Agent-Based Computational Architectures for Distributed Data Processing in Wireless Sensor Networks

by

Andrew T. Zimmerman

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in The University of Michigan
2010

Doctoral Committee:

Associate Professor Jerome P. Lynch, Chair
Professor Victor C. Li
Professor Quentin F. Stout
Associate Professor Vineet R. Kamat

2010

DEDICATIONS

This Work is Dedicated To:

My Parents … for teaching me to live
My Wife … for helping me to live more fully
My Savior … for allowing me to live with purpose

ACKNOWLEDGEMENTS

While conducting the research presented in this dissertation, the author was supported financially by the National Defense Science and Engineering Graduate Fellowship, the University of Michigan Rackham Predoctoral Fellowship, and the University of Michigan Greene Scholarship. Additional support was provided by the US Office of Naval Research (Contracts N00014-05-1-0596 and N00014-09-C0103), and the National Science Foundation (Grants CMS-0528867 and CMMI-0726812).

In addition to this financial support, many individuals have greatly contributed to the research contained herein. I would like to acknowledge first and foremost my advisor, Prof. Jerome Lynch. He has constantly supported my efforts, both personally and professionally, and has provided me with all the resources necessary to succeed at the highest possible level. His guidance and friendship have been invaluable, and I am incredibly grateful for his support over the course of my graduate studies.

The next group of people to whom I owe a debt of gratitude are my fellow graduate students at the University of Michigan with whom I have grown as a researcher and as a person. In particular: Ken Loh, Junhee Kim, and Tsung-Chin Hou. An extra special thanks go to Andrew Swartz, who has been my co-conspirator in wireless intelligence and who has put up with me on various trips to three continents and one infamous ocean.

I am also indebted to many individuals at both the University of Michigan and at other academic institutions who have assisted me in completing the research presented herein: My doctoral committee members, Prof. Victor Li (CEE), Prof. Vineet Kamat (CEE), and Prof. Quentin Stout (CSE) for their guidance and support with my thesis; Michihito Shiraishi (Shimizu Corporation), David Saftner (University of Michigan), Prof. Mehdi Setari (Virginia Tech), and Prof. Richard Woods (University of Michigan), for their assistance with the theatre balcony field tests presented in Chapter 2; Prof. Kincho Law (Stanford University) and Prof. Chin-Hsiung Loh (National Taiwan University) for their help with the steel structure shaking table tests presented in Chapter 3; Frank Ferrese, David Kocsik, and Tony Seman at the Naval Surface Warfare Center for their assistance with the development and testing of both the market-based resource allocation method and wireless model updating technique presented in Chapter 3 and Chapter 4.

Outside of the academic realm, I must also thank those individuals at the University of Michigan who made my social life during my graduate studies one I will look upon fondly: Pascal Laumet, Min-Yuan Cheng, Remy Lequesne, Matt Fadden, Jason McCormick, and pretty much everyone studying Analytical Chemistry (Chris Avery, Anna Clark, Kate Dooley, Katie Hersberger, and Natalie Walker in particular).

Above all, I must acknowledge the love and support of my incredible family: my beautiful wife, Laura, who has always provided me with a reminder of what is truly important in life, and has been a constant source of strength, inspiration, and love; and my parents, Jerry and Jo Ann, who have always told me that I could accomplish anything I put my mind to, and have lovingly supported every step of my academic journey since I first stepped on that yellow bus to go to kindergarten. I owe them everything.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Agent-Based Computational Architectures for Distributed Data Processing in Wireless Sensor Networks

by

Andrew T. Zimmerman

Chair: Jerome P. Lynch

As the structural health monitoring (SHM) community continues to develop algorithms for monitoring performance and detecting degradation in engineered systems, the importance of pervasive sensing and autonomous data processing methodologies will increase. Fortunately, the emergence of wireless sensor technologies at the forefront of SHM research has provided a platform on which problems related to both sensor density and processing autonomy can be addressed. By utilizing wireless communication links instead of expensive data cables, wireless monitoring systems can be deployed with much greater sensor density and at significantly lower costs than traditional SHM systems. Perhaps more importantly, because wireless sensing units typically integrate a traditional sensor with a low-power microprocessor, analog-to-digital converter, and wireless

transceiver, wireless sensing networks (WSNs) have shown great promise in their ability to process sensor data in-network (*i.e.*, without the need for a centralized data center).

Over the past decade, the wireless SHM community has shown that it is possible to minimize problems associated with power efficiency, data loss, and finite communication range by processing data before transmitting it to a central repository. Recently, in an effort to further improve the efficiency and capability of in-network computation, researchers have started to move away from centralized processing frameworks (where no data is shared between wireless nodes) towards more hierarchical data processing architectures. However, work to date in this area has yet to fully leverage the computational advantages provided in large networks of wireless sensors.

In this dissertation, several distinct agent-based architectures are developed for distributed data processing in WSNs. Each of these agent-based architectures leverages the ad-hoc communication and pervasive nature inherent to wireless sensing technology, and can be viewed as a parallel computing system with an unknown and possibly changing number of processing nodes. As such, sophisticated data analysis can be performed while maintaining a scalable environment that is not only resistant to sensor failure, but that also becomes increasingly efficient at higher nodal densities. These agent-based architectures represent a significant step towards the creation of a fully autonomous WSN for application to SHM.

CHAPTER 1

INTRODUCTION

1.1 The Need for Structural Health Monitoring

Complex structural systems such as buildings, bridges, pipelines, aircraft, and ships, among many others, all play a vital role in maintaining the commercial, social, and recreational interests of modern society. While these systems are designed to provide many years of safe functionality under normal operating conditions, it is often desirable or even necessary to evaluate system performance either in the wake of time-based deterioration such as that caused by cracking (Zagrai and Giurgiutiu 2001), fatigue (Wu and Huang 1993), and corrosion (Simmers Jr.*, et al.* 2006), or after extreme loading scenarios such as earthquakes (Hou*, et al.* 2006), tsunamis (Schmitz*, et al.* 2007), and terrorist attacks (Kevin 2004). Traditionally, this type of structural performance evaluation has been performed by trained professional engineers using either visual inspection methods (Estes and Frangopol 2003) or, in some cases, more sophisticated non-destructive techniques (Washer 1998). However, in light of recent catastrophic structural failures, such as the September 11, 2001 collapse of the World Trade Center towers in New York City (NIST 2005), the May 23, 2004 roof collapse at the Charles de Gaulle international airport in Paris (Reina 2004a, 2004b), and the August 1, 2007 I-35W

bridge collapse in Minneapolis, Minnesota (FEMA 2007), it has become clear that more sophisticated means of structural inspection may be necessary to help prevent future loss of life and/or property due to structural degradation or in the aftermath of a catastrophic damage event.

To this end, structural health monitoring (SHM) systems, which combine networks of sensors with automated system identification and damage detection techniques, have garnered much attention in both the academic and commercial sectors (Farrar and Worden 2007). Representing a long-term method of continuous system evaluation that can be applied to critical infrastructure systems in a variety of fields, SHM systems are important because they reduce the routine maintenance and inspection costs associated with critical infrastructure. In addition, they can increase the safety of a structure by alerting engineers to potential structural problems well before catastrophic failure occurs.

1.2 Current State-of-Practice in Structural Health Monitoring

Since the initial development of critical infrastructure systems (such as temples, roads, aqueducts, canals, *etc.*), mankind has relied almost exclusively on either visual cues (such as cracking, leaking, or decreased performance) or the use of simple tool-based diagnostics (hammer tapping, load testing, *etc.*) in order to determine the state of health of a given system or structure. Despite the many advances in sensor technologies, data collection capabilities, and computer-based data processing tools, most of today's physical infrastructure (bridges, pipelines, automobiles, airplanes, *etc.*) are still subjected to some form of visual inspection. For example, in the United States, the Federal

Highway Administration requires highway bridges to be inspected visually once every 24 months (FHWA 2004). The National Bridge Inspection Program (NBIP) was mandated by congressional legislation after the catastrophic Silver Bridge collapse in 1967 (Lichtenstein 1993). Similar visual inspections are also the most common mandatory inspection technique used in the naval (Ludwig and Conrardy 2007) and aerospace (Samsonov 1995) communities.

In order to supplement traditional visual inspection methodologies, many non-destructive testing (NDT) methods have been developed over the past three decades (Malhotra and Carino 2004). These techniques are all designed to add some degree of quantitative data to the damage detection process without harming the inspected structure. For example, a large suite of non-destructive techniques including proof loading, coring, vibration and impact testing, ultrasonics, conductivity mapping, and radar methods have all been used to supplement visual inspections of concrete and masonry-based civil structures (McCann and Forde 2001). Similarly, in the aerospace industry, NDT methods like ultrasonics, radiography, thermography, and acoustic emissions have all been used to assist in the inspection of aluminum and composite aircraft structures (Mahoon 1988).

By leveraging recent advances in sensing capabilities and computing power, the pace of SHM research has quickened significantly over the past decade in an attempt to supplement or replace visual and technician-based (manual) NDT inspection methods. Specifically, considerable effort has been placed on developing automated damage detection algorithms using a variety of techniques including vibration-based methods (Doebling*, et al.* 1998) and guided-waves (Raghavan and Cesnik 2007). To date,

however, the only truly significant commercial application to which automated SHM techniques have proved completely successful is the condition monitoring (CM) of rotating machinery (Randall 2004a, 2004b). In this field, changes in both vibration signature and lubricant content can be directly correlated to certain kinds of damage in an operating machine, and it has been shown that these changes can be detected and analyzed without human interaction.

However, even without the development of successful automated damage detection methodologies, most industries today have still embraced (to some degree) the installation of networks of sensing transducers on their structural systems in order to assist the diagnostic capabilities of human inspectors. For example, networks of sensors have already been installed in many large civil structures around the world (Hipley 2000; Wu 2003; Ko and Ni 2005). Similarly, modern aircraft (Staszewski*, et al.* 2004) and naval vessels (Slaughter*, et al.* 1997) are currently outfitted with hundreds of sensors which are used to assist inspectors in assessing the health and remaining lifespan of the structural systems they are monitoring.

1.2.1 Limitations of Current SHM Systems

While commercial applications of SHM technologies are becoming more prevalent, the current state-of-practice in structural inspection (with or without SHM assistance) is woefully inadequate at detecting the onset of structural failure before it becomes catastrophic. A poignant example of this truth is the aforementioned I-35W bridge collapse (FEMA 2007). Despite numerous visual and NDT inspections in the years prior to the collapse, each of which detected serious problems, the bridge was

deemed to have met minimum tolerable limits and was left in place without major repair (Dedman 2007). Clearly, in this case more sophisticated SHM methods may have aided the inspectors and possibly prevented disaster.

Unfortunately, there are many factors preventing the widespread adoption and use of SHM technology in today's structural systems. As mentioned before, the lack of effective autonomous damage detection methods have been a detriment to the value of SHM systems; since no one method or algorithm has been shown to detect damage in a variety of large structural applications, the field has been unable to produce any solution that functions as reliably as CM systems do for rotating machinery. But perhaps the most important hindrance to modern SHM is actually the high cost associated with the installation and maintenance of large numbers of sensing transducers distributed throughout a large structural system.

In a typical civil SHM application, for example, a variety of point sensors (such as accelerometers, temperature sensors, strain gages, linear voltage displacement transducers, among others) are deployed throughout the monitored structure and are connected to a central data repository using long runs of coaxial cable. These coaxial cables are utilized to provide power to each individual sensing transducer and to carry sensor data back from each transducer to the central repository for storage and analysis. In large buildings, it has been found that tethered monitoring systems can incur cabling costs on the order of several thousand dollars per sensing channel (Celebi 2002). Since many damage detection methods desire high sensing density (*i.e.*, sensing transducers placed at as many discrete locations as possible throughout a structure), the cost associated with deploying a high resolution SHM system in a complex engineered system

grows steeply with the size of the associated sensing network. The Tsing-Ma Suspension Bridge in Hong Kong, for example, is currently monitored with over 350 tethered sensors that were installed at a cost of over $8 million (Farrar 2001).

1.3 Current State-of-the-Art in Structural Health Monitoring

In an attempt to address some of the limitations inherent to current in-practice SHM systems, the research community has been actively working on a variety of new sensing and damage detection technologies. At the forefront of this research includes work with guided waves for damage detection, sensor minimization for decreased sensor costs, and wireless sensor networks for a reduction in installation and maintenance costs. While the specific approaches may differ, each of these areas of research aims to make structural monitoring implementations more effective by providing greater amounts of data, either through the use of spatial sensing technologies or by lowering the costs associated with dense arrays of sensing transducers.

1.3.1 Guided Waves for Distributed Damage Detection

As SHM researchers have gone about trying to determine the best methods of damage prognosis in civil, mechanical, and aerospace structures, guided wave testing has emerged as a technique that can provide an estimate of the location, severity, and type of damage using a relatively low number of sensing transducers (Raghavan and Cesnik 2007). Guided wave testing builds on the strengths of acoustic emission (AE) methodologies, which use stress waves generated by the mechanical deformation of a monitored material to detect the presence and severity of damage locations (Holford

2009). AE techniques have shown a great deal of success in the SHM community, but suffer from the need for large numbers of sensors to be deployed on a monitored structure. Also, interpretation of data and the establishment of trigger thresholds adds some degree of subjectivity to the AE method. However, the guided wave approach relies on a controlled acoustic or ultrasonic stress wave input to the structure (initiated through a sensor-actuator pair). In thin metallic plate structures, these intentional stress waves have the ability to travel long distances. This large interrogation range offers spatial damage detection capabilities within a rather large area surrounding the actuator-sensor. As such, guided waves promise the same spatial benefits of AE without the need for large numbers of installed sensors. In addition, a controlled input ensures repeatability of the method. However, while the number of sensors in a guided wave system may be few relative to a traditional network of distributed point sensors, guided wave technology comes at a price: guided waves require a significant amount of energy to generate, and must be sampled at high rates (up to 25MHz). As a result, the per-sensor costs and data acquisition requirements hamper the market adoption of guided wave technologies.

1.3.2 Sensor Minimization for Cost-Effective SHM

One technological advancement that has made a direct impact on health monitoring over the past ten years is the development of microelectromechanical systems (MEMS) (Mohamed and Peter 2004). Referring to a set of miniaturized sensing transducers and actuators, MEMS technology brings about three main advantages to the SHM community: lowered cost, increased power efficiency, and an enhanced application range. MEMS sensors and actuators can be manufactured in bulk using an integrated

circuit-based manufacturing process, with hundreds of sensors on a given semi-conducting wafer (Gardner 1994). This approach to manufacturing is vital to keeping sensor costs low compared to macro-scale manufactured counterparts. This miniaturization also decreases the amount of electricity required to power a given transducer, lowering both the manufacturing and usage costs associated with sensors commonly used for health monitoring (such as accelerometers). Additionally, because of its small and lightweight form factor, MEMS technology can be more easily applied to lightweight structures (where sensor weight may have a significant impact on system performance), embedded within a structural material (like concrete), or deployed in dense numbers for monitoring large civil structures.

1.3.3 Wireless Sensors for Cost-Effective SHM

As previously mentioned, networks of tethered sensors have already been installed on a small number of operational civil structures around the world, but the high costs associated with tethered monitoring systems have prevented their widespread adoption. As such, the SHM community has begun to investigate the use of wireless communication as a means of increasing the affordability of large-scale sensor installations. The idea of integrating traditional sensing components with a wireless radio for the purpose of structural monitoring was first proposed over a decade ago (Straser*, et al.* 1998). Typically, by integrating a low-power microprocessor, an analog-to-digital converter, and a wireless transceiver, wireless sensor platforms can serve as data acquisition nodes capable of collecting, storing, processing, and transmitting data from traditional sensing transducers to a distant data repository. By eliminating the need for the

extensive lengths of cable required to link sensors to a central data repository, wireless sensing technologies can be deployed at both reduced costs and with higher nodal densities than traditional tethered monitoring systems. For a point of comparison, a wireless sensing system can be deployed at a cost of a few hundred dollars per sensing channel (Lynch and Loh 2006), whereas tethered monitoring systems have been known to incur costs on the order of several thousand dollars per sensing channel (Celebi 2002). As a result, a wide variety of commercial and academic wireless sensor prototypes have been developed and validated in the past ten years.

1.3.4 Problems with Data Glut in State-of-the-Art SHM Systems

Each of the state-of-the-art technologies addressed above works in some way to overcome the data limitations imposed by current in-practice SHM systems (*i.e.,* either by using a single sensor to directly quantify a spatially distributed indication of damage or by enabling the distribution a greater number of sensors to provide damage-relevant data). Unfortunately, these technologies by their very nature create an additional problem: data glut. Data glut from a dense sensor installation is essentially the inverse of the problem the dense sensing system was designed to solve (Bryson 1995). By creating an environment where hundreds, or even thousands, of channels of sensor data are being streamed to central data repository, the shear amount of data being collected and stored in these state-of-the-art SHM systems is simply too great for modern data storage and data processing techniques to efficiently handle.

1.4 Distributed Data Processing in Wireless SHM Systems

As the amount of data being collected by SHM systems continues to grow, and as data processing capabilities and techniques continue to rapidly improve across disciplines, the modern engineering community will inevitably become increasingly reliant on sensor data to provide an accurate assessment of system behavior and performance. For example, experimentally sensed data is vital to properly validating and calibrating analytical models, as well as detecting degradation and failure in engineered systems including rotating machinery (Loutas*, et al.* 2008), civil structures (Ni*, et al.* 2008), hydrological systems (Parjajka and Bloschl 2008), and aerospace vehicles (Staszewski*, et al.* 2009), among others. Traditional methods of data collection in all of these application spaces involve the use of tethered data acquisition systems. But, as discussed above, tethered monitoring systems are not feasible in large engineered systems because of the high cost of installing and maintaining large numbers of coaxial data cables. As such, networks of wireless sensors are emerging as an effective new interface between sensor and data repository (see Section 1.3.3).

In addition to the cost savings generated by the elimination of unnecessary data transfer and power cables, wireless sensing networks (WSNs) have also shown great promise because of their ability to process sensor data locally at each wireless node. In fact, the ability of wireless sensors to autonomously collect and analyze data locally has led to these devices being recently labeled as "smart" sensors (Spencer*, et al.* 2004). Local data processing is especially advantageous when confronted with the huge amounts of data commonly associated with dense networks of sensors. As such, many different architectures have been developed for embedded data processing using wireless sensors.

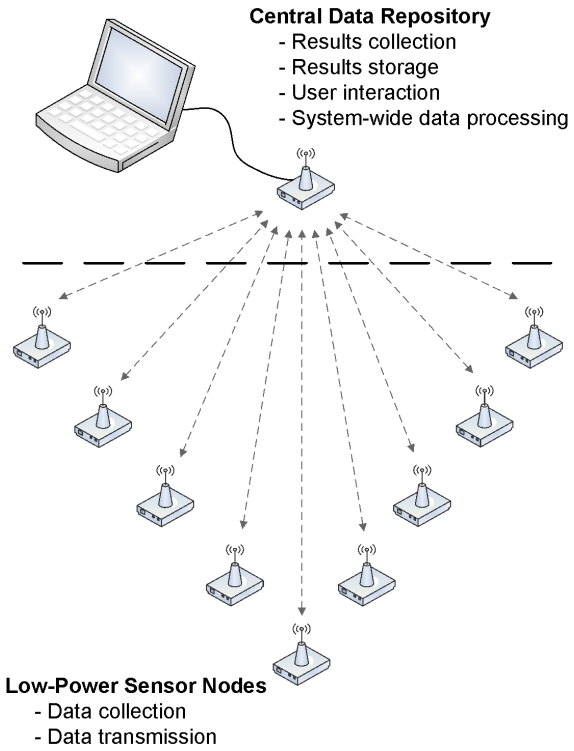


Figure 1.1: Centralized architecture for in-network data processing in a wireless sensor network.

Early on, researchers focused primarily on centralized implementations of engineering algorithms that required no communication between sensors (Figure 1.1). For example, wireless data processing architectures were developed for embedding algorithms such as such as Fast Fourier Transforms (FFTs) (Lynch 2002), autoregressive model fitting (Lynch*, et al.* 2004), and wavelet transforms (Hashimoto*, et al.* 2005) within the computational core of a network of wireless sensors.

