPEED}, and (c) the accompanying analytical model fit

the buyer side, a utility function, U_{BUYER} can be intuitively thought of as the total amount of computational savings a computational task gains by adding an additional processing node, and can be defined as follows:

$$U_{BUYER} = T_{SPEED} - \alpha_{BUYER} \cdot T_{SENSOR\ FAIL} - \beta_{BUYER} \cdot T_{COMM\ FAIL} \tag{7.2}$$

where T_{SPEED}, $T_{SENSOR\ FAIL}$, $T_{COMM\ FAIL}$, α_{BUYER}, and β_{BUYER} are defined in detail below.

For any simulated annealing task, T_{SPEED} represents the decrease in expected time required to complete the task brought about by the addition of one processor. While there is no way of directly formulating an analytical expression for this value, a trend can be established for any computational task by looking at the average amount of time it takes a task of complexity C_{SA} to converge on a solution from a given temperature step, TS_{SA} with a given number of processors, P_{SA}. Using data collected over a large number of experimental trials, the time saved by adding an additional processor at any given search state can be expressed as the difference between the average time required to complete a search where P_{SA} nodes are currently searching up to temperature step TS_{SA} and the average time required to complete a search where P_{SA}+1 nodes are currently searching up to temperature step TS_{SA}+1. Figure 7.3(a) shows the time required to complete the 100-Queens problem from a given state (P_{SA}, TS_{SA}) in the search, and Figure 7.3(b) shows the amount of time saved by adding an additional processing node when the 100-Queens problem is in a given state (P_{SA}, TS_{SA}). Note that if all impossible states (where $P_{SA} > TS_{SA}$) are excluded, this relationship is independent of P_{SA}, and can be approximated by an easily computable algebraic function:

$$T_{SPEED}\left(\mathrm{C_{SA}}, \mathrm{TS_{SA}}\right) = \alpha + \frac{\beta - \alpha}{1 + e^{0.5 \cdot (TS_{SA} - \delta)}} - (\beta - \gamma) \cdot e^{-TS_{SA}} \tag{7.3}$$

where the values for α, β, δ, and γ are specific to each task complexity, C_{SA}, and are tabulated in Table 7.1. The quality of the analytical fit provided from this function for the 100-Queens problem can be seen in Figure 7.3(c). Fits of similar quality can be found for all other problem complexities considered (25-Queens, 50-Queens, and 75-Queens).

For any simulated annealing task, $T_{SENSOR\ FAIL}$ represents the increase in expected processing time lost due to sensor failure brought about by the addition of one processor. Unlike T_{SPEED}, this quantity can be derived analytically. Intuitively, if any sensor succumbs to either hardware or software failure while it is involved in a parallel SA task, all work done by the failed node, as well as all nodes below it would be lost. As such, $T_{SENSOR\ FAIL}$ can be expressed as the amount of time required for the newly added processor to complete N_{SA} search iterations multiplied by the probability that either it or any one of the P_{SA} processors above it in the search tree succumbs to sensor failure. Analytically, this value can be expressed as:

$$T_{SENSOR\ FAIL}(\mathrm{P_{SA}}) = \left(1 - p_{SENSOR\ SUCCESS}{}^{P_{SA}+1}\right)\cdot T_{N_{SA}} \tag{7.4}$$

where T_{NSA} is the average time required for one sensor to complete N_{SA} search iterations and $p_{SENSOR\ SUCCESS}$ is the probability that a given sensor completes its N_{SA} search iterations without failing. This value is dependent on the wireless sensor platform being used and the environment in which it is deployed, but is typically very high (>0.999).

For any simulated annealing task, $T_{COMM\ FAIL}$ represents the increase in expected processing time lost due to communications failure brought about by the addition of one processor. Like $T_{SENSOR\ FAIL}$, this quantity can also be derived analytically. In a similar fashion, if any sensor loses communication with its parent while it is involved in a parallel SA task, any work done by the failed node and all nodes below it would be lost. As such, $T_{COMM\ FAIL}$ can be expressed as the amount of time required for the newly added processor to complete N_{SA} search iterations multiplied by the probability that either it or any one of the P_{SA}-1 processors immediately above it in the search tree loses parental communication. At first, it appears that this value can be calculated in a manner similar to that of $T_{SENSOR\ FAIL}$, but the probability of success in each parent-child communication link is not only dependent on P_{SA}, but also on the signal strength of the respective wireless communication link. This value can be quantified for each communication link, c, and represented by a radio signal strength indicator ($RSSI_C$). As such, an analytical value for $T_{COMM\ FAIL}$ can be expressed as:

$$T_{COMM\ FAIL}(\mathrm{P_{SA}}, RSSI_C) = p_{COMM\ FAIL} \cdot T_{N_{SA}} \tag{7.5}$$

where T_{NSA} is as before and

$$p_{COMM\ FAIL} = 1 - \prod_{c=1}^{P_{SA}} \frac{p_{COMM\ SUCCESS}}{1 + e^{-0.4(40.0+RSSI_C)}} \tag{7.6}$$

Having examined in more detail the derivation of T_{SPEED}, $T_{SENSOR\ FAIL}$, and $T_{COMM\ FAIL}$, it can now be seen

Table 7.1. Coefficients for calculating T_{SPEED} and T_{COMM}

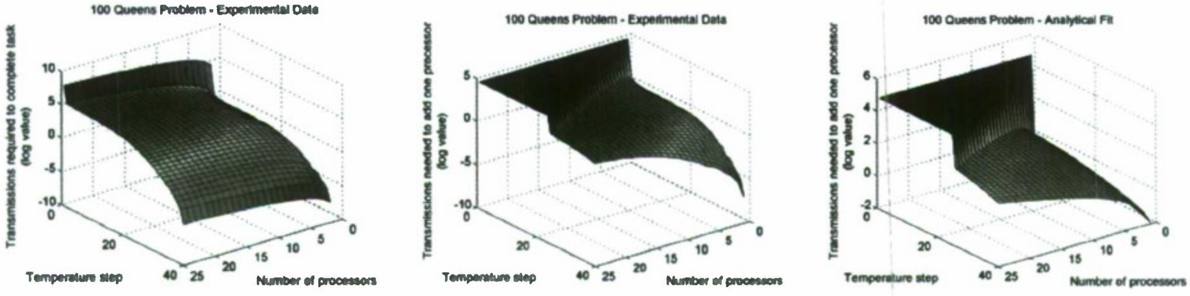
		Number of Queens (C_{SA})			
		25	50	75	100
T_{SPEED}	α	0.0	1.0	8.0	20.0
	β	12.3	35.9	73.5	126.9
	δ	13.0	23.0	27.0	29.5
	γ	8.3	19.7	39.5	63.4
T_{COMM}	α	2.5	7.4	12.7	17.9
	β	0.3	2.1	4.6	7.3
	δ	27.5	56.7	85.5	114.2

from Equation 7.2 that α_{BUYER}, and β_{BUYER} are weighting parameters that allow the wireless sensor network to exactly prioritize between speedup, communication reliability, and sensor reliability. This type of weighting creates an extremely adaptable network that can change, in real-time, to shifting computing needs within a wireless network.

On the seller side, a somewhat simpler utility function, U_{SELLER}, can be developed in a similar fashion to U_{BUYER}. Intuitively, seller utility can be thought of as the total amount of additional power a computational task requires as a result of adding an additional processing node. Because the majority of power consumption in a wireless sensing device comes from the wireless radio (which consumes five times more power than a microcontroller), the seller can maximize its utility by minimizing the amount of time the wireless network spends communicating. As such, U_{SELLER} can be defined as follows:

$$U_{SELLER} = -T_{COMM} \tag{7.7}$$

For any simulated annealing task, T_{COMM} represents the increase in expected communication time required to complete the task brought about by the addition of one processor. Much like T_{SPEED}, there is no way of directly formulating an analytical expression for this value. As such, a communications trend can be established for any computational task by looking at the average amount of time that a task of complexity C_{SA} spends communicating before converging on a solution from a given temperature step, TS_{SA}, with a given number of processors, P_{SA}. Using data collected over a large number of experimental trials, this value can be expressed as the difference between the average number of communications required to complete a search where P_{SA} nodes are currently searching up to temperature step TS_{SA} and the average number of communications required to complete a search where P_{SA}+1 nodes are currently searching up to temperature step TS_{SA}+1. Figure 7.4(a) shows the average number of communications required to complete the 100-Queens problem from a given state (P_{SA}, TS_{SA}) in the search, and Figure 7.4(b) shows the amount of additional communications required by adding one more processing node when the 100-Queens problem is in a given state (P_{SA}, TS_{SA}). Note that all surface plots in Figure 7.4 are presented on a logarithmic scale. This is because the number of communications required during the first temperature step are magnitudes larger than what is required during any other temperature step and a log scale is required in order to properly view the remainder of the graph. This phenomenon occurs because of the large number of new global minimum states that are found at the beginning of the SA search process. Using simple linear approximations, the surface in Figure 7.4(b) can be approximated by the following easily computable algebraic function:



(a) (b) (c)

Figure 7.4. (a) Experimental time data for each temperature step/processor combination, (b) graphical representation of experimentally determined T_{COMM}, and (c) the accompanying analytical model fit.

$$T_{COMM}(\mathrm{C_{SA}}, \mathrm{P_{SA}}, \mathrm{TS_{SA}}) = \begin{cases} \left(\dfrac{\gamma - TS_{SA}}{\gamma - P_{SA}}\right) \cdot \left(\dfrac{\alpha \cdot P_{SA}}{25}\right) & \mathit{if}\ TS_{SA} > P_{SA} \\ \partial & \mathit{otherwise} \end{cases} \tag{7.8}$$

where

$$\gamma = 25 + \frac{15 \cdot \alpha}{(\alpha - \beta)} \tag{7.9}$$

and where values for α, β, and δ are specific to each task complexity, C_{SA}, and are tabulated in Table 7.1. The quality of the analytical fit provided by this function for the 100-Queens problem can be seen in Figure 7.4(c). Fits of similar quality can be found for all other problem complexities (25-Queens, 50-Queens, and 75-Queens).

Interestingly, the experimentally derived coefficients α, β, and δ are linearly correlated to the problem complexity, C_{SA}, allowing us to easily expand the algebraic expression for T_{COMM} to n-Queens problems of any arbitrary non-negative complexity:

$$T_{COMM}(\mathrm{C_{SA}}, \mathrm{P_{SA}}, \mathrm{TS_{SA}}) = \begin{cases} \left(\dfrac{40 + 0.11 \cdot C_{SA} - TS_{SA}}{40 + 0.11 \cdot C_{SA} - P_{SA}}\right) \cdot \left(\dfrac{0.21 \cdot C_{SA} \cdot P_{SA}}{25}\right) & \mathit{if}\ TS_{SA} > P_{SA} \\ 1.16 \cdot C_{SA} & \mathit{otherwise} \end{cases} \tag{7.10}$$

7.2.3 Wireless task distribution algorithm:

Having developed utility functions associated with both buyers and sellers, it is now possible to create a methodology with which wireless sensing units can buy and sell processing time. By expanding on the fundamental principles of an auction, the following procedure is developed: (Note that at any time, a new computational task can be assigned to any wireless sensing unit not otherwise occupied).

Step 1) All sensing units not currently processing will broadcast their availability to the network.

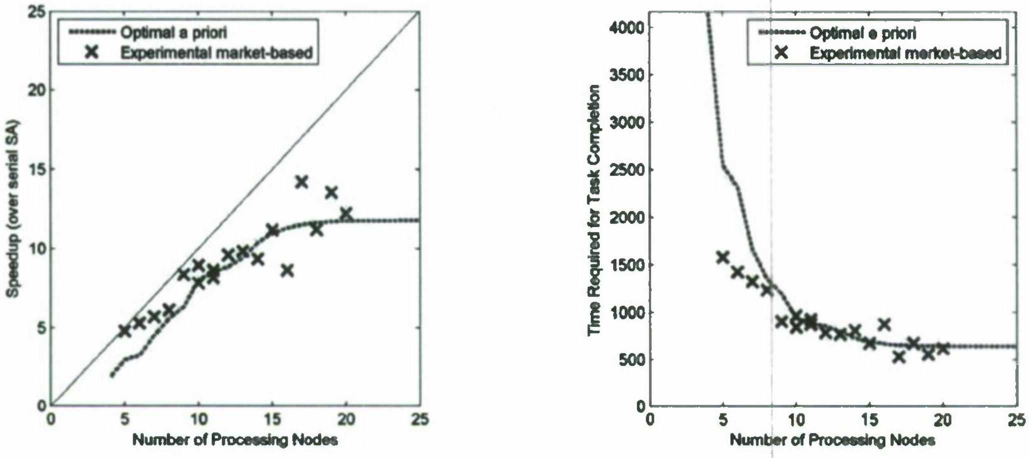
Step 2) Buyers at the bottom of each existing computational task chain will calculate U_{BUYER} based on the computational task they are working on, and place a bid of U_{BUYER} if $U_{BUYER} > 0$.