These algorithms were designed to perform independently at the sensor, without direct sharing of data between nodes. As a result, spatial information can only be obtained at a central data repository where data processed by the wireless nodes is centrally aggregated. For example, an instrumentation of a 14-node wireless monitoring system installed on a concrete box girder bridge illustrated embedded mode shape

estimation by peak picking (Lynch*, et al.* 2006). In this study, each sensor locally calculated the frequency response function (FRF) of the instrumented system using an embedded FFT. Using this FRF, embedded peak picking logic is used to identify the primary modal frequencies. Only after this frequency information is calculated do the wireless sensors communicate the imaginary component of the Fourier spectrum at modal peaks to a central data repository where global mode shapes can be assembled.

A critical benefit gained by processing raw sensor data locally and transmitting only processed results is that the size of data to be communicated decreases drastically. Hence, these embedded data processing methods can be relatively power efficient when compared to the transfer of large tracts of time history data to a central location (Lynch*, et al.* 2004). However, in this type of sensor-centric approach, there is little to no sharing of sensor data between nodes, preventing these architectures from autonomously determining system-wide properties (such as vibrational mode shapes).

Since wireless devices can be deployed in ad-hoc networks featuring peer-to-peer communication, many analytical routines can be easily decentralized and distributed across a large number of wireless nodes with individual processing capabilities. By employing distributed or parallel processing techniques, an ad-hoc wireless sensing network can obtain spatial information without the need for a central repository. As a result, researchers have begun to look at various techniques for distributed data processing on wireless sensing networks. For example, Chintalapudi, *et al.* (2006) present a tiered system where data processing tasks can be performed on a distributed network using powerful gateway nodes (Figure 1.2). This method involves a top-down approach that allows for a flexible and highly abstracted user interface, but in which the

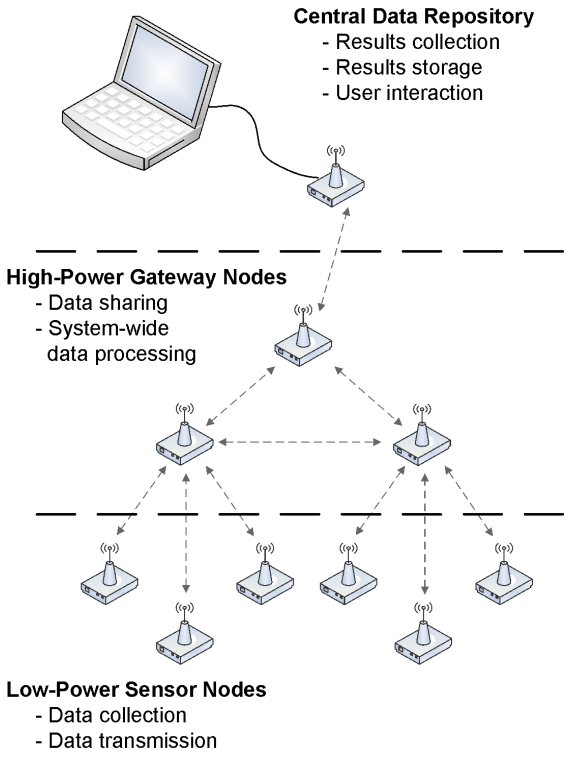


Figure 1.2: Hierarchical architecture for in-network data processing in a wireless sensor network.

computational capabilities of the prolific lower nodes are largely ignored. Other methods involving hierarchical sensing networks, such as data aggregation and fusion techniques (Gao 2005; Nagayama*, et al.* 2006; Akkaya*, et al.* 2008), and query processing (Rosemark and Lee 2005) have also been presented in the literature. These promising techniques can help improve network scalability by limiting data size and mitigating data loss problems through averaging, but they rely on a tradeoff between data size and accuracy. While wireless sensing technology has seen significant growth in recent years, additional work is still needed to modify existing analysis methods for distributed in-network execution within a network of wireless sensors.

1.5 Agent-Based Data Processing in Wireless SHM Systems

As sensing transducers and wireless sensing hardware continue to improve in capability while reducing in price, it has become reasonable to envision a future in which hundreds (or thousands) of sensing transducers can be affordably deployed on large civil structures. As mentioned above, such an explosion in sensor density may be necessary for most damage detection techniques to prove themselves viable for application to civil infrastructure. However, higher sensor density must be joined with an improvement in data collection, processing, and communication technologies in order to offset new problems associated with data glut. Based on the aforementioned benefits of in-network processing using networks of wireless sensors, it can be seen that advances in this promising area of research may prove to be incredibly beneficial to the entire SHM field.

In this dissertation, a focus is placed on the development of agent-based methods for processing sensor data in wireless structural health monitoring systems. An agent-based system can be defined as any system in which multiple intelligent agents (in this case, wireless sensor nodes) interact directly with each other and with the environment (in this case, any sensors and actuators associated with the SHM system) (Russell and Norvig 2003). In a multi-agent system, the idea is that a collection of agents, each of which has an incomplete view of its environment and acts according to its own knowledge and set of rules, can be more effective at solving a given problem than a single agent with a complete view of the world. Multi-agent systems (MAS) have been successfully applied to a large number of real world problems, including structural monitoring (Ruiz-Sandoval 2004), resource allocation (Anussornnitisarn*, et al.* 2005), online trading (O'Malley 2001), environmental monitoring (Athanasiadis and Mitkas

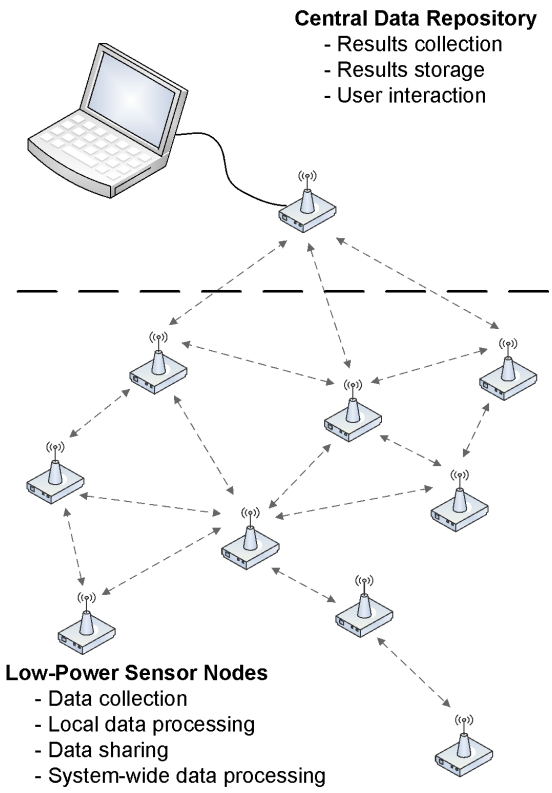


Figure 1.3: Agent-based architecture for in-network data processing in a wireless sensor network.

2004), disaster response (Boloni*, et al.* 2006), and personnel distribution (McCauley and Franklin 2002).

When applied to the processing of sensor data within a network of wireless devices, an agent-based environment allows us to distribute computational tasks across a large number of sensing nodes in a parallel fashion. As such, problems associated with power efficiency, data loss, and finite communication ranges can be minimized while providing a powerful framework for the autonomous, in-network processing of sensor data. An agent-based architecture (as seen in Figure 1.3) provides several advantages over the centralized or hierarchical data processing architectures presented in Section 1.4. Because each node in an agent-based network is free to communicate amongst its neighbors, this type of agent-based architecture retains the ability to infer spatial

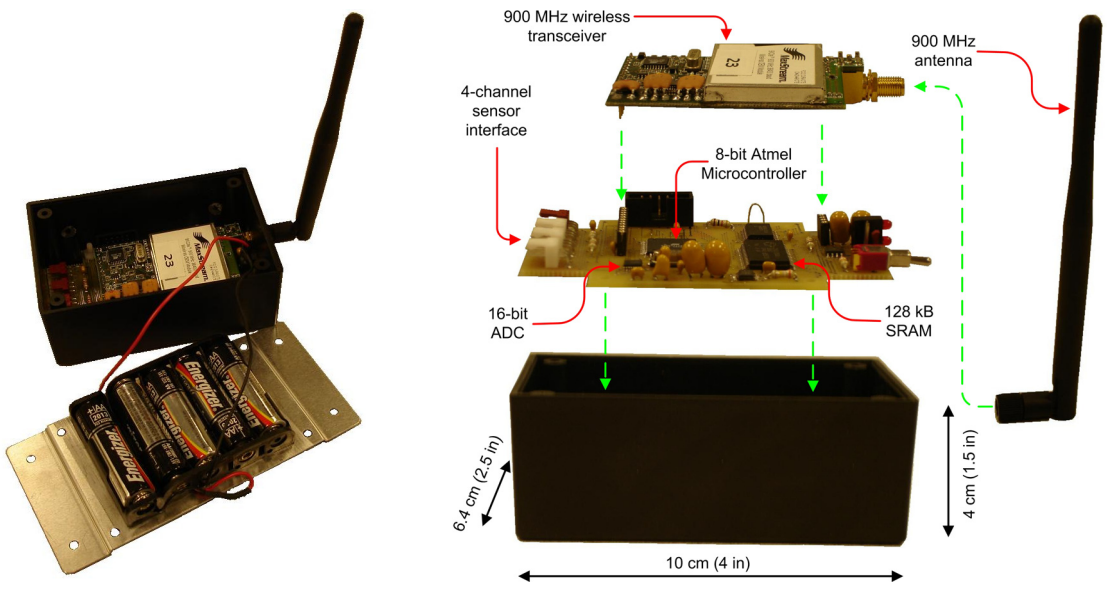
information from the sensing network (as in the case of a hierarchical system). However, because each node in an agent-based wireless network has the opportunity to play an equal part in any computational task, an agent-based framework plays directly to the strength of a WSN: its prolific low-powered nodes.

1.6 Wireless Hardware Platforms for Agent-Based Data Processing

While there are a large number of wireless sensor prototypes that are capable of being leveraged for agent-based data processing (Lynch and Loh 2006), the work presented in this dissertation has been developed for and validated on two particular wireless sensor prototypes designed specifically for structural health monitoring: the *WiMMS* wireless sensor (Wang*, et al.* 2005) and the *Narada* wireless sensor (Swartz*, et al.* 2005).

1.6.1 WiMMS Wireless Sensing Prototype

The *WiMMS* wireless sensing prototype, proposed by Wang*, et al.* (2005) and shown in Figure 1.4, utilizes a Texas Instruments ADS8341 ADC for 16-bit data collection on four simultaneous sensor channels and a Maxstream 9XCite wireless modem for communication up to 300m on the 900MHz radio band. The computational core of this sensor centers around the low-power, Atmel ATmega128 8-bit microcontroller for local data processing, and employs an external Samsung KM681000CLG-7 CMOS SRAM for an additional 128kB of SRAM for computation and data storage. This extended memory space allows the unit to store over 60,000 data points at any one time. Powered by 5 AA batteries, these units can continuously collect data for



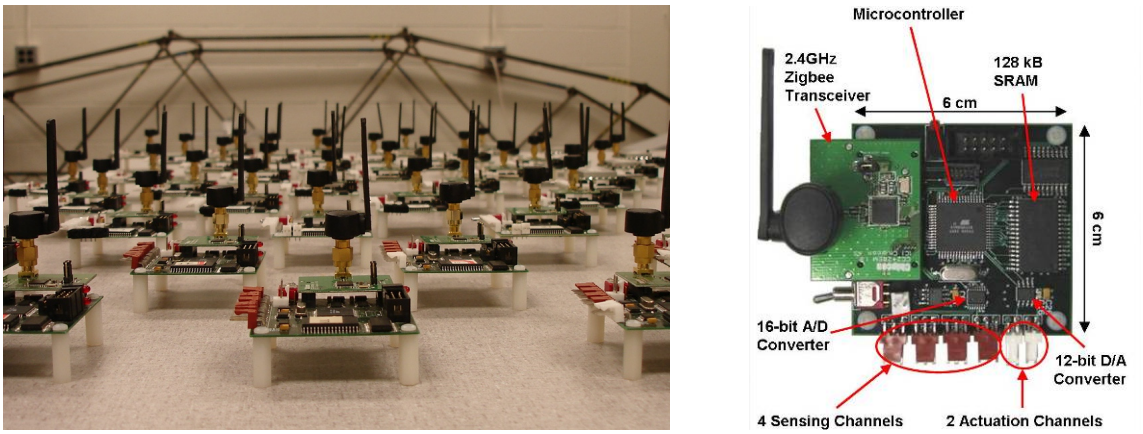
(a) (b)

Figure 1.4: WiMMS wireless sensing prototype (a) fully assembled and (b) with individual components highlighted.

up to 30 hours. Instrumentation studies of the unit on various long-span bridges have validated the accuracy of the system (Hou and Lynch 2006; Lu, *et al.* 2006; Wang, *et al.* 2006), including tight time synchronization (*i.e.,* a synchronization error of less than 5ms between nodes) (Lynch, *et al.* 2006). A rich library of data interrogation algorithms have also been included in the operating system (Lynch 2007).

1.6.2 Narada Wireless Sensing Prototype

The *Narada* wireless sensing unit, developed at the University of Michigan (Swartz, *et al.* 2005), can be seen in Figure 1.5. This wireless device, like the *WiMMS* wireless sensor, is powered by an Atmel ATmega128 microprocessor with 128kB of external SRAM and utilizes the four channel, 16-bit ADS8341 ADC for data acquisition. However, the *Narada*'s wireless communication interface consists of the Chipcon CC2420 IEEE 802.15.4-compliant transceiver, which makes it an extremely versatile unit



(a) (b)

Figure 1.5: *Narada* wireless sensing prototype (a) network and (b) schematic.

for developing large, scalable wireless sensing networks. Another distinguishing feature of the *Narada* wireless sensor is its actuation capabilities, made possible through the use of a two channel, 12-bit Texas Instruments DAC7612 digital-to-analog converter (DAC). This prototype is powered by a constant DC supply voltage of between 7 and 9 volts (the equivalent of five or six AA batteries, respectively), and has an operational life expectancy of approximately 48 hours with 6 AA batteries, given constant communication and data analysis demands. Duty cycle usage strategies can be employed to extend the life expectancy of the unit to one or more years, depending on how often the unit is out of a deep sleep state.

1.6.3 Computational Characteristics of the Atmel ATmega128

From a computing perspective, both the *WiMMS* and *Narada* wireless sensors are reliant almost exclusively on the capabilities of their ATmega128 microcontrollers. This fixed-point microprocessor is designed for low-power applications (it draws only 17.5mA at 5V when active), and has rather modest computational abilities relative to the

processing power of many higher powered microprocessors. Floating point arithmetic is possible on the ATmega128 through support provided by the compilation environment. On both the *WiMMS* and *Narada* platforms, the ATmega128 is coupled with an 8MHz external clock source, and almost all available instructions execute within a single clock cycle. For comparison purposes, it is useful to note that a 4096-point complex valued FFT can complete on either of these platforms in approximately 20 seconds.

From a memory perspective, the ATmega128 contains 128kB of in-system reprogrammable flash, 4kB of EEPROM and 4kB of internal SRAM. The *WiMMS* and *Narada* platforms both supplement this storage space with 128kB of external SRAM, which is enough to store over 32,000 floating point values.

1.7 Research Objectives and Dissertation Outline

In this dissertation, several novel agent-based computational architectures for distributed in-network data processing are presented and evaluated in the context of SHM (Figure 1.6). These architectures will allow dense wireless SHM systems to collect, store, and autonomously process large amounts of sensor data, eliminating common problems associated with sensor distribution, power consumption, and data glut and providing a powerful framework for performing complex data analyses in-network. Each successive architecture developed herein moves farther away from the current reliance on a centralized architecture for in-network computing and towards an agent-based paradigm where network computing demands can be handled autonomously (and optimally) without the need for human interaction.

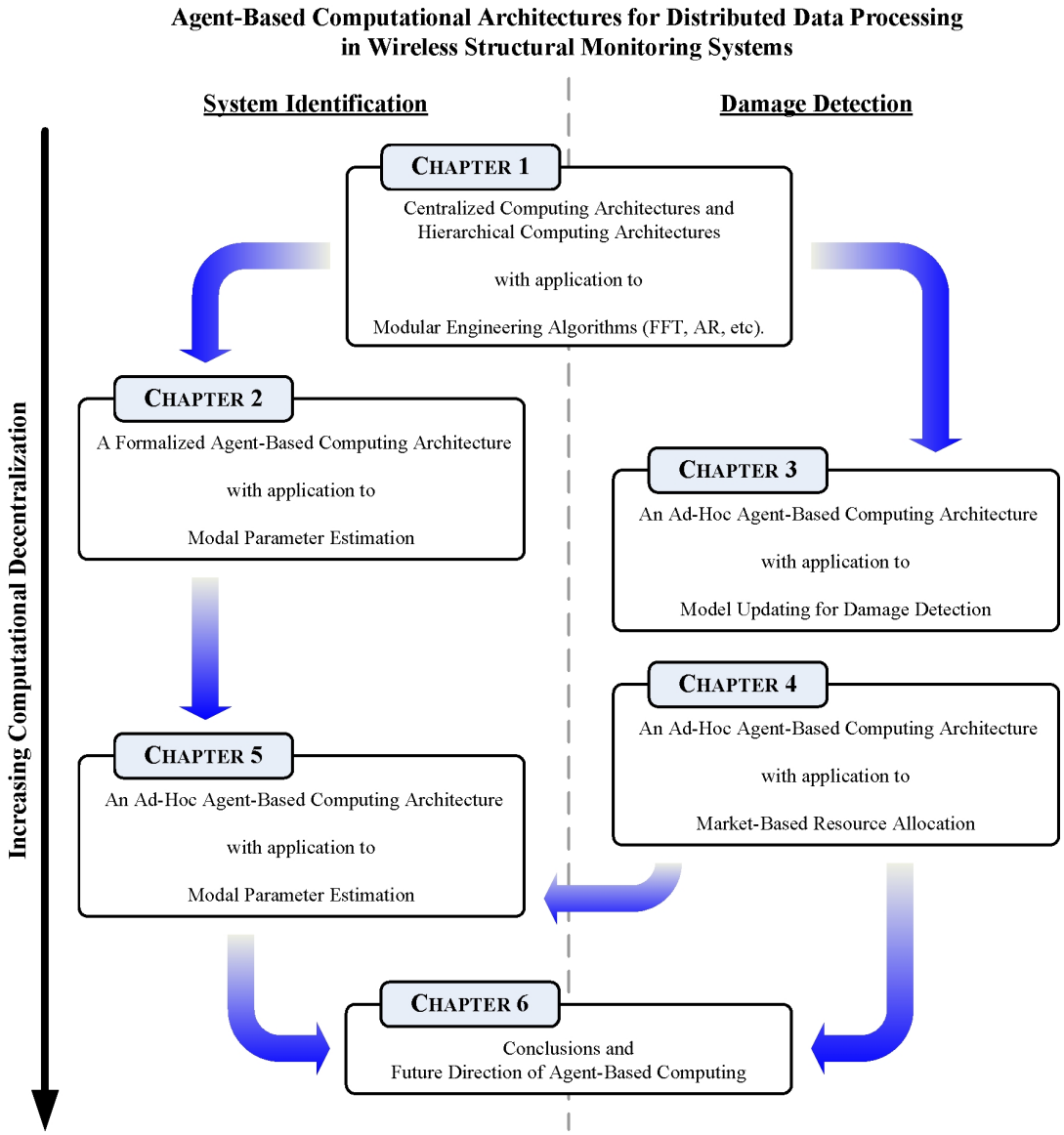


Figure 1.6: A schematic representing major topics covered in this dissertation.

In *Chapter 2: Automated Modal Parameter Estimation by Parallel Processing within a WSN*, a set of distributed computing techniques are developed for automated in-network estimation of modal parameters (modal frequencies using peak picking, damping ratios using the random decrement method, and mode shapes using the frequency domain decomposition algorithm) given a set of wirelessly collected sensor data. These distributed techniques are evaluated by deploying a network of wireless sensors on both

the main balcony of a large theatre located in Detroit, Michigan and on a pedestrian bridge located in Ann Arbor, Michigan. In each case, the network of sensors is shown to be capable of automatically and accurately estimating modal properties of a real world structure.