Step 3) Sellers will calculate U_{SELLER} based on each proposed computational job.

Step 4) Once all bids have been received, sellers will calculate their expected profit from each proposed job using a market power / speed exchange rate (γ_{MARKET}) that represents the minimum number of seconds of computational speedup that must be gained in order to warrant an additional second of communication:

$$\mathit{profit} = U_{BUYER} - \gamma_{MARKET} \cdot U_{SELLER} \tag{7.11}$$

Step 5) Sellers will choose the bid that generates the greatest non-negative profit.



(a) (b)

Figure 7.5. Experimental market-based task assignment vs. optimal a priori task assignment (a) speedup and (b) time to completion

In this way, computational assignments will be distributed throughout the network in such a way that the overall utility of the market as a whole is maximized. Because of the addition of the weighting parameters, α_{BUYER}, β_{BUYER}, and γ_{MARKET}, the resulting framework is capable of optimally adapting, in real-time, to shifting computing needs within a wireless network. For example, assume a computing task surfaces where quality communication channels are absolutely essential. Without any reprogramming of the sensing network, the network can reassign a larger β_{BUYER} value in order to reflect the added emphasis on avoiding communication failure.

7.3 Experimental testbed:

In this study, four optimization tasks of varying complexity (the 25-Queens, 50-Queens, 75-Queens, and 100-Queens problems) are randomly assigned to four available *Narada* wireless sensors. Each of these four sensors then becomes the "master" node in the P_{SA} search tree associated with their given *n*-Queens task. After these assignments have been made, an additional pool of processing nodes (containing between 1 and 16 wireless sensors) is made readily available for computational use. At this point, the market-based bidding process begins, and each of the four master nodes are allowed to "bid" on the computational services of the free sensing nodes. This bidding process proceeds as described in Section 7.2.3. If a "master" node finishes the PSA search at its assigned temperature step without finding a global minimum, it will pass its "master" status on to its child, making itself available for computation on any of the four computational tasks.

7.4 Experimental results:

In order to evaluate the performance of the proposed market-based task distribution methodology, it is first necessary to establish a benchmark against which to compare timing results. In this case, in order for the market-based method to be proven effective, it must be shown that a sensing network utilizing market-based methods is capable of completing the four assigned tasks at least as quickly as if an optimal subset of processors had been assigned to each task at the outset of computation. In fact, even a certain amount of degradation in performance with respect to an *a priori* optimization may serve to validate the method in this case, as the scalable benefits of real-time task assignment would greatly outweigh a small amount of time savings when dealing with full-scale problems. For example, while an optimal *a priori* task distribution is trivial to calculate when dealing with only four distinct tasks, that optimization problem grows exponentially as additional tasks are added.