In *Chapter 3: A Parallel Simulated Annealing Architecture for Model Updating within a Wireless Sensor Network*, a distributed architecture is developed for the in-network updating of structural models using a parallel simulated annealing stochastic optimization technique. This distributed method could potentially be used to validate design assumption, test analytical models, or even to detect the onset of structural damage or degradation. It is validated on a wireless sensor network by successfully updating a 6-DOF dynamic structural model with unknown mass, stiffness, and damping properties.

In *Chapter 4: Market-Based Resource Allocation for Distributed Data Processing in Wireless Sensor Networks*, the distributed in-network data processing paradigm developed in Chapter 3 is expanded through the application of market-based techniques. Using the *n*-Queens problem as a basis for validation, it is shown that the use of market principles to assign computational resources to multiple, simultaneously processed computational tasks allows for computational optimization in the wake of competing objectives such as power consumption, memory usage, and time to completion.

In *Chapter 5: Market-Based Frequency Domain Decomposition for Automated Mode Shape Estimation in Wireless Sensor Networks*, the market-based architecture of Chapter 4 is applied to the distributed modal identification techniques of Chapter 2 in order to improve the accuracy of the distributed frequency domain decomposition mode

shape estimates. Lastly, in *Chapter 6: Conclusions*, a summary of the previous four chapters, as well as an outline for potential future work in the area of distributed in-network data processing using agent-based methods is presented.

CHAPTER 2

AUTOMATED MODAL PARAMETER ESTIMATION BY PARALLEL PROCESSING WITHIN A WSN

2.1. Introduction

One area in which agent-based computing techniques can be applied to SHM is through the development of in-network system identification methods than can enable a WSN to autonomously determine the modal properties of a monitored structure. Traditional centralized modal estimation methods are inadequate when applied to the wireless environment, as they require large amounts of data to be transmitted to a centralized location. As such, problems associated with wireless bandwidth arise, restraining network scalability and limiting the spatial diversity and resolution of the resulting modal estimates. In this chapter, which is modified from Zimmerman *et al.* (2008a), a significant first step is made toward the complete automation and decentralization of SHM techniques within agent-based networks of wireless sensors. Specifically, a suite of parallel methods for in-network modal parameter estimation is developed. While these methods are not completely agent-based (*i.e.,* they rely on a fixed and preassigned topology), they move away from existing centralized and hierarchical processing techniques by truly parallelizing computation over an entire WSN, increasing

network scalability and robustness while improving the capability of a wireless network to determine spatial information about the dynamic response of a monitored structure.

The idea of identifying system parameters from dynamic response data originated two decades ago within the mechanical and aerospace engineering communities (Ewins 1986; Ljung 1987; Juang 1994). The subsequent development of a set of system identification techniques was fueled largely by the need for analytical tools that could be used to build effective models of dynamic physical systems from observed system data. For obvious reasons, the ability to experimentally extract system parameters from sensor data offers enormous benefits across all engineering disciplines. In civil engineering, the ability to ascertain modal information (modal frequencies, mode shapes, and damping ratios) from sensor data has paved the way for the assessment of structural performance and the calibration of analytical design models (Alampalli 2000). In some instances, modal parameters can even be used to detect and locate structural damage in the wake of natural events like earthquakes (Doebling*, et al.* 1998).

In the aerospace and mechanical engineering communities, modal parameter identification techniques are typically carried out using both input and output measurement data, which can be related through frequency response functions (FRFs) in the frequency domain. However, it is often difficult to excite a large civil structure in a controlled manner with measurable input excitation forces. Thus, modal parameter estimation techniques using output-only dynamic data have become quite popular within the civil engineering field (Cunha and Caetano 2006). Many of these approaches assume the input driving the system dynamics is broad-band white. In order to extract meaningful system properties from a large civil structure, a large amount of data must be available

from a dense array of sensors. Fortunately, recent advances in low-cost wireless sensing technologies have made the dense instrumentation of large civil structures possible. As a result, it is important for researchers to translate traditional system identification techniques to a distributed setting for use in wireless sensing networks.

In this chapter, three output-only modal identification techniques are adopted and modified for use within a distributed wireless sensing network: the peak picking (PP) method, the random decrement (RD) method, and the frequency domain decomposition (FDD) method. This work sets itself apart from current work in distributed data processing using wireless sensors by leveraging the parallel data processing environment available within large sensing networks. Parallel and distributed computing minimizes the need for inter-process wireless communication which is more power consuming than local processing. This parallel approach, while still addressing problems associated with power consumption and bandwidth, allows a WSN to employ typical offline modal analysis techniques to autonomously extract spatial modal information from a large network of sensors without the need for a central data repository. In order to validate the performance of these embedded algorithms, two field validation studies are performed. In the first, the cantilevered balcony of a historic theater in metropolitan Detroit is instrumented with a dense network of wireless sensing prototypes. Over the span of several vibration tests, acceleration response data from the balcony is collected by the wireless network. Using the stored data, each of the distributed modal identification techniques is executed to estimate the modal properties of the system. In the second field test, a pedestrian bridge in Ann Arbor, Michigan is instrumented with a network of wireless sensing prototypes, and the distributed computing techniques are again used to

estimate modal properties in-network. In both field tests, results from the embedded algorithms are compared with modal analysis techniques run off-line using the time history data collected by the WSN.

2.2 Distributed Output-Only Modal Identification in a WSN

In general, it is very difficult to excite a large civil structure in a controlled manner. As a result, several output-only modal estimation methods have been adopted for common use in structural system identification. In this section, three of these methods are modified for a distributed setting and implemented on a network of wireless sensing prototypes. The first method is the peak picking (PP) method (Ewins 1986; Allemang 1999). This frequency domain method is commonly used in civil engineering because of its simplicity. The second method is the frequency domain decomposition (FDD) technique (Brincker*, et al.* 2001b), which is similar to peak picking but is much more robust when dealing with closely spaced modes. The third method is the random decrement (RD) method (Cole 1968; Ibrahim 1977). In a multiple degree of freedom system, this technique is dependent upon previous knowledge of the system's modal frequencies (which can be provided by the PP algorithm), but it offers a superior way of determining accurate estimates for modal damping values. In this section, the theory behind each of these methods and their distributed implementation within a wireless sensing network is described in detail.

2.2.1 The Peak Picking Method

The peak picking (PP) method is the simplest known technique for estimating the modal properties of a structure from system output data. This method, like many other output-only techniques, assumes that the immeasurable excitation input can be characterized as zero-mean Gaussian white noise. In civil engineering applications, this type of excitation is generally achieved using either impulse or broad-band ambient vibration loading conditions. PP analysis is based on the fact that the frequency response function (FRF) of a given system will experience extreme values around that system's modal frequencies (Ewins 1986). Assuming a white noise excitation, the FRF of a structure at sensor location k, $H_k(j\omega)$, can be considered equivalent to the Fourier spectrum of the response data collected at that sensor. This spectrum can be formulated by converting measured accelerations to the frequency domain using a fast Fourier transform (FFT).

If a structure is lightly damped with well separated modes, operational mode shapes can also be determined with the PP method using the system's FRFs (Allemang 1999). The imaginary component of an FRF at modal frequency ω_i, at sensor locations 1 through n, can be assembled to yield the i^{th} mode shape, ϕ_i, as follows: $\phi_i = \{\text{imag}[H_1(j\omega_i)] \cdots \text{imag}[H_n(j\omega_i)]\}^T$. From the perspective of a wireless sensing network, this method is relatively easy to implement in a decentralized fashion. In this implementation, the user first specifies the maximum number of peaks, p, that should be identified. Then, a consistent set of acceleration time history data is collected at each sensing node and converted to an FRF using an embedded version of the Cooley-Tukey FFT algorithm. Each node picks the p largest peaks from its frequency response function

by scanning for frequencies at which the value of the FRF is significantly and consistently higher than the value of the FRF at surrounding frequencies. If less than p peaks are found, zeros will be returned in place of the missing peaks. This algorithm assumes that there are no closely spaced modes and thus can only detect peaks separated by at least 10 points in the frequency spectrum. Because some sensing nodes may not be capable of detecting peaks at all modal frequencies due to positioning or poor data, it is necessary to transmit peak information to a central node that can view the individual PP results for the network as a whole. It should be noted that every wireless sensor communicates its identified peaks (p of them) to the central node; hence, the amount of data to be transmitted is fixed. By tabulating the periodicity at which a given frequency has been "picked" by nodes on a network, this central node can infer a subset of p (or fewer) reasonable modal frequencies from the original PP data. Once the central node has determined a global set of peak frequencies, it can then share its findings (namely modal frequencies) with the rest of the network, and the imaginary components of the FRFs at the picked frequencies can be broadcast from each sensor to the rest of the network, one sensor at a time. This sharing of data provides all wireless sensor nodes with operational deflection shapes (ODS), which are correlated to system modes. If the system input is Gaussian white noise, then the ODSs are equivalent to true mode shapes. If necessary, other local data (such as time histories or frequency spectra) can be subsequently communicated by each wireless sensor to a central server in the network. A graphical representation of the implementation of the PP method on a distributed network of wireless sensors can be seen in Figure 2.1.

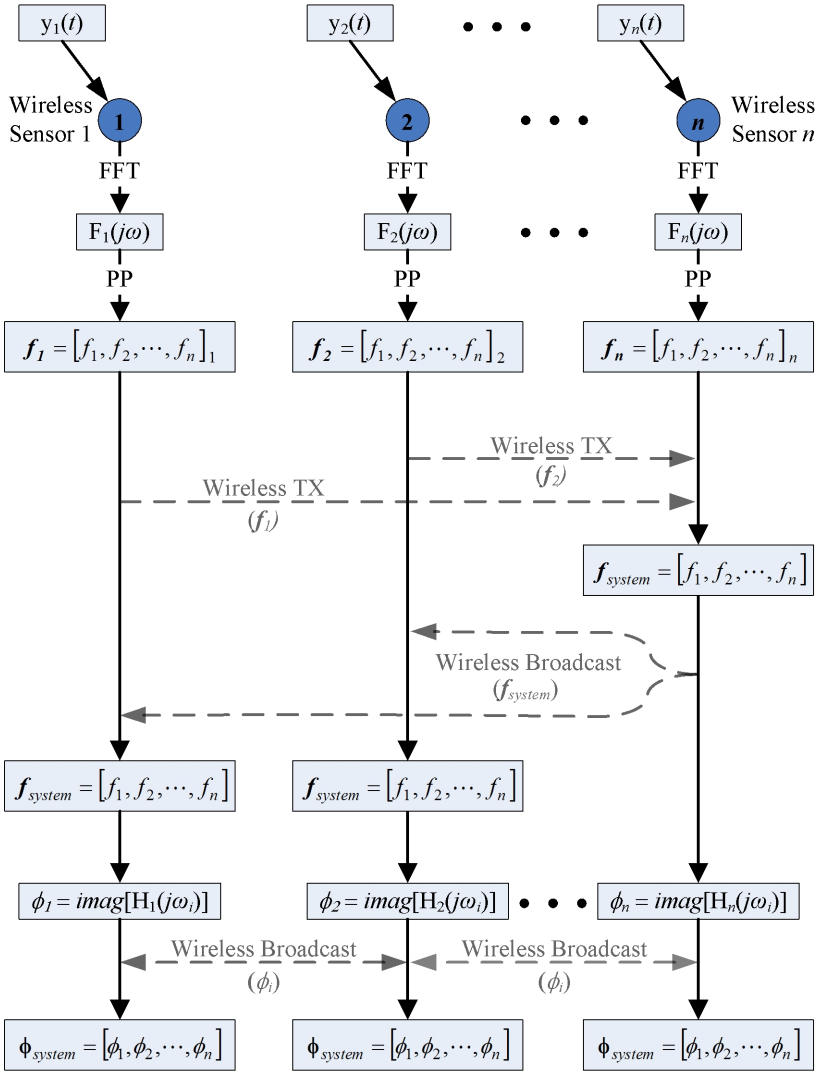


Figure 2.1: Implementation of the peak picking method on a network of wireless sensors.

By limiting the amount of communication between individual sensing units, this approach drastically limits the amount of bandwidth needed for wireless data transmission. For example, if in a centralized sensing network 20 wireless sensors are used to send data to a central server for modal estimation, then 4096 data points are transmitted resulting in 163,840 bytes being transmitted (each point is a two-byte (16-bit) integer sampled by the ADC). If the central server communicates modal information to each node in a peer-to-peer configuration, an additional 7,040 bytes are transmitted

(bringing the total number of bytes to 170,880). However, should the central server be able to broadcast to the entire network, then only an additional 352 bytes need to be transmitted (bringing the total number of bytes to 164,192).

In a similar scenario using the parallel in-network approach to PP outlined above, the same results can be obtained by transmitting a total of only 2,128 bytes of data. In this method, if 19 nodes each send four peak frequencies to a central node, then 340 bytes are communicated. Then, by peer-to-peer communication, the central node would send the final modal frequencies back to the original 19 nodes (requiring an additional 340 bytes). Once each node knows what the network has decided the modal frequencies are, each node in the peer-to-peer network can then communicate the imaginary components of their frequency response function allowing each node to assemble the four mode shapes of the structure. This requires 1,520 bytes to be communicated in the wireless sensor network. If ideal broadcasting is possible, this approach can be further reduced to require only 640 bytes of communication. A summary of this detailed breakdown can be found in Table 2.1. This method is also advantageous because it is relatively simple to implement on a sensing network, and it utilizes engineering algorithms that can be processed quickly. However, there are several drawbacks to distributed PP analysis. Primarily, peak picking is always a subjective practice, and it is therefore difficult to implement perfectly in software. Additionally, peak picking does not properly handle closely spaced modes, for which other methods may be more appropriate.

Table 2.1: Summary of wireless data transmission needed in a network with twenty nodes where 4096 data points are used to calculate modal information for four modes.

Method	Transmission	Payload Type	Bytes	Results	Assumption
Centralized Server	4096 shorts x 20 nodes	Time history data to server from each node	163,840	$\{a\}$	
	4 floats x 20 nodes	4 damping ratios from server to each node	320	$\{\xi_i\}$	
	80 floats x 20 nodes	4 mode shapes from server to each node	6,400	$\{\phi_i\}$	
	4 floats x 20 nodes	4 frequencies from server to each node	320	$\{f\}$	
		TOTAL:	170,880		
Decentralized Peak Picking (PP)	4 floats x 19 nodes	4 frequency peaks to central node	304	$\{f\}$	
	4 floats x 19 nodes	4 modal frequencies back each node	304	$\{f\}$	
	4 floats x 20 nodes x 19 nodes	Imaginary components with each node (20) sending to every other node (19)	1,520	$\{\phi_i\}$	
		TOTAL:	2,128		
Decentralized Frequency Domain Decomposition (FDD)	8 floats x 19 nodes	Spectral value at each mode from one node to a neighboring node	608	$A(\omega)$	$\{f\}$ *previously known*
	8 floats x 18 nodes	2 node mode shape to central node	576	$\{\phi_i\}$	
	80 float x 19 nodes	4 stitched mode shapes back to each node	6,080	$\{\phi_i\}$	
		TOTAL:	7,264		
Decentralized Random Decrement (RD)	4 floats x 19 nodes	4 identified frequencies to central node	304	$\{f\}$	$\{f\}$ *previously known*
	4 floats x 19 nodes	4 identified damping ratios to central node	304	$\{\xi_i\}$	
	4 floats x 19 nodes	4 modal frequencies back each node	304	$\{f\}$	
	4 floats x 19 nodes	4 modal damping ratios back each node	304	$\{\xi_i\}$	
		TOTAL:	1,216		

NOTE: short = 2 bytes; float = 4 bytes

2.2.2 The Frequency Domain Decomposition Method

The frequency domain decomposition (FDD) technique, developed by Brincker, *et al.* (2001b) maintains most of the advantages of other classical frequency domain methods, such as peak picking. However, the FDD technique approximately decomposes the spectral density matrix into a set of single degree of freedom (SDOF) systems using singular value decomposition (SVD), allowing close modes to be identified with

relatively high accuracy. In this method, the relationship between measured responses $y(t)$ and unknown inputs $x(t)$ can be expressed as:

$$\boldsymbol{G}_{yy}(j\omega) = H^{*}(j\omega)\, \boldsymbol{G}_{xx}(j\omega)\, H(j\omega)^{T} \tag{2.1}$$

where $\mathbf{G}_{yy}(j\omega)$ is the ($m \times m$) power spectral density (PSD) matrix of the responses, $\mathbf{G}_{xx}(j\omega)$ is the (1 × 1) PSD matrix of the input, $H^{*}(j\omega)$ is the complex conjugate of the (m × 1) FRF matrix, $H(j\omega)^{T}$ is the transpose of the (m × 1) FRF matrix, and m is the number of output degrees of freedom.

In the FDD method, the first step is to obtain an estimate of the output PSD matrix, $\hat{\boldsymbol{G}}_{yy}(j\omega)$ for each discrete frequency $\omega = \omega_i$. This can be done by creating an array of FRFs using FFT information from each degree of freedom in a system:

$$\hat{\boldsymbol{G}}_{yy}(j\omega i) = \{F_y(j\omega_i)\}\{F_y{}^{*}(j\omega_i)\}^{T} \tag{2.2}$$

where $\{F_y(j\omega_i)\}$ is an array of FFT values for each degree of freedom at a given frequency ω_i and $\{F_y{}^{*}(j\omega_i)\}^{T}$ is the complex conjugate transpose (Hermitian matrix) of that array (Allemang 1999).

The second step in the FDD process is to extract singular values and singular vectors from the PSD of the response by taking the singular value decomposition (SVD) of the matrix $\hat{\boldsymbol{G}}_{yy}(j\omega)$:

$$\hat{\boldsymbol{G}}_{yy}(j\omega_i) = \boldsymbol{U}_i\, \boldsymbol{S}_i\, \boldsymbol{U}_i{}^{H} \tag{2.3}$$

where the matrix $\boldsymbol{U}_i = [u_{i1}, u_{i2}, \ldots, u_{im}]$ is a unitary matrix holding singular vectors u_{ij}, $\boldsymbol{S}_i$ is a diagonal matrix holding the scalar singular values s_{ij}, and $\boldsymbol{U}_i^H$ is the Hermitian matrix of $\boldsymbol{U}_i$. Near a peak in the PSD function corresponding to a given mode in the spectrum, this mode or a possible close mode will be dominating. Thus, the first singular vector, u_{i1}, can be an estimate of the mode shape ϕ_i: $\hat{\phi}_i = u_{i1}$. An extension of the FDD method that allows for the detection of additional modal information (*i.e.,* modal frequencies and damping ratios) is often called enhanced frequency domain decomposition (EFDD), and was originally proposed by Brincker*, et al.* (2001a). However, in this chapter the FDD method will only be used to determine system mode shapes.

Unfortunately, because of the need to store and manipulate the output power spectral density matrix for each degree of freedom in a system, the implementation of a centralized FDD method requires a significant amount of memory relative to the PP method. On a wireless sensing network where there are heavy constraints on the amount of available storage at each sensing node, an alternate decentralized method is proposed and implemented. The key feature of this approach is that mode shapes are determined by creating a collection of overlapping two-node modes and stitching them together after computation is complete.

First, the wireless sensing network collects a synchronized set of time history acceleration data. This data is then transformed to the frequency domain via an embedded FFT algorithm, and the aforementioned embedded PP technique is employed to identify modal frequencies at each node on the network. Peak picking results are then transmitted wirelessly to a central node, where a final set of modal frequencies are decided upon and shared among the nodes in the network. At this point, every unit in the network transmits

its complex FFT results corresponding to the picked modal frequencies to the next unit in a pre-determined chain (except the last unit in the chain, which has no successor). Using this shared data, all but one of the sensing nodes (the first in the chain) is able to construct a two degree of freedom output PSD matrix at each modal frequency using the two sets of FFT results. After each wireless sensor performs an SVD on the PSD matrix, two-node mode shapes are extracted from the resulting singular vectors at the frequencies previously determined by PP. Finally, all two-node mode shapes are transmitted back to a central node, where they are recombined to form full system mode shapes; global mode shapes are then shared with the entire network. A graphical representation of this decentralized FDD method embedded within a network of wireless sensors can be seen in Figure 2.2. It is also possible to extract damping information from the SVD results for each two-node mode shape by performing an embedded inverse FFT (IFFT) on its SDOF PSD function and calculating its logarithmic decrement. In this study, however, all FDD damping estimates are performed offline. If desired, a user can also request complete recorded time histories, FFT information, and complete SVD results from each unit.

This approach requires slightly more wireless communication than the PP method. Assuming modal frequencies have been previously identified, the decentralized FDD method requires a total of 7,264 bytes to be communicated in a 20 node network, as summarized in Table 2.1. In the current implementation, data is communicated by peer-to-peer communication links. However, the number of bytes to be communicated could be reduced to 1,504 if the central node is able to broadcast the global modes to all of the network nodes as opposed to one at a time as is currently implemented. As seen in Table 2.1, these numbers are significantly less than the 170,880 bytes of data required in the

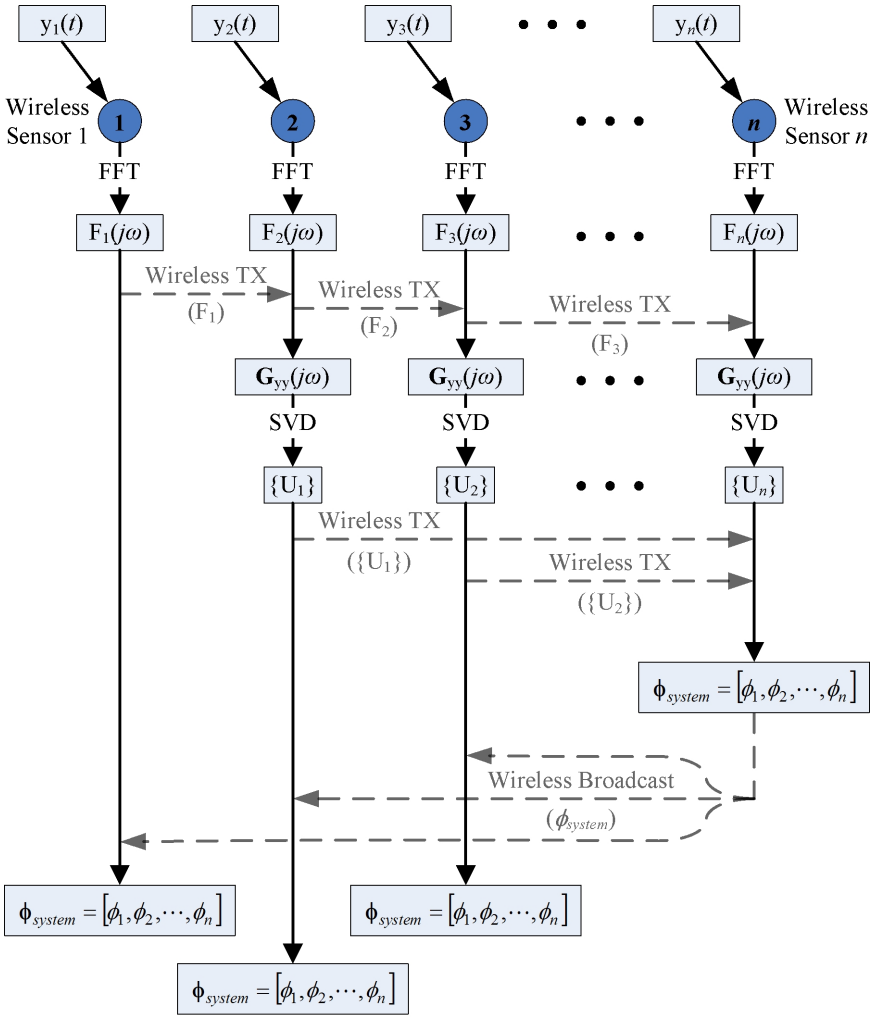


Figure 2.2: Implementation of the frequency domain decomposition method on a network of wireless sensors, assuming previous knowledge of modal frequencies.

centralized setting. As such, while the distributed FDD analysis technique presented above requires significantly more computation than does peak picking, it is effective in limiting the amount of data transmission necessary to ascertain modal frequencies and mode shapes. In addition, the implemented FDD method provides more reliable and robust mode shape estimates compared to PP, especially in the case of closely spaced modes. Lastly, because all FFT and SVD computation is performed simultaneously in a parallel fashion, significant time savings can be realized from the parallel

implementation. As a result, this method can be made scalable to an almost infinite number of nodes.

2.2.3 The Random Decrement Method

The random decrement (RD) technique is based upon the concept of the "random decrement signature," proposed initially by Cole (1968), and explored in greater detail by Ibrahim (1977) and Asmussen (1997). This concept essentially states that the response of a single degree of freedom structure due to a random input is composed of a deterministic impulse part and a random part with an assumed zero mean. Thus, by averaging enough samples of the same random response, the random part will average out, leaving only the deterministic part of the signal. In order to avoid averaging out the deterministic part of the signal, random decrement analysis consists of averaging N windows of length τ. Each of these windows must always start with one of the following:

a) A constant level, which yields the free decay step response.
b) Positive slope and zero level, which yields the free decay positive impulse response.
c) Negative slope and zero level, which yields the free decay negative impulse response.

Thus, if $y(t)$ is the random response, the free decay impulse response, $x(t)$, can be written as:

$$x(\tau) = \frac{1}{N}\sum_{n=1}^{N} y(t_n + \tau) \tag{2.4}$$

with the condition $t = t_n$ when $y(t)$ is equal to y_s (a constant level), or when $y(t)$ is equal to 0 and dy/dt is non-zero.

The response resulting from the application of the random decrement signature technique is equivalent to the free decay response of the structure. From this free response function, modal frequencies can be extracted by examining zero crossings; modal damping can also be estimated using the logarithmic decrement of the decay function. In a multiple degree of freedom structure, the random decrement response for each mode can be calculated by taking the time history response of the structure to the frequency domain, and filtering out all frequencies that do not correspond to a given mode. In essence, the spectrum surrounding a mode is kept $F_j \in [\omega_i - \Delta\omega, \omega_i + \Delta\omega]$ where ω_i is the i^{th} modal frequency and $\Delta\omega$ defines the region of F retained.

In this study, a distributed RD algorithm is designed and embedded within the computational core of a network of wireless sensors so that the network can autonomously estimate modal frequencies and damping ratios. For this algorithm, a set of consistent time history acceleration data is first collected at each sensing node. Each node in the network then transfers this data to the frequency domain using an embedded FFT. Employing a frequency window provided by the user (or calculated from prior peak picking information), frequencies irrelevant to a given mode are filtered out, and the signal is taken back to the time domain using an embedded inverse FFT (IFFT). This window is specific to one modal frequency, and thus the RD process must be repeated for each mode. At this point, a summation trigger y_s (which is also designated by the user) is

used within each sensing node to create a number of samples for random decrement averaging. These samples are converted into a SDOF free decay impulse response function by applying the concepts in Equation 2.4. Zero crossing and logarithmic decrement techniques are employed to automatically extract modal frequency and damping information from the impulse response. These parameters, calculated independently in each node in the network, can then be sent wirelessly to a central node where a system-wide modal frequency and damping ratio can be determined using statistical measures and broadcast to the network. A graphical representation of the distributed RD algorithm can be found in Figure 2.3. Note that it is also possible to extract mode shapes using embedded RD analysis by choosing a common lead node with which to trigger the RD averaging (Ibrahim 1977). However, in this study, the RD method is only used to calculate modal frequencies and damping ratios.

Much like the embedded PP method, this decentralized RD technique greatly limits the amount of data needed to be transmitted wirelessly. In a network with twenty nodes, where 4096 points of data from each sensor are being used to calculate modal information for four distinct modes, the decentralized RD method presented above requires only 1,216 bytes of data to be transmitted wirelessly. However, it does rely on previous knowledge of approximate modal frequencies. Thus, as seen in Table 2.1, this method, when used in conjunction with the decentralized PP algorithm found in Section 2.2.1, can provide accurate estimates of modal frequencies and damping ratios by transmitting a total of only 3,344 bytes of data. In a wireless network allowing for ideal broadcasting, this number can be reduced to a mere 1,040 bytes. This is a significant improvement over the requirements of the centralized setting. The decentralized RD

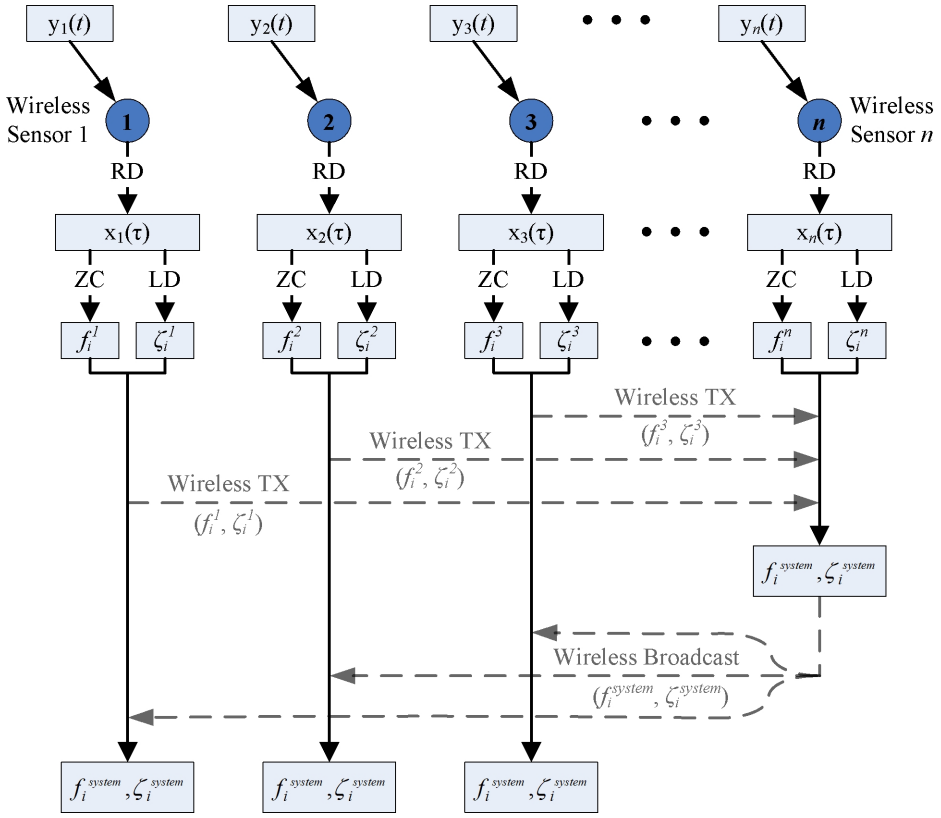


Figure 2.3: Implementation of the random decrement method on a network of wireless sensors, assuming previous knowledge of modal frequencies.

method is also rather simple to implement on a wireless sensing network and utilizes engineering algorithms that can be processed quickly. This method provides accurate estimates of modal damping ratios by taking advantage of the great degree of redundancy available within a sensing network. However, in a multiple degree of freedom system, prior knowledge of the frequency characteristics of the system (possibly obtained from an embedded PP analysis) is required in order to properly window the Fourier spectrum. This method is also not suited to determining modal properties involving closely spaced modes.

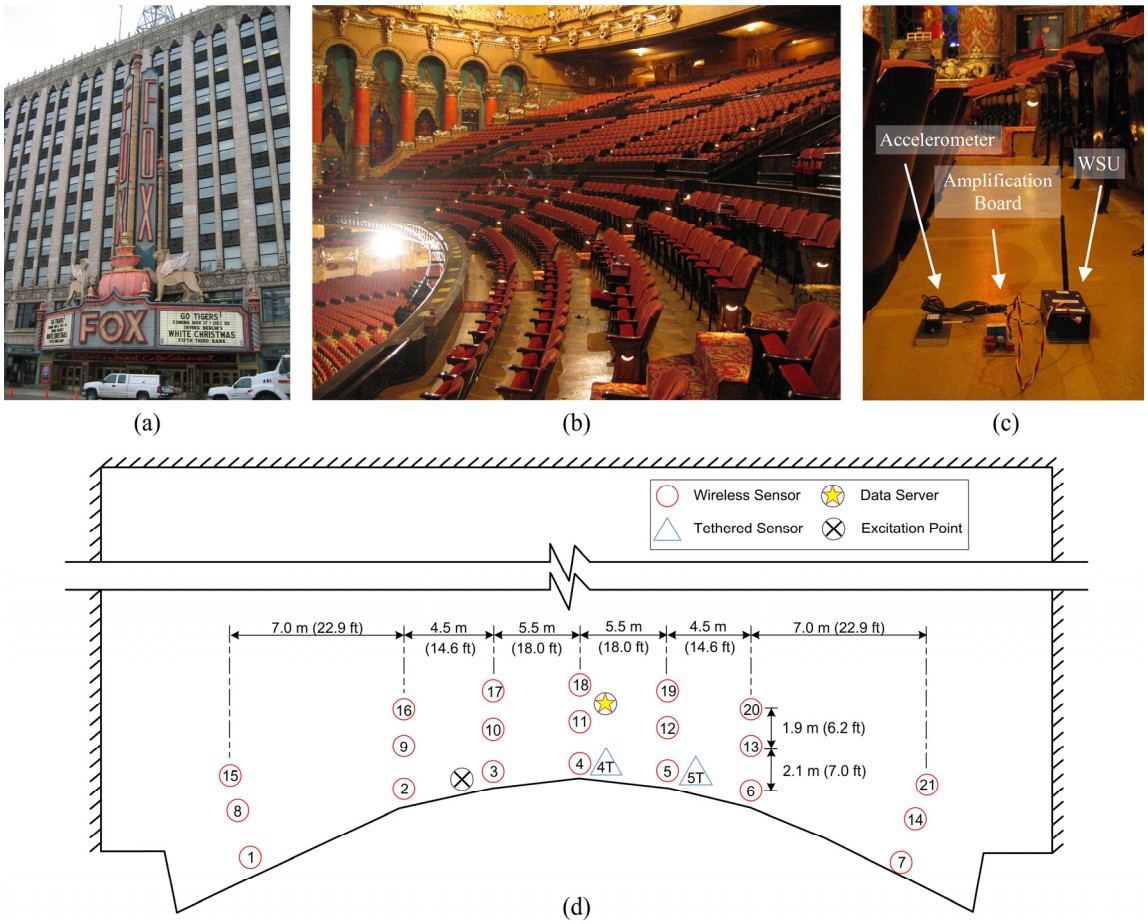


Figure 2.4: Wirelessly instrumented theater balcony: (a) theatre, (b) main balcony, (c) a typical wireless sensor layout, and (d) location of wireless and tethered accelerometers (not drawn to scale).

2.3 Theater Balcony Testbed

A historic theater, located in metropolitan Detroit, Michigan, is selected as an appropriate structure to validate the embedded algorithms proposed for use within a wireless sensing network. This theater is one of the largest in the United States, and is part of a large complex which includes several theater service areas and an attached office building. The auditorium itself has two balconies: a main balcony located at the fifth floor level of the building, and a loge balcony located at the third floor level. The main balcony, shown in Figure 2.4a, is chosen for instrumentation purposes. This balcony is approximately 50m (150ft) wide, and is structurally supported only at the rear and

sides of the auditorium. As a result of its long unsupported span, the theater's balcony is known to suffer from humanly perceptible vibrations (Setareh 1990).

2.3.1 Instrumentation and Excitation Strategy

On February 2, 2007, the front section of the main balcony of the theater (specifically the first five rows within a 3m (15ft) band of the balcony edge) was instrumented using a network of *WiMMS* wireless sensors (described in detail in Section 1.6.1). In this study, twenty-one wireless sensing units were installed in a seven-by-three grid, with seven units distributed evenly across the span of the balcony in each of rows 1, 3, and 5. The location of these sensing units is shown in Figure 2.4c. Attached to each wireless sensing unit was either a PCB Piezotronics 3801D1FB3G or Crossbow CXL02LF1Z MEMS capacitive accelerometer; each was oriented to monitor the vertical acceleration of the balcony. The sensitivity of the PCB accelerometer is 0.7 V/g and its dynamic range is 3g, peak-to-peak. The sensitivity of the Crossbow accelerometer is 1.0 V/g and its dynamic range is 2g, peak-to-peak. To improve the performance of the wireless monitoring system, a signal conditioning circuit proposed by Lynch*, et al.* (2006) was included with each sensor to both amplify and band-pass (0.02 – 25 Hz) acceleration response data before inputting to the wireless sensor's ADC. This circuit essentially amplifies the accelerometer output so that the noise floor of the accelerometer controls the data quality as opposed to the quantization error of the ADC; this is especially useful for ambient structural accelerations. To verify the integrity of the wireless monitoring system, two additional tethered acceleration channels were monitored using a cable-based Freedom Data Acquisition System PC from Olson

Instruments, Inc.; this system comes equipped with its own data acquisition software. Internally, data acquisition is accomplished using a National Instruments 1.25 MS/s, 16 channel 12-bit PCI Data Acquisition Card. Tethered Dytran accelerometers (models 3165A and 3116A) were employed with the Freedom system. These accelerometers have sensitivities of 1.0 V/g with a dynamic range of ±5g. As seen in Figure 2.4c, the locations of the tethered sensors are collocated with wireless sensors 4 and 5; as a result, tethered sensors are denoted as "4T" and "5T", respectively.

Because all three output-only identification methods previously presented assume a broadband white input, an appropriate method of excitation had to be adopted for testing. For the purposes of this study, impulsive excitation was delivered using a simple heeldrop test. This type of loading is performed by one of the authors quickly raising and dropping both heels simultaneously. This approach to excitation is typically thought to mimic an impulse load. The location of this heeldrop loading was between sensor 2 and 3 at the front of the balcony, as seen in Figure 2.4d.

2.3.2 Experimental Results

On the day of testing, a set of nine nearly identical tests (denoted as runs #1-9) were run using impulse loadings generated by a single person weighing 82kg (180lb) and performing a heeldrop. The objectives of these tests were to validate the accuracy of the wireless data acquisition system and to compare the ability of the proposed distributed modal identification methods to accurately determine the balcony's modal parameters (modal frequencies, damping ratios, and mode shapes) using the embedded processing capabilities residing on the spatially distributed network of wireless sensor nodes.

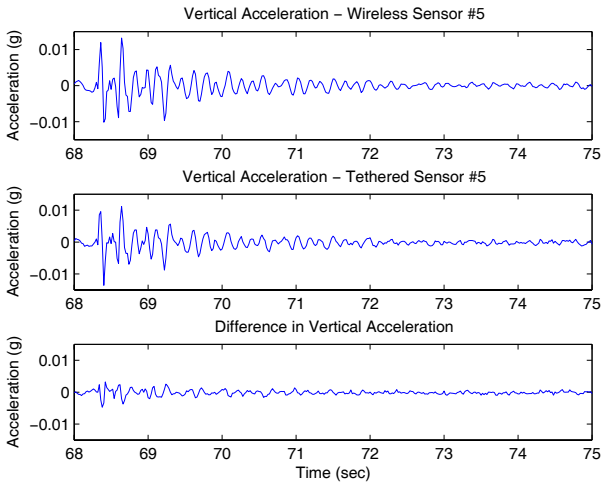


Figure 2.5: Comparison of tethered and wireless sensing systems.

2.3.2.1 Wireless System Performance

The first testing objective was to validate the accuracy of the proposed wireless sensing network against a traditional, tethered monitoring system. During all nine tests, two channels of tethered acceleration data were collected in parallel with the wireless network. Both monitoring systems employed a sample rate of 50 Hz. If the response of the tethered system is compared alongside that of the wireless system, it can be seen that the recorded time history data is nearly identical, as seen in Figure 2.5. In this figure, the response from both monitoring systems is plotted between 68 and 75 seconds. Very little discrepancy is observed if the two acceleration time histories are subtracted from one another. Similar results were obtained in other locations and in all testing scenarios.