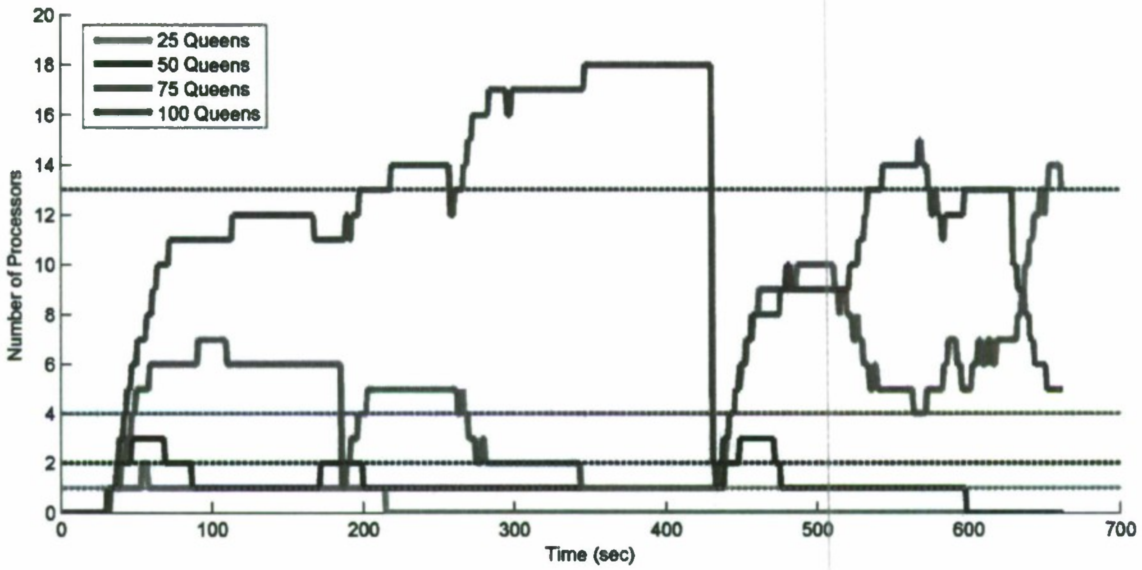


Figure 7.6. Example record of market-based task distribution over time, compared with optimal a priori assignment (dotted)

Thus, the first step in evaluating the proposed market-based method is to calculate the optimal *a priori* task assignment of processing units for each of the test cases (5-20 *Narada* units). This is accomplished by using a brute-force search of the possible task assignments. Then, experimental data is gathered using *Narada* networks of varying sizes. In each experimental instance, the network of sensors is asked to solve all four *n*-Queens problems. Figure 7.5(a) and 7.5(b) show the experimental market-based performance against the *a priori* performance. Figure 7.5(a) displays the speedup of each method relative to a serial implementation, and Figure 7.5(b) shows the time required for each method to complete the four tasks assigned to it. It can be seen from these plots that the market-based task distribution method performs as well, if not better than an a priori optimal assignment of tasks. Note that there is inherent scatter in the market-based results, as the SA algorithm itself fluctuates widely in its speed to convergence, and not enough data was gathered to form a statistically significant curve. But as a average, it can be seen that the proposed market-based method actually performs better than an a priori optimal distribution. Figure 7.6 shows an example time history of the number of sensing units in a 20-node network assigned to each task as a function of time. The dotted lines indicate the optimal a priori sensor distribution for a 20-node network working on these four tasks.

7.5 Section summary and conclusions:

This section demonstrates a free-market method of optimally allocating and consuming scarce system resources (such as battery power, data storage capacity, MPU time, wireless bandwidth, etc.) within a network of wireless sensing devices. In this free-market framework, multiple computational tasks are assigned to individual sensors within a wireless network, and additional sensors available for computation are distributed amongst these computational tasks through a bidding process. In this section, computational tasks are represented by simulated annealing search problems of varying complexity C_{SA} that can be broken into separate blocks of N_{SA} SA search iterations. It is found that free-market distribution methods allow a set of multiple computational tasks to be completed more quickly than possible with any set distribution of processors, assigned at the outset of the task assignment.

8. Conclusions:

This project has explored the deployment of wireless telemetry in naval plant systems. Unlike traditional wired counterparts, wireless sensors and wireless actuators can be installed at a fraction of the cost of wired counterparts. In addition, the flexibility of the wireless communication channel supports ad-hoc connectivity which allows wireless topologies to heal when battle damage occurs. The low-cost Narada wireless node has been developed capable of collecting data from sensors, processing data at the node, actuating actuators, and communicating with other nodes via the wireless channel.

With a very capable wireless node proposed for integration in shipboard plants, the project turned its attention to the computational functionality of the wireless sensor/actuator network to perform condition assessment (*i.e.*, damage detection) in the plant. In particular, model updating methods were adopted to find a best fit model of a plant using measurements collected from a network of wireless sensors. In particular, simulated annealing is adopted due to its computational simplicity and flexibility to be distributed in a network of computing agents. This study proposes a parallel simulated annealing algorithm designed specifically to efficiently utilize the distributed resources available in large networks of wireless devices. This algorithm gains efficiency as the number of sensors in a network grows, making it scalable to very large networks, and it can be applied to many of the large number of combinatorial optimization problems seen across many engineering disciplines. The proposed algorithm is embedded within a network of Narada nodes to update an analytical model describing the flow of water in a chilled water plant. By altering the model's properties (e.g., pipe diameters) such that the analytical modal output (pipe flows) match experimentally sensed data, properties of the physical plant can be accurately estimated. The method is proven effective for estimating rupture conditions in a chilled water demonstrator system.