2.3.2.2 Embedded Peak Picking Results

The second testing objective was to validate each of the distributed data processing algorithms (PP, FDD, RD) proposed in this study. In order to validate the ability of the PP method to extract modal frequencies from an output-only system, it is

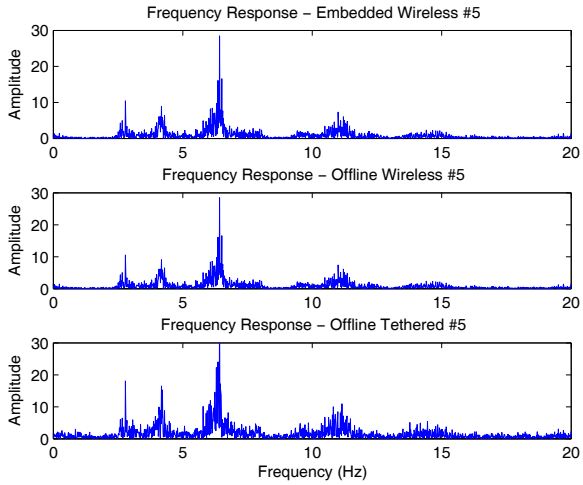


Figure 2.6: Fourier spectra for the balcony response at sensor location 5: (top) embedded FFT executed by wireless sensor; (middle) calculated offline using the wireless data; (bottom) calculated offline using the tethered data.

necessary to first prove the effectiveness of each of the numerical tools used within this identification technique. The first of these tools is the embedded FFT. In all of the testing runs in which PP analysis was requested, each node of the wireless sensor network was required to calculate a 4096-point complex-valued Fourier spectrum from the time history data collected. A Fourier spectrum from one sensing location is shown in Figure 2.6. For comparison, Fourier spectra calculated offline in MATLAB (MathWorks) using time history data from the tethered and wireless monitoring systems are also shown. It can be seen that the frequency characteristics extracted from the embedded algorithm are very similar to the results obtained using an offline analysis of either tethered or wireless time history data.

The second numerical tool in question is the embedded PP algorithm itself. This algorithm is required in both the PP and the FDD output-only identification methods presented in this study. In all of the test cases in which one of these two methods was used, each wireless sensor in the network was asked to extract the three highest peaks

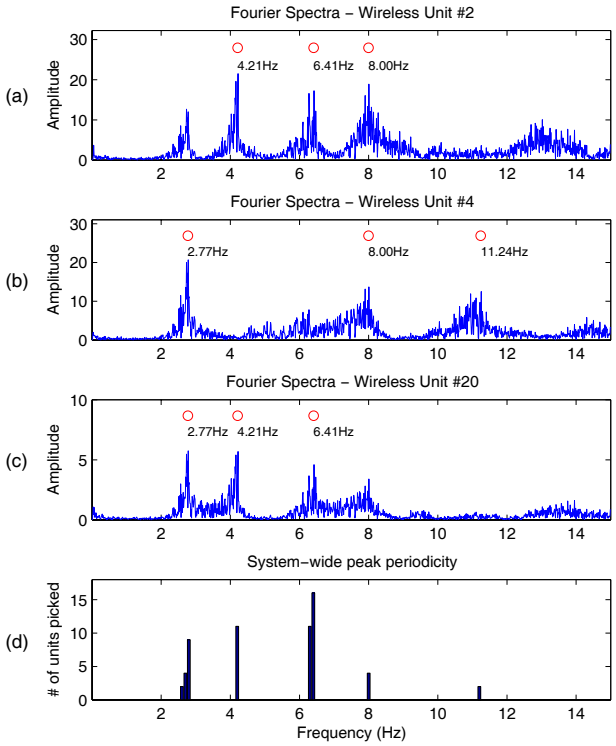


Figure 2.7: Embedded PP modal frequency results from: (a) sensor location 2, (b) sensor location 4, and (c) sensor location 20. (d) System-wide distribution of picked peaks tabulated at a central wireless sensor node.

from the Fourier spectrum created using the embedded FFT algorithm. Because peak-picking is a somewhat subjective science, no one sensing unit can be solely relied upon to correctly identify three distinct modal frequencies. As such, PP results from each sensing node must be transmitted to a designated node or central server where an intelligent decision can be made about final modal frequencies. PP results from three different units can be seen in Figure 2.7, which also shows the ability of a central server to determine system-wide modal frequencies from a complete set of PP data (compiled from all 21 nodes). It can be seen that by looking at the peak picking results as a whole, a reasonable global estimate of peak frequencies can be extracted from system-wide data. The central node was able to identify system-wide modal frequencies and extract mode shapes for the first (2.77Hz), second (4.14Hz), fourth (6.40Hz), and fifth (7.93Hz) modes. Note that the

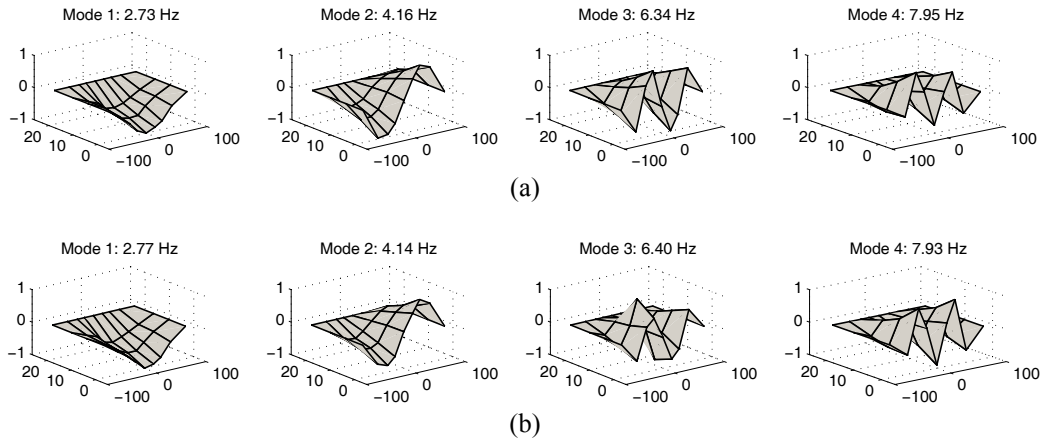


Figure 2.8: (a) Offline centralized FDD mode shape results and (b) embedded PP mode shape results based on in-network processing.

third mode (5.11Hz) is absent, as the chosen excitation point did not provide adequate spectral content at this modal frequency for proper peak picking mode detection. Figure 2.8 compares a set of mode shapes calculated using the embedded PP method with a set of mode shapes calculated offline using the centralized FDD technique. Numerical comparisons between modal frequencies and mode shapes calculated using the two methods are presented in Table 2.2. In this table, mode shapes determined with peak picking are compared with the offline FDD modes using the modal assurance criteria (MAC), as defined by Allemang and Brown (1982). Strong agreement is observed in the modal frequencies and mode shapes between those derived by the wireless sensor network and those found off-line using a centralized server running MATLAB. Modal frequencies are within 1% of one another while MAC values of 0.9 or greater are observed in most modes.

2.3.2.3 Embedded Frequency Domain Decomposition Results

The second embedded modal identification method presented in this paper is the FDD technique. This method was chosen for this study because of its advantages over

Table 2.2: Summary of modal identification results from embedded peak picking method.

Method	Run #	Natural Frequency (Hz)				MAC			
		Mode 1	Mode 2	Mode 4	Mode 5	Mode 1	Mode 2	Mode 4	Mode 5
Centralized FDD (off-line)	1	2.734	4.163	6.335	7.946	1.000	1.000	1.000	1.000
	2	2.727	4.210	6.349	7.996	-	0.949	0.937	0.779
Peak Picking (embedded)	3	2.734	4.135	6.342	8.020	0.825	0.678	0.427	0.817
	4	2.772	4.144	6.396	7.929	0.990	0.973	0.869	0.944

peak picking when estimating mode shapes from output response data. When implemented within a wireless sensing network, this method creates a large array of overlapping two-node mode shapes that can be easily assembled at a later time by a central processor (either a designated node or a server). This distributed technique provides a great degree of scalability by parallelizing a typically centralized algorithm to be executed by a community of wireless sensor nodes. As such, three distinct network topologies were designed and tested for the sharing of Fourier spectra and the creation of two-node mode shapes. As can be seen in Figure 2.9, data sharing in each of the three network topologies begins with the same root node (wireless sensor 1), but creates a very different set of two-node pairs. Each topology is meant to test different nodal overlaps so as to observe the sensitivity of the distributed FDD method to topology and to validate the scalability of this method. Because it has been shown that a very small synchronization error (with a maximum of 5ms) may occur between distantly spaced nodes, the assumption is that topologies with closely spaced two-node mode shapes should behave better than topologies with distantly spaced nodal connections.

Additionally, it is assumed that increased symmetry within a topology will lead to a decrease in mode shape accuracy, depending on the nodal locations of the detected modes. Figure 2.10 displays the extracted mode shapes using these three distinct network

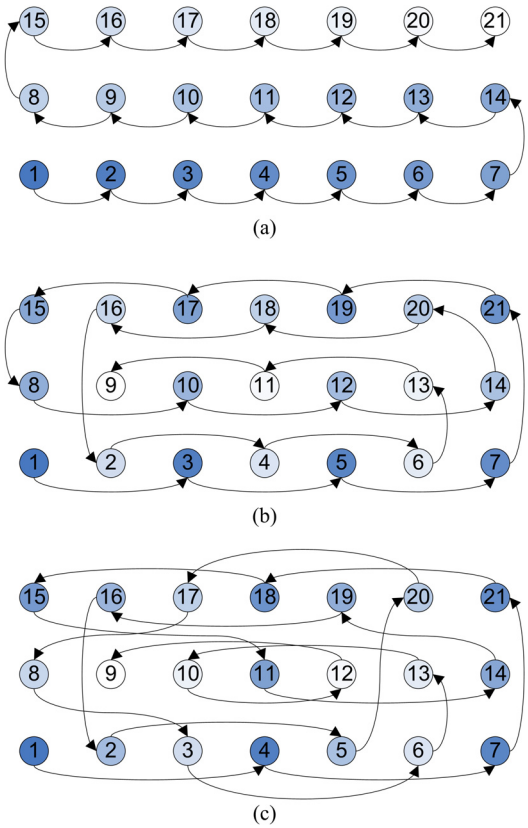


Figure 2.9: Network topologies for two-node FDD data sharing (arrows and shading indicate transmission of Fourier spectra for 2-point mode determination): (a) topology 1, (b) topology 2, and (c) topology 3.

topologies in addition to the mode shapes found using an offline centralized FDD method. Because of the loading location, the third mode (5.11Hz) was not captured in each of these cases. Table 2.3 provides a numerical comparison between mode shapes calculated using the embedded FDD method and those calculated offline (runs 5, 6, and 7 correspond to topologies 1, 2, and 3, respectively). Again, these mode shapes are compared using the MAC, and MAC values of 0.9 or greater are typically obtained for the network determined modes.

It can be seen from Figure 2.10 that the first topology provides excellent mode shape estimates for all four detected modes. However, the fourth mode in the second topology and the third mode in the third topology appear to be somewhat inconsistent

Table 2.3: Summary of modal identification results from embedded frequency domain decomposition method.

Method	Run #	MAC			
		Mode 1	Mode 2	Mode 4	Mode 5
Centralized FDD (off-line)	1	1.000	1.000	1.000	1.000
Decentralized FDD (embedded)	5	0.957	0.985	0.961	0.840
	6	0.988	0.943	0.821	0.373
	7	0.994	0.984	0.630	0.960

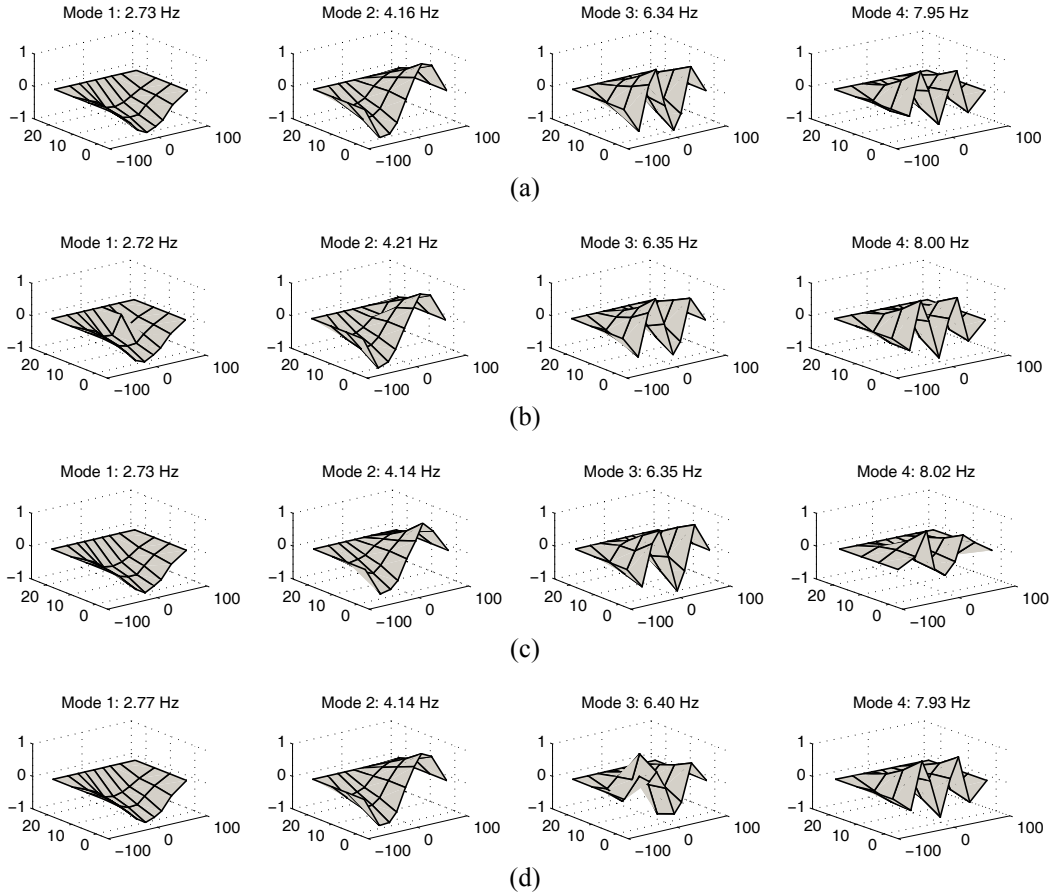


Figure 2.10: (a) Offline centralized FDD mode shape results and embedded FDD mode shape results for (b) topology 1, (c) topology 2, and (d) topology 3.

with the mode shapes calculated using a centralized FDD (MAC < 0.9). This is most likely due to the nodal locations of the third and fourth modes, as well as symmetry between two-node pairs in the second and third topologies. Because of the impact that topology can have on the accuracy of mode shapes extracted with this distributed FDD

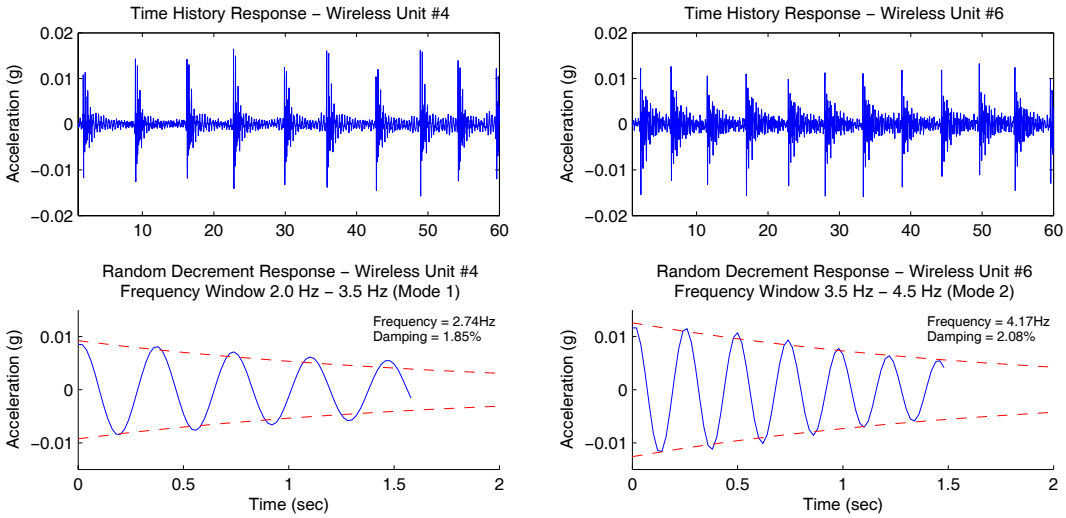


Figure 2.11: Embedded RD modal frequency and damping results.

technique, more work is required to fully understand the effects of topology choice on this method.

2.3.2.4 Embedded Random Decrement Results

The third distributed system identification technique implemented on the wireless sensor network in the study is the RD method. For each test in which the RD method was used, each sensor on the network collected a consistent set of time history data. Using the RD algorithm, this time history response was transformed at each node into a SDOF free decay response function using a user-defined trigger amplitude (which is defined as a certain percent of the standard deviation of the time history response) and frequency window (*e.g.,* 2.0 to 3.5 Hz for mode 1, *etc.*) meant to target a specific mode. Figure 2.11 shows an output response time history alongside a random decrement free decay response for each of the first two modes, calculated by wireless sensors 4 and 6, respectively. It can be seen that by employing zero crossing and logarithmic decrement techniques on the resulting free decay response functions, estimates of modal frequencies and damping

Table 2.4: Summary of modal identification results from embedded random decrement method.

Method	Run #	Natural Frequency (Hz)		Damping Ratio (%)	
		Mode 1	Mode 2	Mode 1	Mode 2
Centralized FDD (off-line)	1	2.734	4.163	2.321	1.610
Random Decrement (embedded)	8	2.740	-	1.792	-
	9	-	4.159	-	1.864

ratios can be determined at each sensing location. The type of quality result seen in Figure 2.11 was repeated in each testing instance and at almost all sensing locations. Once collected at the individual sensor, modal frequency and damping data can be shared with a centralized node or server and a global set of modal frequencies and damping ratios can be determined by throwing out outliers and averaging the remaining results. After averaging, the distributed RD method produced system-wide modal frequencies of (2.74 Hz and 4.16 Hz) and damping ratios of (1.79% and 1.86%) for the first two modes. These results are compared with offline results obtained using a centralized EFDD method and are displayed in Table 2.4.

2.4 Pedestrian Bridge Testbed

In order to further validate the ability of a network of wireless sensing units to autonomously estimate modal properties in a full-scale structure, the Bandemer Park pedestrian bridge in Ann Arbor, MI is chosen as an ideal testbed. This bridge, shown in Figure 2.12, consists of a wooden deck supported by a simple steel truss, and is approximately 30m (100ft) long and 2m (8ft) wide.

Figure 2.12: Bandemer Park pedestrian bridge.

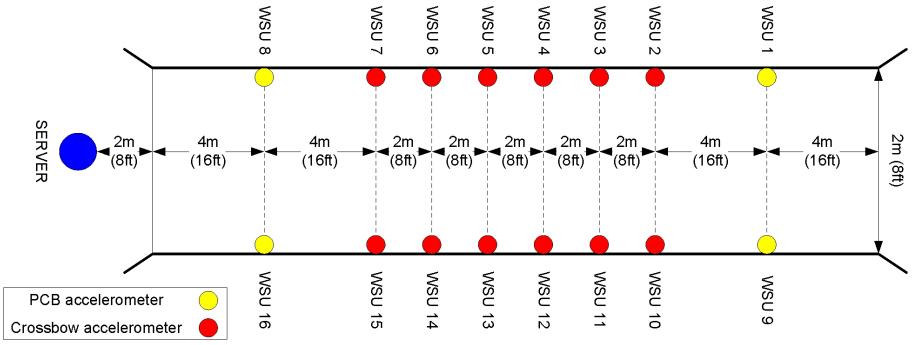


Figure 2.13: *Narada* sensor layout for Bandemer Park pedestrian bridge field test.

2.4.1 Instrumentation and Excitation Strategy

On November 27, 2007, a network of sixteen *Narada* wireless sensing prototypes (described in detail in Section 1.6.2) were programmed with the distributed modal analysis algorithms described in Section 2.2 and deployed on the Bandemer Park pedestrian bridge. As displayed in Figure 2.13, wireless sensors were placed at consistent intervals along both sides of the deck and connected to either a PCB Piezotronics 3801D1FB3G or Crossbow CXL02LF1Z MEMS capacitive accelerometer; both accelerometers were oriented to monitor the vertical acceleration of the bridge deck. As in the theatre balcony tests, the signal conditioning circuit proposed by Lynch*, et al.* (2006) was included with each sensor to both amplify and band-pass (0.02 – 25 Hz) acceleration response data before inputting to the wireless sensor's ADC.

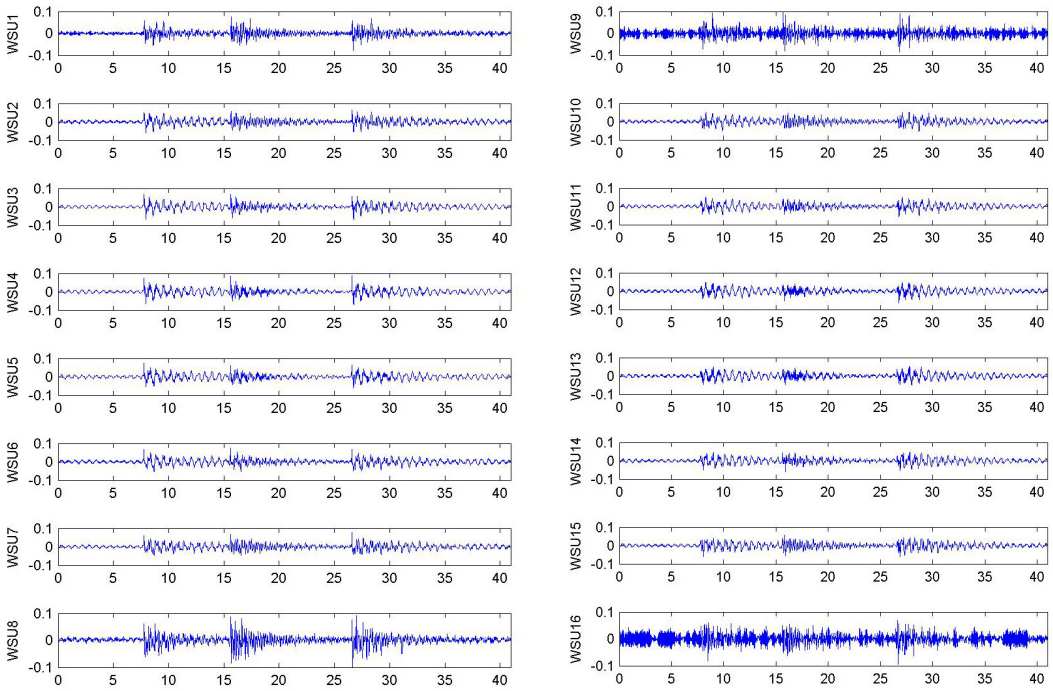


Figure 2.14: Acceleration (g) vs. time (s) response of the Bandemer bridge, as collected by *Narada* sensors.

2.4.2 Experimental Results

On the day of testing, several vibration tests were run using impulse loadings generated by a single-person (weighing 82kg) performing a heeldrop. These heel drops, which are performed by quickly raising and dropping both heels simultaneously, were executed in various locations a short distance from the center of the bridge span, in an attempt to avoid exciting directly at a modal node. Because of the impulse nature of this type of loading, it can be assumed that each heel drop test applies a broadband input to the structure. An example set of acceleration time history plots collected by the wireless sensors for one of these heel drop tests can be found in Figure 2.14.

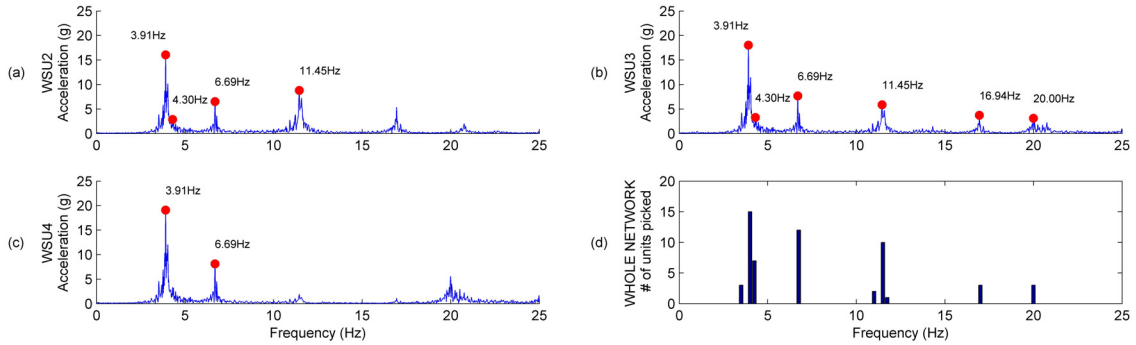


Figure 2.15: (a,b,c) Example FFT and PP results, as collected by three *Narada* sensors and (d) network-wide distribution of picked peaks for one test, collected at a central node.

2.4.2.1 Embedded Peak Picking Results

In all of the test cases where a fast Fourier transform (FFT) was requested, each wireless sensor in the network was also asked to extract up to ten peaks from its individual Fourier spectrum. Figures 2.15(a), 2.15(b), and 2.15(c) show the Fourier spectrum calculated online in three of the *Narada* wireless sensors, as well as the modal peaks that each individual sensor picked within its frequency response. Figure 2.15(d) shows a periodogram of the network-wide PP results. It can be seen from this figure that while each individual sensor may not have individually determined all pertinent modal frequencies, the network as a whole did a decent job of determining all five distinct modal frequencies (3.9, 6.8, 11.6, 16.9, 20.0 Hz).

Although simplistic, the PP method is not only useful for identifying modal frequencies, but also for estimating mode shapes. The mode shapes calculated using the distributed PP method are compared with similar mode shapes calculated offline using a centralized FDD method on the wirelessly collected raw data. Both of these sets of mode shapes are plotted alongside one another in Figure 2.16, and are presented numerically in Table 2.5. All comparisons between mode shapes are formulated using the modal assurance criteria (MAC) as defined by Allemang and Brown (1982). The centralized

Table 2.5: Summary of modal identification results from autonomous embedded methods.

Method	Natural Frequency (Hz)					Damping Ratio			MAC				
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 1	Mode 2	Mode 3	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Centralized FDD (off-line)	3.91	6.79	11.57	16.94	20.04	--	--	--	1.000	1.000	1.000	1.000	1.000
Peak Picking (embedded)	3.91	6.69	11.41	16.94	20.00	--	--	--	0.141	0.969	0.886	0.944	0.589
Decentralized FDD (embedded)	--	--	--	--	--	--	--	--	0.994	0.990	0.953	0.689	0.499
Random Decrement (embedded)	3.89 ± 0.00	6.72 ± 0.04	11.48 ± 0.08	--	--	0.83% ± 0.12%	1.38% ± 0.70%	0.16% ± 0.04%	--	--	--	--	--

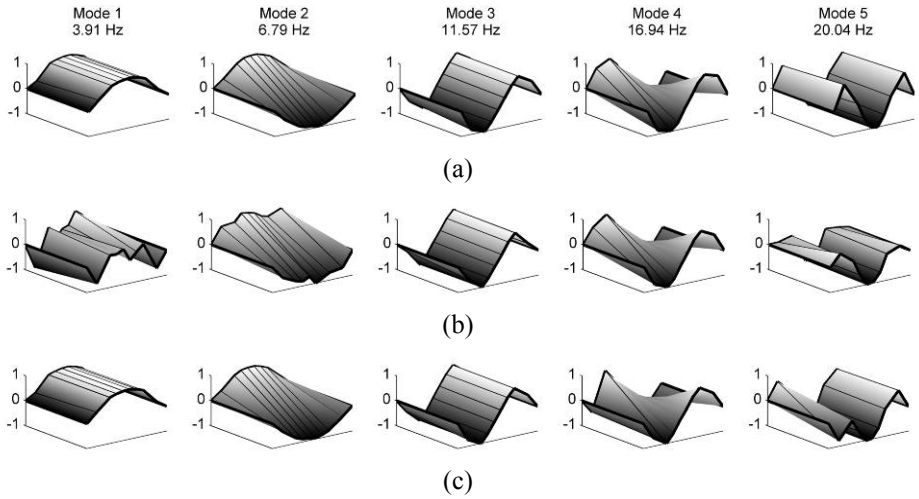


Figure 2.16: (a) Offline FDD modes, (b) embedded PP modes, and (c) embedded FDD modes.

FDD results are used as a baseline. It can be seen that the PP method performs acceptably for the higher three modes, but does not produce very accurate results for lower frequency mode shapes.

2.4.2.2 Embedded Frequency Domain Decomposition Results

The FDD technique has several significant advantages over the PP method when estimating mode shapes from output-only response data; it not only provides more reliable and robust mode shape estimates, but it can be effectively used in systems with closely spaced modes. The mode shapes determined using the distributed FDD algorithm

embedded within the network of *Narada* wireless sensors can be seen in Figure 2.16, alongside those calculated offline using the centralized FDD method and those calculated in-network using the PP technique. Table 2.5 provides a numerical comparison between these three methods using MAC values. It can be seen that this technique performs much better that the PP method for the low frequency modes, but is not as accurate as the centralized FDD algorithm at estimating the higher frequency mode shapes.

2.4.2.3 Embedded Random Decrement Results

The last embedded analysis technique used in this study is the RD method. When running RD tests in the field, a consistent set of acceleration data is first collected at each node. Using different frequency windows provided by the user (each calculated using PP results), each mode can be singled out for RD analysis. For example, a window of 3.0-5.0 Hz was used to isolate mode 1 (at 3.9 Hz). At this point, each node in the wireless sensing network performs RD calculations and returns the resulting random decrement response along with an estimated frequency and damping ratio. In this way, 16 independent estimates of these two modal properties are created. A graphical example of several random decrement responses for different modes can be seen in Figure 2.17, and numerical estimates of modal frequencies and damping ratios found with this method can be found in Table 2.5. Note that because the fourth and fifth modes (17 and 20 Hz) are significantly high relative to the sampling rate (100 Hz), the random decrement response for these modes did not produce meaningful damping ratios or modal frequency estimates.

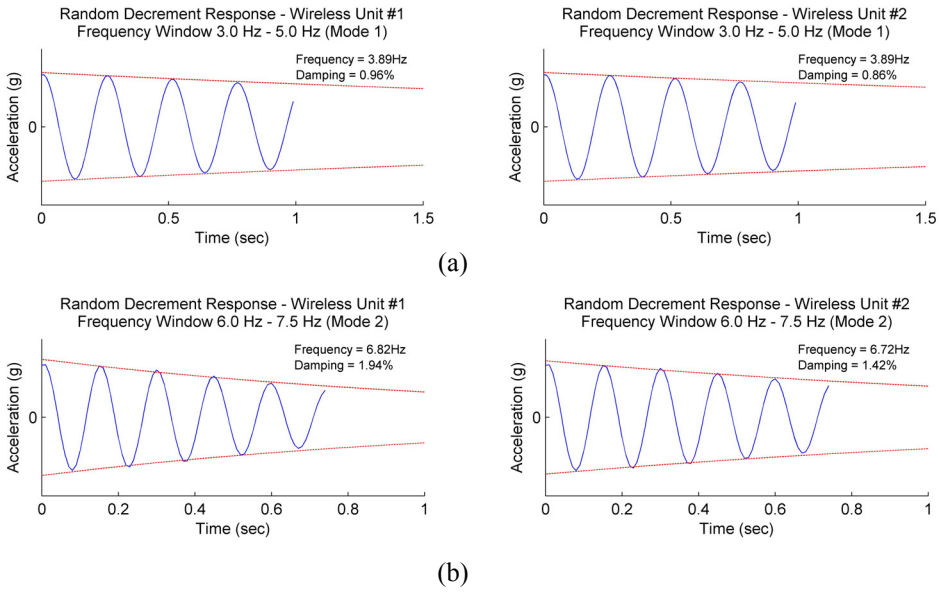


Figure 2.17: Example random decrement frequency and damping ratio results for (a) mode 1 and (b) mode 2.

2.5 Chapter Summary

Structural monitoring systems have become increasing popular for monitoring the response characteristics of large civil structures subjected to ambient and forced vibrations. By leveraging wireless communication technology, wireless monitoring systems can be installed at a fraction of the cost and in much higher sensor densities than traditional tethered sensing systems. In addition to these cost savings, however, wireless sensors have an enormous advantage over their tethered counterparts because of their local analog-to-digital conversion and data processing capabilities. By taking advantage of the embedded computing resources distributed across a large network of wireless sensors, decentralized wireless monitoring systems can perform as well as centralized tethered systems, with the added advantage of being able to process sensor data locally.

Using the embedded output-only modal parameter estimation methods proposed in this chapter, a wireless monitoring system was shown to be capable of collecting and processing measured data at the individual sensor. The wireless network was able to

autonomously determine modal frequencies using a distributed peak picking (PP) algorithm, mode shapes using a distributed frequency domain decomposition (FDD) method, and modal damping ratios using a distributed random decrement (RD) technique. It can be seen that the embedded techniques yield modal parameters comparable to those obtained using traditional offline analyses.

This chapter, modified from (Zimmerman*, et al.* 2008a), presents a successful implementation of distributed modal parameter estimation techniques within the computational core of a network of wireless sensing prototypes. By moving away from the traditional centralized approach to modal estimation, the work presented herein represents a first step towards an autonomous and decentralized environment for computing within a wireless structural monitoring system. By providing an architecture for parallel data processing in WSNs, system scalability and spatial resolution problems typically associated with centralized architectures can be greatly minimized. However, the architecture developed in this chapter does not represent a truly agent-based approach to data processing in WSNs, and as such retains some of the reliability issues associated with centralized architectures. In particular, the need for a defined network topology creates a limitation on the flexibility of the proposed solution in the wake of communication or sensor failure.

That said, the work presented herein can be used to motivate additional, increasingly agent-based techniques for distributed data processing in WSNs. Building directly on the success of this implementation, Chapter 3 presents an ad-hoc, agent-based approach to model updating in WSNs, and Chapter 5 outlines an improved agent-based approach to mode shape estimation using the FDD technique.

CHAPTER 3

A PARALLEL SIMULATED ANNEALING ARCHITECTURE FOR MODEL UPDATING WITHIN A WIRELESS SENSOR NETWORK

3.1 Introduction

In this chapter, which is modified from (Zimmerman and Lynch 2009), a novel data processing architecture that builds on the work in Chapter 2 is created for use within an agent-based WSN. By viewing a wireless network as a parallel computer with an unknown and possibly changing number of processing nodes, this 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. As such, it moves us further away from the scalability issues, spatial disadvantages, and reliability problems associated with traditional centralized data processing architectures, and towards a truly autonomous and pervasive agent-based architecture for data processing in wireless monitoring systems. 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. By moving away from the rigid network topologies that constrain computation in Chapter 2, the work presented in this chapter begins to fully leverage the benefits of ad-hoc, agent-based computation in WSNs.

In order to examine the data processing capabilities of this novel architecture from an SHM 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 arise in almost all engineering disciplines. In this chapter, a wireless parallel SA (WPSA) method is developed for use within a WSN for the updating of structural models. This type of model updating can be used for many purposes in SHM, including analytical model validation, design iteration and improvement, and the detection of damage or degradation within a structure. In order to validate this WSN-based model updating approach, acceleration data collected from a three-story steel structure is used to update an analytical model of the structure using a network of wireless sensing prototypes. It can be seen that the WPSA algorithm, when applied in-network to a model updating application, can be used to accurately determine the mass, stiffness, and damping properties of a physical structure. It is also experimentally determined that the algorithm exhibits significant performance gains as the size of the wireless network is increased.

3.2 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:

$$\Pr(accept) = e^{\frac{-\Delta E}{k_B \cdot T}} \tag{3.1}$$

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 3.1. 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:

$$E(s_{new}) \leq E(s_{old}) + T \cdot |\ln(U)| \tag{3.2}$$

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 3.2). 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.

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 chapter, 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 3.1a). 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

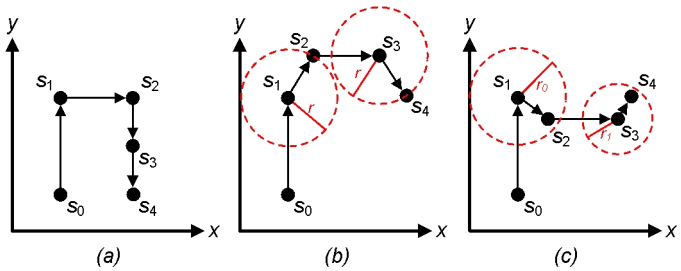


Figure 3.1: Random state generation for a two-dimensional search problem using (a) standard SA, (b) BSA, and (c) WPSA.

annealing state (Figure 3.1b). 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 3.1c). 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.

3.3 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,

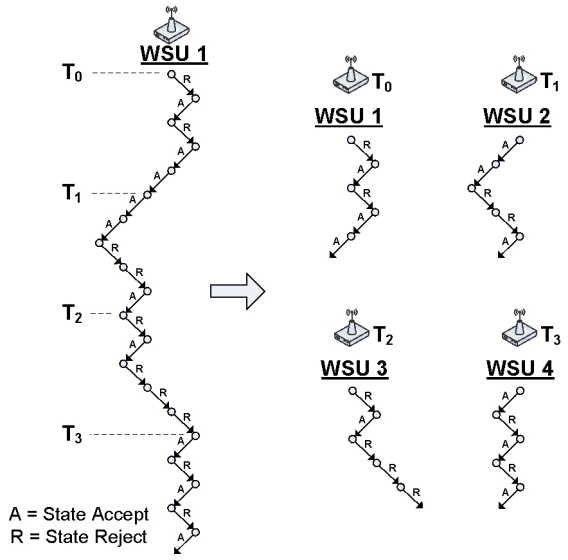


Figure 3.2: A simple serial SA search tree, shown up to the fourth temperature step and its corresponding WPSA search trees, assigned to wireless sensors.

and thus multiple temperature steps can be searched concurrently. This concept is displayed graphically in Figure 3.2. 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, 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.

3.3.1 Wireless Implementation of the WPSA Algorithm

In the wireless 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.

3.3.2 Illustrative Example of the WPSA Algorithm

Figure 3.3 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,

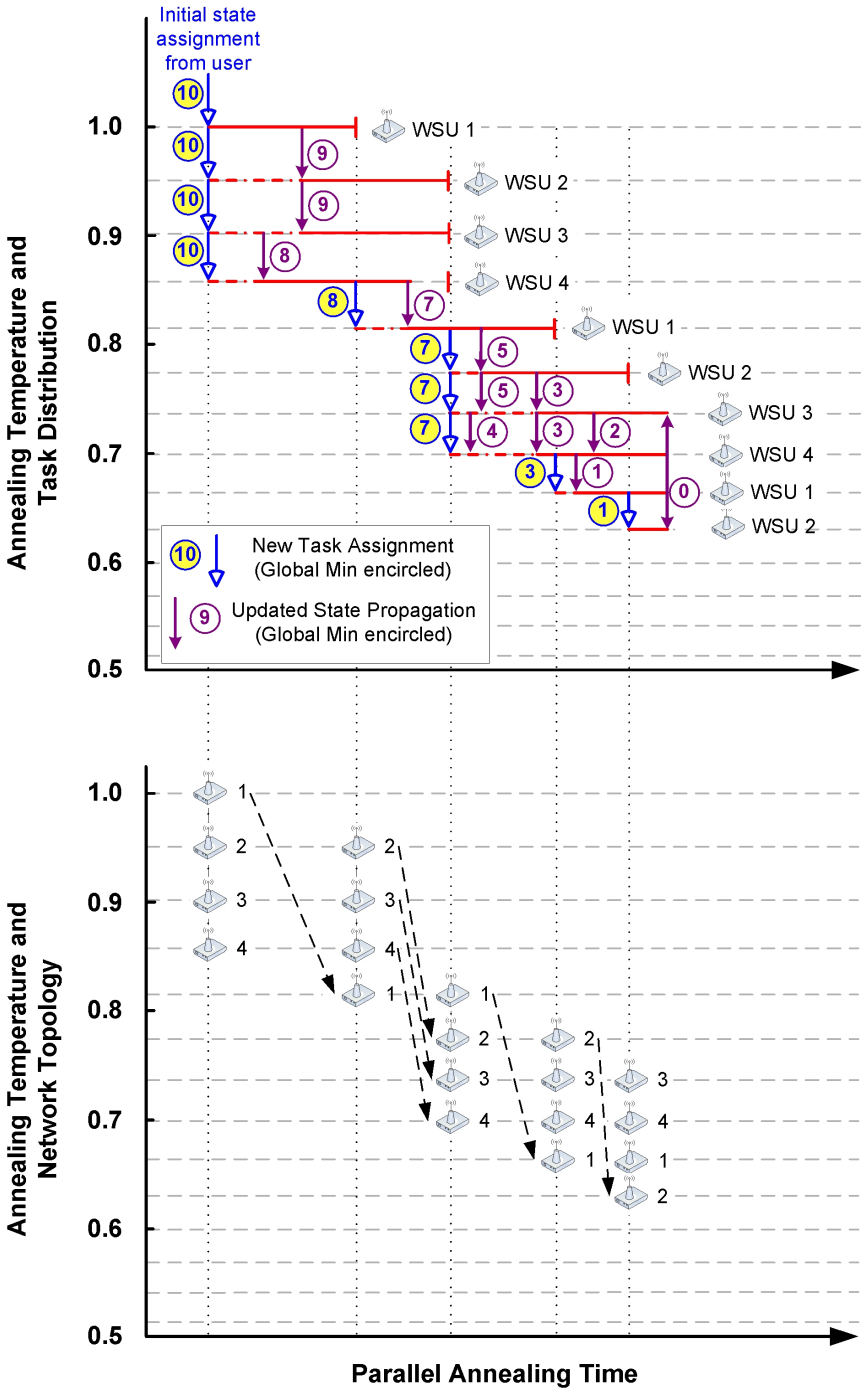


Figure 3.3: One wireless parallel simulated annealing task running on four wireless sensors.