This project also proposed a market-based method of optimally allocating scarce system resources (such as battery power, data storage capacity, CPU time, wireless bandwidth, *etc.*) amongst a set of multiple computational objectives within a wireless sensor/actuator network. In this buyer/seller framework, available wireless sensors (sellers) are distributed amongst multiple computational tasks (buyers) through a utility-driven bidding process. Because buyers and sellers in this market gain utility in different ways (buyers by maximizing speed and reliability and sellers by minimizing power consumption), a Pareto-optimal allocation of scarce resources can be reached while completing a set of multiple computational objectives as quickly as possible. When evaluating the proposed resource allocation algorithm on a physical network of wireless sensor prototypes, it is found that this method allows a set of multiple computational tasks to be completed as quickly as if an optimal number of sensors were assigned *a priori* to each computational task at the outset of computation. This property is extremely advantageous, especially as the number of computational tasks and/or available processors increases. By showing how this market-based allocation methodology can be applied to the problem of rupture detection within shipboard chilled water systems, the real-world applicability of the proposed method is demonstrated.

9. Papers Published and Technology Transfer:

A variety of technology transfer activities were conducted in conjunction with this project. In collaboration with researchers from the Naval Surface Warfare Center Carderock Division (Point of Contact: Mr. Thomas Brady), the wireless sensors developed as part of this project have been used to record the strain and acceleration response of the FSF-1 SeaFighter, a high-speed aluminum littoral combat ship in current service. A 31-channel wireless sensor network was installed to record hull strains and accelerations prior to the ship departing from its home port in Panama City, Florida. As the ship travelled from Panama City to Portland, Oregon, the response of the ship was recorded to various sea states. This project validated the performance of the wireless monitoring system on an operational naval vessel while simultaneously showing the applicability of the wireless sensor technology to hull monitoring applications. In addition, a total of 24 high quality publications were published based on the work conducted in this project:

1. Zimmerman, A. T., and Lynch, J. P. "Market-Based Frequency Domain Decomposition for Automated Mode Shape Estimation in Wireless Sensor Networks," *Journal of Structural Control and Health Monitoring,* Wiley, *submitted,* 2009.

2. Zimmerman, A. T. and Lynch, J. P. "A Parallel Simulated Annealing Architecture for Model Updating in Wireless Sensor Networks," *IEEE Sensors Journal*, IEEE, *in press,* 2009.

3. Swartz, R. A. and Lyneh J. P. "Strategic Network Utilization in a Wireless Structural Control System for Seismically Exeited Structures," *Journal of Structural Engineering,* ASCE, *in press,* 2009.

4. Lynch, J. P., Wang, Y., Swartz, R. A., Lu, K. C. and Loh, C. H. "Implementation of a Closed-Loop Structural Control System using Wireless Sensor Networks," *Journal of Structural Control and Health Monitoring*, Wiley, 15(4): 518-539, 2008.

5. Lynch, J. P. and Loh, K. J. "A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring," *Shock and Vibration Digest*, Sage Publications, 38(2): 91-128, 2006.

6. Swartz, R. A., and Lynch, J. P., "Self-Tuning Control of Seismically Excited Structures over a Wireless Sensor Network," *Proceedings of ANCRiSST 2009: the 5th International Workshop on Advanced Smart Materials and Smart Structures Technologies*, Northeastern University, Boston, MA, 2009.

7. Lynch, J. P., Swartz, R. A., Zimmerman, A. T., Brady, T. F., Rosario, J., Salvino, L. W. and Law, K. H., "Monitoring of a High Speed Naval Vessel using a Wireless Hull Monitoring System," *Proceedings of the 7th International Workshop on Structural Health Monitoring*, Stanford, CA, 2009.

8. Zimmerman, A., and Lynch, J. P., "A Market-Driven Computing Architecture for the Simultaneous Execution of Multiple Computational Tasks within Dynamic Wireless Structural Monitoring Systems," *Proceedings of the 7th International Workshop on Structural Health Monitoring*, Stanford, CA, 2009.

9. Swartz, R. A., Lynch, J. P., and Loh, C. H., "Near Real-Time System Identification in a Wireless Sensor Network for Adaptive Feedback Control," *Proceedings of the American Controls Conference (ACC2009)*, IEEE, St. Louis, MO, 2009.