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.

3.4 Overview of Model Updating in Structural Health Monitoring

In SHM, CO techniques have been successfully used for many purposes, including determining optimal sensor placement (Rao and Anandakumar 2007), establishing decision boundaries for damage identification (Park and Sohn 2006), and updating model parameters to fit experimental sensor data (Levin and Lieven 1998). In this study, SA is investigated as a possible optimization tool to be used for SHM applications involving the updating of dynamic structural models. These dynamic model updating methods function by iteratively adjusting structural parameters in an analytical (*e.g.*, finite element) model such that the analytical system produces modal properties similar to those obtained experimentally in the physical structure (Mottershead and Friswell 1993). These results can then be used to track structural performance over time or to look for signs of long-term structural degradation (Doebling*, et al.* 1998). This approach has been used to effectively detect and locate damage in a variety of real structures (Teughels and DeRoeck 2004; Wu and Li 2006).

The most common vibration-based model updating methodology centers around the minimization of an objective function, E, which mathematically expresses the numerical difference between modal properties (*i.e.*, mode shapes, modal frequencies, and modal damping ratios) generated by a given analytical model and those obtained experimentally through vibration testing. For example:

$$E = \alpha \cdot \sum_{i=1}^{q} \left(\frac{\omega_{ai} - \omega_{ei}}{\omega_{ei}} \right)^2 + \beta \cdot \sum_{i=1}^{q} \frac{\left(1 - \sqrt{MAC_i}\right)^2}{MAC_i} + \gamma \cdot \sum_{i=1}^{q} \left(\frac{\zeta_{ai} - \zeta_{ei}}{\zeta_{ei}} \right)^2 + f(t) \tag{3.3}$$

where ω_{ai} and ω_{ei} are the i^{th} analytical and experimental modal frequencies, respectively, and ζ_{ai} and ζ_{ei} are the i^{th} analytical and experimental modal damping ratios, respectively. Also, MAC_i is the modal assurance criteria, which is a measure of correlation between two modes using the i^{th} analytical and experimental mode shapes as input (Allemang and Brown 1982). Finally, α, β, and γ are weighting constants determined experimentally to properly account for differences in the magnitudes of the three modal parameters. It is assumed that there are q modes in the system. A optional function $f(t)$ can also be added to the objective function in order to match experimental and analytical data characterized in the time domain.

Once formulated, the objective function in Equation 3.3 can be optimized by altering the values of a set of structural "updating" parameters. In most cases, these parameters consist of unknown and/or possibly transient mass, stiffness, and damping properties of individual structural components. In theory, if this objective function can be minimized, then there is an accurate match between experimental results and analytical prediction. This means that a set of structural parameters representing the true physical state of the system can be optimally determined.

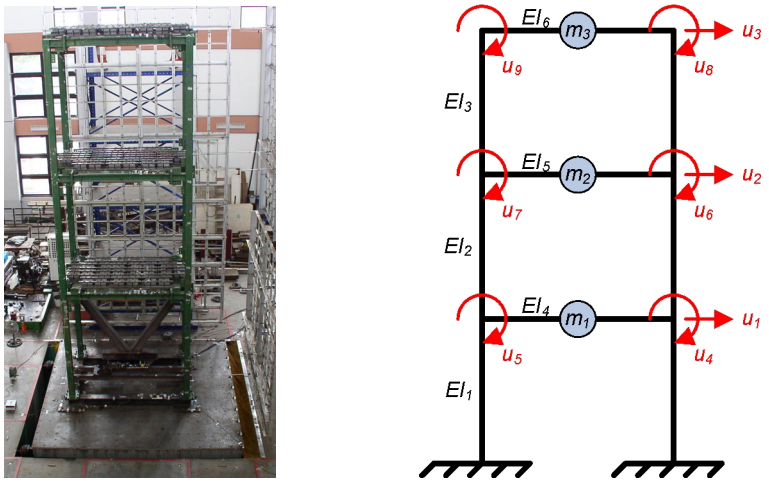


Figure 3.4: Three-story (a) structure and (b) model used in this study.

3.5 3-Story Steel Structure Testbed

In order to validate the ability of a WSN to update an analytical model of a physical structure, a three-story steel structure located at the National Center for Research in Earthquake Engineering (NCREE) at National Taiwan University in Taipei, Taiwan is chosen as a testbed. This structure sits on a 5m x 5m tri-axial seismic shaking table. Each floor of this structure (seen in Figure 3.4a) is 3m (9.84ft) wide by 2m (6.56ft) deep and 3m (9.84ft) tall. Seismic ground motion is applied parallel to the longer floor dimension. A mass of approximately 6000kg (13,228lb) is supported by each floor. Four H150x150x7x10 steel sections are employed as columns with the weak axis aligned with the direction of lateral motion. The orientation of the columns result in a theoretical stiffness of approximately 2000kN/m (11420lb/in) at each floor. Each floor is instrumented with a wireless sensor measuring acceleration using a Crossbow CX02LF1Z accelerometer oriented in the direction of lateral excitation. An accelerometer-wireless sensor pair (using the *Narada* wireless sensor described in

Section 1.6.2) is also placed at ground level to measure the ground acceleration driving the system.

3.5.1 Analytical Model of the 3-Story Steel Structure

Because of the computational constraints (processing speed and memory) imposed by the wireless sensing hardware used in this study, it is decided to utilize a damped frame model with lumped masses (Figure 3.4b) to analytically describe the dynamic response of the steel structure. This model is characterized by the following equation of motion:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = -\mathbf{M}\mathbf{1}\ddot{u}_g(t) \tag{3.4}$$

where $\mathbf{u} \in \Re^{9x1}$ is a vector of displacements (relative to the ground) and rotations for each of the 3 degrees of freedom, $\mathbf{M} \in \Re^{9x9}$, $\mathbf{C} \in \Re^{9x9}$, and $\mathbf{K} \in \Re^{9x9}$ are the structure's mass, damping, and stiffness matrices, respectively, $\mathbf{1} \in \Re^{9x1}$ is a unity vector, and $\ddot{u}_g(t)$ is the lateral ground acceleration. Since mass is not associated with the rotational degrees of freedom, static condensation is used to create reduced order $\mathbf{M_R} \in \Re^{3x3}$ and $\mathbf{K_R} \in \Re^{3x3}$ matrices. A Rayleigh damping matrix, $\mathbf{C_R} \in \Re^{3x3}$, is constructed given the modal damping ratios in two lateral modes (Caughey and O'Kelley 1965). Given these matrices, the simplest way to obtain analytically derived modal properties is by using the state space formulation of the equation of motion. This formulation is as follows:

$$\dot{\mathbf{z}}(t) = \mathbf{A}\,\mathbf{z}(t) + \mathbf{B}\,\ddot{u}_g(t) \tag{3.5}$$

$$\mathbf{y}(t) = \mathbf{C}\,\mathbf{z}(t) + \mathbf{D}\,\ddot{u}_g(t) \tag{3.6}$$

where $\mathbf{z} \in \Re^{6x1}$ is the system state vector ($\mathbf{z}^T = \{\mathbf{u}^T \ \dot{\mathbf{u}}^T\}$), $\mathbf{y} \in \Re^{3x1}$ is the output vector corresponding to a measurement of acceleration relative to the base motion at each lateral degree of freedom, and $\ddot{u}_g$(t) is the time-dependent ground acceleration input to the system. The state space matrices **A**, **B**, **C**, and **D** can be expressed as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ -\mathbf{M_R}^{-1}\mathbf{K_R} & -\mathbf{M_R}^{-1}\mathbf{C_R} \end{bmatrix} \in \Re^{6x6} \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \in \Re^{6x1}$$
$$\mathbf{C} = \begin{bmatrix} -\mathbf{M_R}^{-1}\mathbf{K_R} & -\mathbf{M_R}^{-1}\mathbf{C_R} \end{bmatrix} \in \Re^{3x6} \quad \mathbf{D} = [\mathbf{1}] \in \Re^{3x1} \tag{3.7}$$

Modal frequencies (ω_i), mode shapes (ϕ_i), and modal damping ratios (ζ_i) can be easily extracted from the state space formulation by finding the eigenvalues (λ_i) and eigenvectors (ψ_i) of the system matrix, **A**:

$$\omega_i = |\lambda_i|, \quad \phi_i = \psi_i, \quad \zeta_i = \cos[\tan^{-1}(\mathrm{Im}(\lambda_i)/\mathrm{Re}(\lambda_i))] \tag{3.8}$$

where ω_i is expressed in radians per second.

3.5.2 Model Updating of the 3-Story Steel Structure

In this study, three unknown mass (m_i) values, six unknown stiffness (EI_j) values corresponding to the columns and beams of each floor, and two unknown modal damping (ζ_k) ratios are treated as updating parameters (see Figure 3.4b). The WPSA algorithm is

used to stochastically search the range of each of these eleven model parameters for a system state ($s = \{m_1 \ldots m_3, EI_1 \ldots EI_6, \zeta_1 \ldots \zeta_2\}$) which provides an optimal match between modal properties produced by experimental response data and those extracted from the analytical model, defined as a parameter assignment which minimizes the objective function presented in Equation 3.3. In this study, the optional function parameter $f(t)$ is not used. Note that if these model parameters are updated repeatedly over time, significant changes could indicate structural degradation.

The first step in the model updating procedure is to determine baseline experimental values for the modal properties (ω_i, ϕ_i, ζ_i) used in the objective function. For this study, the objective function weighting parameter α is taken to be 0.6, β is taken to be 0.1, and γ is taken to be 0.005. By employing peak picking, frequency domain decomposition, and random decrement methods already embedded in the wireless sensors (Zimmerman*, et al.* 2008a), modal frequency (ω_i), damping ratio (ζ_i), and mode shape (ϕ_i) values can be automatically extracted from acceleration data collected at each floor. Once experimental modal parameters have been determined, the user (or a predetermined "controller unit") will commence the model updating procedure by selecting an available sensor at random and assigning to it the model updating problem, as well as a predetermined initial annealing temperature step, T_0. This initial sensor then searches for additional available units and the WPSA process begins. When the WSN has converged on an optimal system state, results are communicated back to a centralized server for viewing by the user.

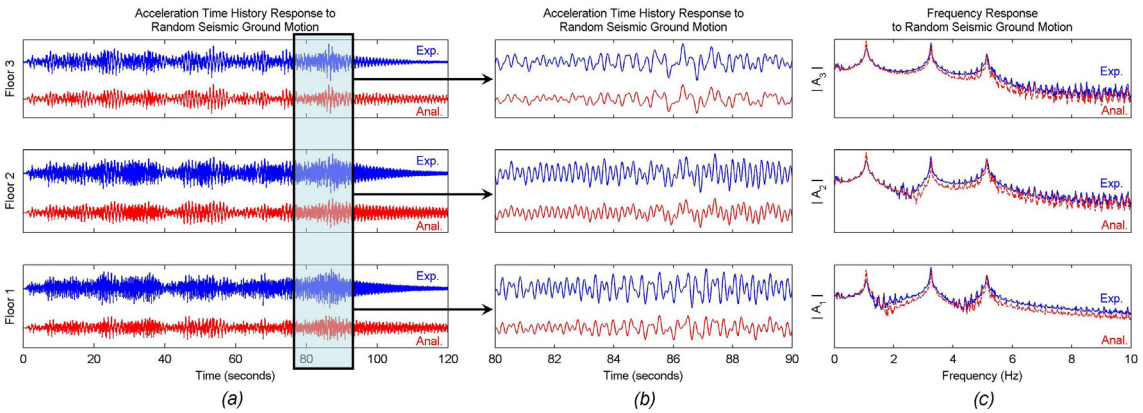


Figure 3.5: Comparison of experimental and updated analytical model response to seismic base motion in the (a),(b) time and (c) frequency domains.

3.5.3 Model Updating Results

When evaluating how well the proposed WPSA algorithm performs in a model updating application, it is first necessary to examine how successfully it can be used to produce model outputs that match experimental data in both the time and frequency domains. Since the objective function in Equation 3.3 focuses solely on matching frequency domain information, it is decided to utilize the optional term, $f(t)$, in order to improve the quality of the time history match. This is accomplished by writing a Newmark numerical integration scheme and embedding it within each wireless sensor. Then, $f(t)$ can be calculated by finding the average sum-squared difference between the analytically projected and experimentally sensed time histories. As seen numerically in Table 3.1 and graphically in Figure 3.5c, the WPSA algorithm is capable of producing a good match between analytical and experimental frequency domain properties in the three-story structure. Additionally, Figure 3.5a and Figure 3.5b show the similarity between experimental and analytical acceleration time history responses to the same random ground excitation.

Table 3.1: Comparison of experimentally sensed and analytically derived modal properties for the three-story test structure.

	Floor Mass (kg)			Column Stiffness (kN/m)			Floor Stiffness (MN/m)			Modal Frequency (Hz)			MAC Value			Damping Ratio (%)		
Data Source	m_1	m_2	m_3	k_1	k_2	k_3	k_4	k_5	k_6	ϕ_1	ϕ_2	ϕ_3	ϕ_1	ϕ_2	ϕ_3	ϕ_1	ϕ_2	ϕ_3
Experimental	6000	6000	6000	2000	2000	2000	n/a	n/a	n/a	1.07	3.25	5.13	0.99	0.98	0.99	0.78	0.05	0.11
Updated Model	5937	6092	6337	1113	2194	1869	400	411	375	1.07	3.26	5.17				0.20	0.07	0.05

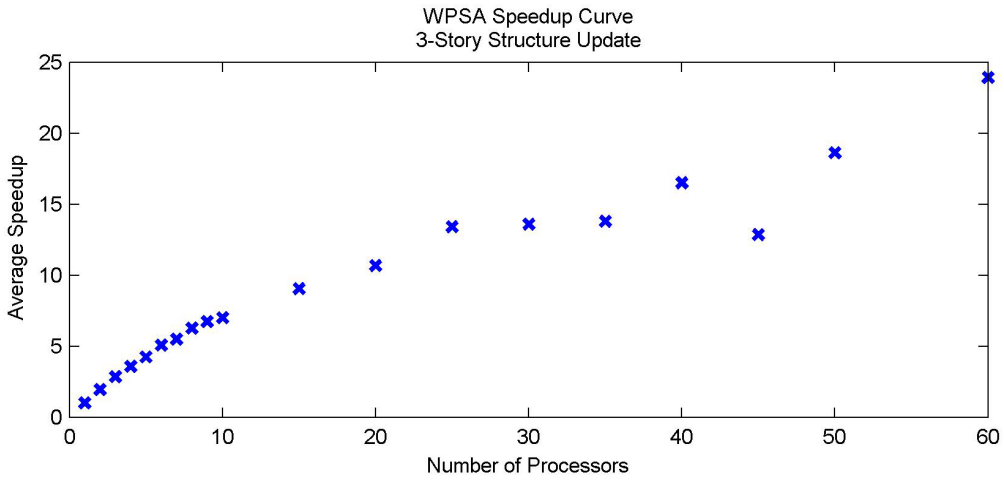


Figure 3.6: Experimental speedup curve structural model using WPSA.

To evaluate the feasibility of using WPSA to solve large model updating problems in wireless sensor networks, it is important to look at the scalability of the algorithm to large numbers of sensors. As such, Figure 3.6 displays the average speedup that is achieved by utilizing a given number of wireless sensors in a model update. Speedup is defined as the amount of time required to reach an optimal solution in the serial (one processor) case divided by the amount of time required to reach an optimal solution in the parallel (*n* processor) case. It can be seen that by increasing the size of a given wireless network from 1 to 60 units, a speedup of over 20 can be achieved. To put this in perspective, the amount of time required to completely update the three-story model presented in this study is decreased from 8.5 hours to 23 minutes. While this is significantly longer than the amount of time a PC would require to perform the same task, 23 minutes of computation is reasonable in the context of many SHM applications, where

an update may only need to be run at most once a day in order to evaluate long-term performance.

It is important to note that the WPSA algorithm can be utilized to update any analytical model that can be analyzed in a reasonable amount of time on a low-power microprocessor. For reference, in order to calculate the value of the objective function posed in Equation 3.3, the Atmel Atmega128 used in this study took 350ms plus 500ms for every 100 time history points projected using the Newmark numerical integration scheme. The algorithm consumed 35% of the 128kB of available internal flash ROM, 57% of the 4kB of available internal SRAM, and 2.33% of the 128kB of external SRAM for each 100 time history points projected using Newmark numerical integration. As such, there is room available on the Atmega128 for more complex analytical models, but any future work in this area requiring sophisticated analytics (*i.e.*, complex finite element models) may benefit from a wireless sensor with improved computational resources.

3.6 Chapter Summary

This chapter, which is modified from (Zimmerman and Lynch 2009), builds upon previous work in the parallel processing of data on wireless sensor networks by presenting a wireless parallel simulated annealing (WPSA) algorithm designed specifically to efficiently utilize the distributed resources available in large networks of wireless devices. This algorithm utilizes parallel computing concepts to gain efficiency as the number of sensors in a network grows, making it scalable to very large networks. Furthermore, it is robust to sensor or communication failure, and can be applied to many of the large number of combinatorial optimization problems seen across all engineering

disciplines. In this chapter, the proposed algorithm is embedded within a network of wireless sensing prototypes and utilized to update an analytical model of a three-story steel structure subjected to seismic base motion. It is shown experimentally that the WPSA algorithm is not only capable of accurately updating an analytical model, but that it can increase in computational efficiency as the size of the computing network grows.

When combined with the automated modal estimation techniques outlined in Chapter 2, the WPSA algorithm presented in this chapter represents a powerful tool that can be used in many SHM applications, including the validation of structural design assumptions, the improvement of analytical models, and the detection of damage within structural systems. The WPSA algorithm also represents the first truly agent-based approach to distributed data processing in wireless sensor networks. Like the modal estimation techniques of Chapter 2, the WPSA approach overcomes the bandwidth and scalability issues of traditional centralized architectures by processing data in parallel. But the WPSA computing framework also leverages the ad-hoc communications capabilities of a WSN to create an agent-based architecture that can continue to effectively process sensor data even in the wake communication or sensor failure. With this agent-based architecture in place, the work in this chapter lays the groundwork for the market-based resource allocation techniques presented in Chapter 4 of this dissertation.