10. Zimmerman, A. T., Lynch, J. P., Ferrese, F. T., "Market-based Computational Task Assignment within Autonomous Wireless Sensor Networks," *Proceedings of the IEEE International Conference on Electro/Information Technology*, Windsor, Canada 2009.

11. Swartz, R. A., Zimmerman, A. T., Lynch, J. P., Brady, T. F., Rosario, J., Salvino, L. W. and Law, K. H., "Wireless Hull Monitoring Systems for Modal Analysis of Operational Naval Vessels," *Proceedings of the International Modal Analysis Conference (IMAC) XXVII*, Orlando, Florida 2009.

12. Lynch, J. P., Swartz, R. A. and Zimmerman, A. T., "Distributed Data Processing Architectures for Structural Monitoring Systems Implemented on Wireless Sensor Networks," *Proceedings of the Fourth European Workshop on Structural Health Monitoring,* Poland, 2008. **(Invited Keynote Paper)**

13. Zimmerman, A. T. and Lynch, J. P., "Distributed Data Processing within Dense Networks of Wireless Sensors using Parallelized Model Updating Techniques," *SPIE Smart Structures and Materials*, San Diego, CA, 2008.

14. Zimmerman, A., Swartz, R. A. and Lynch, J. P., "Automated Identification of Modal Properties in a Steel Bridge Instrumented with a Dense Wireless Sensor Network," *Proceedings of the 4th International Conference on Bridge Maintenance, Safety and Management*, Seoul, Korea 2008.

15. Zimmerman, A. and Lynch, J. P., "Distributed Model Updating in Smart Wireless Monitoring Systems," *Proceedings of the 2008 ASCE Structures Congress*, Vancouver, Canada 2008.

16. Swartz, R. A, Zimmerman, A. T. and Lynch, J. P., "Structural Health Monitoring System with the Latest Information Technologies," *Proceedings of 5th Infrastructure & Environmental Management Symposium*, Yamaguchi, Japan, 2007. **(Invited Keynote Paper)**.

17. Zimmerman, A. T. and Lynch, J. P., "Automated Damage Estimation in Wireless Sensing Networks using Parallelized Model Updating," *Proceedings of the 6th International Workshop on Structural Health Monitoring*, Stanford, CA, 2007.

18. Swartz, R. A. and Lynch, J. P., "Partial Decentralized Wireless Control Through Distributed Computing for Seismically Excited Civil Structures: Theory and Validation," *Proceedings of the American Controls Conference (ACC2007)*, IEEE, New York, NY, 2007.

19. Zimmerman, A., Swartz, R. A., Saftner, D. A., Lynch, J. P., Shiraishi, M. and Setareh, M., "Parallel Data Processing Architectures for Identification of Structural Modal Properties using Dense Wireless Sensor Networks," *World Forum on Smart Materials and Smart Structures Technology*, Chongqing, China, 2007.

20. Zimmerman, A. T. and Lynch, J. P., "Parallelized Simulated Annealing for Model Updating in Ad-Hoc Wireless Sensing Networks," *Proceedings of the International Workshop on Data Intensive Sensor Networks*, Mannheim, Germany, 2007.

21. Swartz, R. A. and Lynch, J. P., "Redundant Kalman Estimation for a Distributed Wireless Structural Control System," *Proceedings of the US-Korea Workshop on Smart Structures Technology for Steel Structures*, Seoul, Korea, 2006.

22. Wang, Y., Swartz, R. A., Lynch, J. P., Law, K. H. and Loh, C. H., "Performance Evaluation of Decentralized Wireless Sensing and Control in Civil Structures," *SPIE Smart Structures and Materials*, San Diego, CA, 2007.

23. Zimmerman, A. T. and Lynch, J. P., "Data Driven Model Updating using Wireless Sensor Networks," *Proceedings of the 3rd Annual ANCRiSST Workshop*, Lake Tahoe, CA, 2006.

24. Swartz, R. A., Jung, D., Lynch, J. P., Wang, Y., Shi, D. and Flynn, M. P., "Design of a Wireless Sensor for Scalable Distributed In-Network Computation in a Structural Health Monitoring System," 5th *International Workshop on Structural Health Monitoring*, Stanford, CA, 2005.

References:

Adept Systems, Inc. (2004). "Control strategies and algorithms," *Presentation to the ONR Workshop on Distributed Intelligence for Automated Survivability (DIAS)*, Laurel, MD.

Bounds, D. G. (1987). "New optimization methods from physics and biology," *Nature*, vol. 329, pp. 215-219.

Dunnington, L., Stevens, H. and Grater, G. (2003). *Integrated Engineering Plant for Future Naval Combatants – Technology Assessment and Demonstration Roadmap*. Technical Report MSD-50-TR-2003/01, Anteon Systems, Fairfax, VA.

Estes, D. R. J., Welch, T. B., Sarkady, A. A. and Whitesel, H. (2001). "Shipboard radio frequency propagation measurements for wireless networks," *Proceedings of the IEEE Military Communications Conference (MILCOM)*, McLean, VA, pp. 247-251.

Greening, D. R. (1990). "Parallel simulated annealing techniques," *Physica D*, vol. 42, pp. 293-306.

Horton, M., Culler, D., Pister, K. S. J., Hill, J., Szewczyk, R. and Woo, A. (2002). "MICA - The Commercialization of Microsensor Motes," *Sensor Magazine*, Advanstar Communications Inc, 19(4): 23-28.

Jennings, N. R., Sycara, K., and Wooldridge, M. (1997). "A roadmap of agent research and development," *Autonomous Agents and Multi-Agent Systems*, 1:7–38.

Kirkpatrick, S., Gelatt, Jr., C. D. and Vecchi, M. P. (1983). "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680.

Levin, R. I. and Lieven, N. A. J. (1998). "Dynamic finite element model updating using simulated annealing and genetic algorithms," *Mech. Syst. Signal Process.*, vol. 12, no. 1, pp. 91-120.

Lunze, J. (1992). *Feedback Control of Large-Scale Systems*, Prentice Hall, NY.

MacGillivray, P. and Goddard, K. (1997). "Advanced sensor technology for marine propulsion control systems," *Proceedings of the Eleventh Ship Control Systems Symposium*, United Kingdom, pp. 245-256.

Maturana, F. P., Tichy, P., Slechta, P., Discenzo, F., Staron, R. J. and Hall, K. (2004). "Distributed multi-agent architecture for automation systems," *Expert Systems with Applications*, Pergamon, 26(2004):49-56.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E. (1953). "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087-1092.

Min, R., Bhardwaj, M., Seong-Hwan C., Shih, E., Sinha, A., Wang, A. and Chandrakasan, A. (2001). "Low-Power Wireless Sensor Networks," *Proceedings of the 14th International Conference on VLSI Design*, Bangalore, India, pp. 205-210.

Mokole, E. L., Parent, M., Samaddar, S. N., Tomas, E. and Gold, B. T. (2000). *Radio-frequency Propagation Measurements in Confined Ship Spaces Aboard the ex-USS Shadwell*. Report Number A987183, Naval Research Laboratory, Washington D. C.

Office of Naval Research (ONR) (2003). *A Future Naval Capability: Electric Warships & Combat Vehicles.* Media-Release Fact Sheet, Office of Naval Research, Washington, D. C.

Ploeger, R., Newton, W. W., Rabiner, A. and Lally, R. (2003). *Wireless e-diagnostics Reduces Workload and Improves Shipboard Quality of Life.* White Paper, Oceana Sensor Technologies, Virginia Beach, VA.

Schwartz, G. (2002). "Reliability and survivability in the reduced ship's crew by virtual presence system," *Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN'02)*, Washington D.C., pp. 199-204.

Seman, A. J., Toomey, K., amd Lang, S. (2003). *Reduced Ship's Crew by Virtual Presence (RSVP) Advanced Technology Demonstration.* Technical Report, Office of Naval Research (ONR), Arlington, VA.

Siljak, D. D. (1991). *Decentralized Control of Complex Systems*, Academic Press, Boston, MA.

Smith, S. (1994). "An approach to intelligent distributed control for autonomous underwater vehicles," *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, Cambridge, MA, pp. 105-111.

Sun, L.-H. and Cartes, D. (2004). "Reconfiguration of Shipboard Radial Power System using Intelligent Agents", *Proceedings of the American Society of Naval Engineers (ASNE) Electric Machines and Technology Symposium*, Philadelphia, PA.

Wooldridge, N. and Jennings, N. R. (1995). "Intelligent agents: theory and practice," *Knowledge Engineering Review*, 10(2):115-152.