CHAPTER 4

MARKET-BASED RESOURCE ALLOCATION FOR DISTRIBUTED DATA PROCESSING IN WIRELESS SENSOR NETWORKS

4.1 Introduction

Chapters 1 through 3 of this dissertation outline various application-specific architectures that the SHM community has been investigating for processing sensor data within networks of wireless sensors. Chapter 1 discusses centralized architectures for performing traditional engineering analyses like Fast Fourier Transforms (Lynch 2002), autoregressive model fitting (Lynch*, et al.* 2004), and wavelet transforms (Hashimoto*, et al.* 2005), as well as tiered network architectures (Chintalapudi*, et al.* 2006), data aggregation techniques (Gao 2005; Nagayama*, et al.* 2006; Akkaya*, et al.* 2008), and query processing (Rosemark and Lee 2005). Chapter 2 presents a set of explicitly parallel modal estimation methodologies are developed and validated for use within agent-based WSNs (Zimmerman*, et al.* 2008a). Chapter 3 develops a novel adaptation of the simulated annealing algorithm is created for updating structural models using ad-hoc networks of agent-based wireless sensors (Zimmerman and Lynch 2009).

Although the development of these parallel algorithms represents a significant step towards the automation of complex data processing tasks within agent-based WSNs, one of the key challenges yet to be overcome is that within the wireless environment many 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, especially in networks where multiple computational tasks may need to be executed simultaneously, it is important to devise an autonomous, optimal method of distributing and consuming these scarce system resources throughout the network. Due to the ad-hoc nature of many wireless networks, any method used for resource allocation must be able to achieve an optimal (or near-optimal) allocation even in the midst of changes in the network (for example, changes in node availability).

In this chapter, a resource distribution framework based on free-market economics is developed and used to autonomously allocate system resources for the simultaneous processing of multiple computational tasks within a WSN. Free-market economies can be thought of as large collections of autonomous market agents (participants) such as producers (sellers) and consumers (buyers), where each agent is forced to compete against other agents in a competitive marketplace with scarce resources. In such a system, each market agent decides for itself which actions to take based on the utility that a particular action generates. Utility, in this case, is defined as the degree to which the benefits associated with a given action outweigh the opportunity cost of that action. As such, market-based techniques are a logical choice for applications within autonomous sensor networks, where each sensor can act as an independent agent. These methods provide increased efficiency, reliability, and flexibility relative to an *a*

priori resource assignment mechanism, where network resources are explicitly assigned to various computational objectives before computation begins.

While the market-based concepts proposed herein can be merged with any number of parallel data processing frameworks performing a wide range of data analyses, it is decided to adopt the wireless parallel simulated annealing (WPSA) framework developed in Chapter 3 (Zimmerman and Lynch 2009) for solving combinatorial optimization problems as a validation testbed. In order to provide the system with multiple computational objectives, the classical *n*-Queens combinatorial optimization problem is chosen as a simple optimization task that can be easily scaled to varying complexities and solved using the WPSA framework.

The rest of this chapter is organized as follows: Section 4.2 presents a brief overview of work related to market-based resource allocation in wireless sensor networks and Section 4.3 provides background on both the *n*-Queens validation testbed and the WPSA algorithm. Section 4.4 presents the proposed market-based resource allocation algorithm and Section 4.5 discusses the performance of the proposed algorithm when it is applied to the *n*-Queens/WPSA experimental testbed. Lastly, Section 4.6 summarizes and concludes the chapter.

4.2 Background on Resource Allocation

The problem of optimally allocating scare resources across a finite number of competing entities has been studied for a very long time and from a wide variety of viewpoints. Because of the direct correlation between resource allocation problems that occur in applied science and engineering and those that occur naturally in the social and

economic sciences, methodologies involving economic concepts (namely, price and utility) have permeated this field since its inception.

Early on, it became obvious that while completely centralized approaches (where a central entity makes allocation decisions based on complete information) were capable of easily computing an optimal allocation of resources, a more decentralized approach would provide greater scalability as well as reliability in very large systems. This approach was first exemplified by the Arrow-Hurwicz algorithm (Arrow and Hurwicz 1960), in which a central entity announces a price for a resource in question and the units of the system independently compute how much of the resource they need in order to maximize their net return. The computed requests for resources are then sent back to the central entity, and a new price is announced after calculating the difference between total demand and total supply. This process continues until a price is reached that creates a market equilibrium; resources are then distributed accordingly. However, while this price adjustment methodology ensures that an optimal allocation of resources is made, the communication overhead required to make a decision using this technique is prohibitively greater than in the centralized case.

In order to overcome this disadvantage, researchers began to look at completely decentralized (center-free) allocation algorithms (Ho*, et al.* 1980). In the center-free methodology, resource demand information is shared amongst small groups of units, and the resources available within those groups are constantly shifted toward the units which place a greater value on the resources. As such, center-free algorithms yield a constantly improving resource distribution without the need for a coordinating center.

This type of decentralized thinking blossomed in the fields of operational control and mathematical economics, and similar microeconomic approaches were eventually applied explicitly to the allocation of resources in distributed computer systems. For example, the work done by Kurose and Simha (1989) focused on the development of decentralized algorithms to be applied to the classical resource allocation problem of file allocation. By drawing on the set of ideas, methods, and algorithms developed by Ho*, et al*. (1980), this work proved that simple and decentralized algorithms could provide rapid convergence on optimal solutions to file allocation problems.

As time progressed, market and utility-based concepts filtered into many other application spaces within the field of computer science. For example, pricing concepts and utility functions were first applied to network design and performance evaluation over a decade ago from an Internet-based perspective (Cocchi*, et al.* 1993; Shenker 1995). More recently, market-based approaches have become common for managing limited resources such as power and bandwidth within wireless networks. For example, distributed allocation algorithms designed for use within wireless ad hoc networks have been shown to near-optimally allocate resources by using pricing concepts and utility functions in conjunction with techniques developed from linear programming (Curescu and Nadjm-Tehrani 2005; Kao and Huang 2008).

In the past decade, as wireless sensors have begun to emerge as an increasingly important new technology across engineering disciplines, the algorithms developed for resource allocation in distributed computer networks have been quickly transitioned for implementation in WSNs. For example, it has been shown that utility functions can assist large-scale sensing networks in achieving global objectives in a decentralized fashion

using only local information (Byers and Nasser 2000). In this approach, the resource constraints present in WSNs motivate the need for flexible objective functions which allow nodes to choose their role over time, with the goal of optimizing the total utility derived over the lifetime of a network instead of optimizing present resource allocations without regard to future costs.

Other work in this area has focused on the use of utility-based resource allocation techniques to distribute network resources in the wake of multiple application-driven performance objectives. For example, Eswaran*, et al.* (2008) developed a receiver-centric, price-based decentralized algorithm for resource sharing in mission-oriented WSNs. This algorithm is shown to ensure optimal and fair transmission rate allocation amongst a set of multiple data-related objectives ("missions"). Similarly, Jin*, et al.* (2007) employed utility-based concepts to develop an application-oriented flow control framework for heterogeneous WSNs. In this framework, wireless channel usage and sensor node energy are allocated efficiently such that total application performance is maximized.

The work presented in this chapter builds upon the price and utility-based resource allocation methodologies mentioned above. However, it differs from previous work in WSN resource management in two distinct ways. First, in order to account for a greater emphasis on embedded data processing, this work broadens the previous utility function focus on optimal communication and data flow in order to include computational speed and efficiency. Second, the resource allocation algorithm developed in this chapter is implemented directly on a network of wireless sensor prototypes,

allowing the performance of the proposed algorithm to be evaluated directly on the sensing system it was designed for instead of in a simulated environment.

4.3 Application Scenario

The work presented in this chapter is motivated by the desire to perform advanced data processing tasks within networks of wireless sensors, optimally allocating scarce resources so as to optimize the speed and reliability with which a set of computational tasks can be completed within a WSN. While this market-based resource allocation framework can be easily applied to many application-specific data processing algorithms, a simple application scenario is adopted herein so that the elegance and performance of the market-based framework can be better explained and illustrated. In order to simulate a sensing environment where a WSN is asked to perform multiple data processing tasks concurrently, a benchmark problem is needed that can be used to easily represent a number of different computational tasks with varying resource demands. For this purpose, the *n*-Queens problem is chosen as it is a well-known benchmark for evaluating the performance (*i.e.*, speed and efficiency) of combinatorial optimization or search algorithms.

4.3.1 The n-Queens Problem

The objective of the *n*-Queens problem is to place *n* chess queens on an *n* x *n* chessboard (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

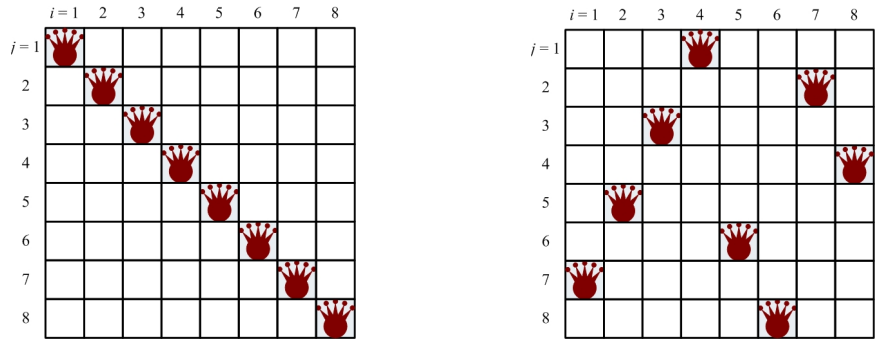


Figure 4.1: (a) Initial board configuration (s_{initial}) and (b) one optimal solution (s_{minimum}) for the 8-Queens problem.

as another queen. An example of an optimal solution to an n-Queens problem can be seen in Figure 4.1b, where one of the many solutions to the 8-Queens problem is presented.

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 a position indexed by (I, J), it is in direct conflict with any queen at position (i, j) if $i = I$ (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:

$$E = \sum_{I=0}^{n-1}\sum_{J=0}^{n-1}\left(\begin{array}{c}\displaystyle\sum_{k=J+1}^{n-1} q_{IJ}q_{Ik} + \sum_{k=I+1}^{n-1} q_{IJ}q_{kJ} + \\ \displaystyle\sum_{k=1}^{\min(n-1-I,\,n-1-J)} q_{IJ}q_{(I+k)(J+k)} + \sum_{k=1}^{\min(n-1-I,\,J)} q_{IJ}q_{(I+k)(J-k)}\end{array}\right) \tag{4.1}$$

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 eliminate 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) where $i = j$, as seen in Figure 4.1a for the 8-Queen problem. Clearly, in this initial state, each queen is in conflict with all other queens. New search states can then be 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 search state generation method allows for significantly faster convergence of the optimization problem, as the first two terms of the objective function (Equation 4.1) can be ignored.

The *n*-Queens problem is an ideal testbed for the market-based resource assignment algorithm proposed in this study because it allows us to easily explore multiple computational tasks of varying complexity by simply increasing the *n*-Queens problem size (namely, by increasing *n*). Specifically, the WSN in this study will be asked to simultaneously solve four *n*-Queens tasks of varying complexity (25-Queens, 50-Queens, 75-Queens, and 100-Queens).

4.3.2 Wireless Parallel Simulated Annealing (WPSA)

There are many existing methods capable of finding or approximating solutions to NP-hard combinatorial optimization problems like *n*-Queens (Rohl 1983; Sosic and Gu

1991; Homaifar, *et al.* 1992). Because an exact solution to these types of problems may require a number of computational steps that grows faster than any finite power of the size of the problem, it is often desirable to use methods that approximate an optimal solution instead of spending the time and computational resources required to find an absolute global optimum. In this study, the wireless parallel simulated annealing (WPSA) algorithm developed by Zimmerman and Lynch (2009) is adopted as a stochastic technique capable of generating approximate solutions to combinatorial optimization problems using the embedded computational resources residing within an ad-hoc WSN. WPSA is a parallel implementation of the traditional simulated annealing (SA) search algorithm, modified explicitly for use within WSNs where overall communication is to be minimized in order to preserve communication bandwidth and power.

4.3.2.1 The Simulated Annealing Search Algorithm

The SA methodology, originally proposed by Kirkpatrick, *et al*. (1983), is modeled after the annealing process of material physics, where a solid substance is melted at a high temperature and then slowly cooled, eventually obtaining an optimal thermal energy state amongst a near-infinite number of atomistic configurations. This annealing procedure can be viewed as a natural optimization problem where the objective is to find an atomistic configuration that represents the absolute minimal energy state possible for a given material. In a similar sense, "simulated" annealing solves optimization problems by representing each possible configuration of optimization parameters as a distinct "atomistic" state, *s*. The SA process attempts to find an assignment of values to these optimization parameters that minimizes an objective

function, $E(s)$. The SA method is presented in much greater detail in Section 3.2 of this dissertation. In the case of the n-Queens optimization problem, each possible chessboard configuration is a distinct state, s, and is represented by a vector of size n containing the column in which a chess queen is present for each row 1 through n. The objective function, $E(s)$, then represents the number of conflicts between queens in a given chessboard configuration, s. In this study, the $E(s)$ is calculated using Equation 1. Here, $E(s)$ takes on integer values with the minimum value of $E(s)$ being 0 (representing no queen conflicts).

As applied to the n-Queens problem, the SA approach begins by adopting an initial system state, s_{initial}, seen in Figure 4.1a for the 8-Queens problem. Then, a new board configuration, s_{new}, is generated by swapping the queens laying on two randomly selected rows, while retaining each queen's initial column. The objective function value (number of queen conflicts) of this new state, $E(s_{\text{new}})$, is then compared with the objective function value of the old state, $E(s_{\text{new}})$, and the new state is probabilistically accepted or rejected based on the Metropolis criterion (Metropolis*, et al.* 1953): 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, and T is the simulated annealing temperature of the system. The addition of the $T \cdot |\ln(U)|$ term allows the system to periodically accept suboptimal states in hopes of avoiding premature convergence on a local minima.

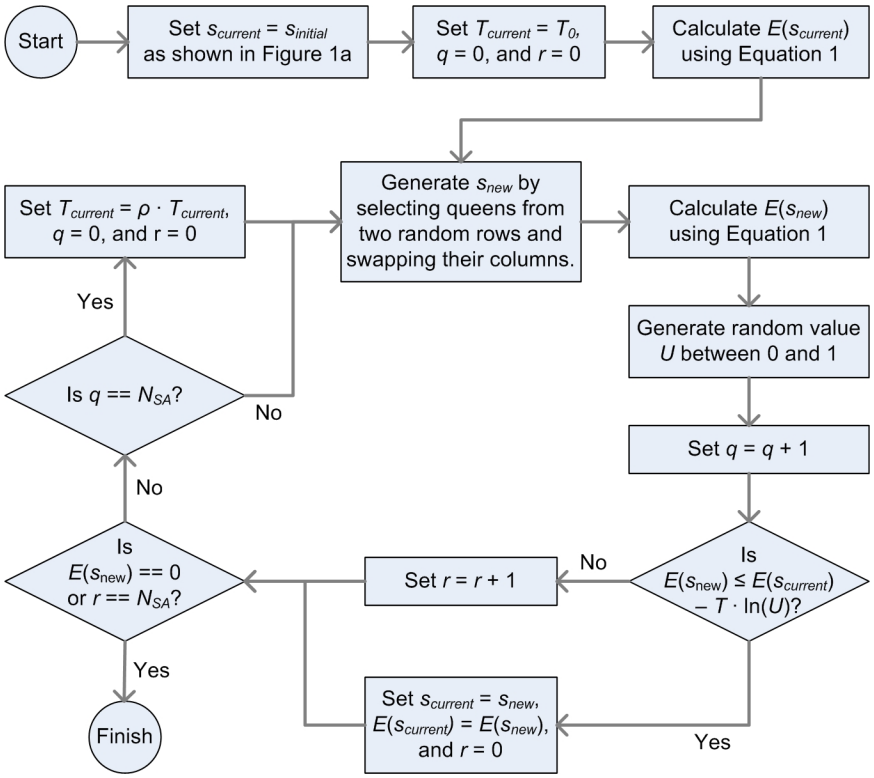


Figure 4.2: Flowchart for a simulated annealing approach to the n-Queens problem.

The SA cooling schedule used in this study assigns a high initial temperature, T_0, at the outset of the search, and then proceeds to evaluate a predefined number (N_{SA}) of newly generated board configurations based on the criteria presented in Equation 4.2. After N_{SA} states have been evaluated, the system temperature is lowered by a factor of ρ, such that $T_{i+1} = \rho \cdot T_i$, and an additional N_{SA} states are generated at the new, lower temperature. This process continues until either a chessboard configuration is found with zero queen conflicts (s_{min}) or N_{SA} consecutive states have been generated which do not meet the criterion presented in Equation 4.2. A graphical illustration of the SA approach to the *n*-Queens problem can be seen in Figure 4.2.

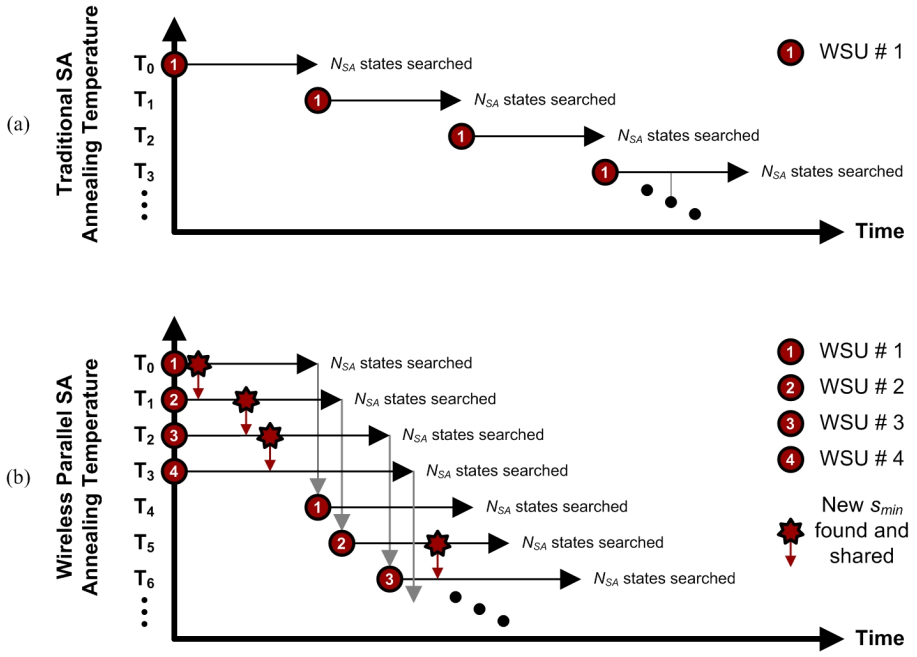


Figure 4.3: (a) Traditional serial SA search progression run on one wireless sensor vs. (b) Wireless parallel SA search progression run on four wireless sensors.

4.3.2.2 Parallelized Simulated Annealing for use in WSNs

Because of the large number of objective function evaluations required when using SA-based optimization, many parallel SA techniques have been developed that, when run on a large number of processors, can drastically increase the speed with which a solution to an optimization problem such as *n*-Queens can be reached (Greening 1990). Unfortunately, most of these parallel methods require a large amount of communication amongst processors (for example, communication before and after each state selection). As such, these approaches are impractical for use within dense networks of wireless sensors, where both communication bandwidth and portable power (namely, battery power) at each node are limited. However, the WPSA method for parallel SA optimization within WSNs (Zimmerman and Lynch 2009) was designed to account for this limitation on processor-to-processor communication. The WPSA algorithm functions

by decomposing the traditionally serial SA search process (which is continuous across all temperature steps) into a set of smaller searches, each of which corresponds to a given temperature step and begins with the best search state yet visited. This concept is shown in Figure 4.3. Because each smaller search problem can be completed by any available wireless sensor, this method allows multiple temperature steps to be searched concurrently, leading to a significant speedup in the overall optimization process.

In the implementation of WPSA used in this study, a user-initiated n-Queens optimization task along with a user-defined initial temperature, T_0, can be randomly assigned to any one sensing node available for computation. If additional sensing nodes are available in the network, this first sensor, n_0, can then assign a WPSA search starting at the next temperature step, $T_1 = \rho \cdot T_0$, to a second sensor, n_1, along with information regarding the most optimal system state yet visited. This type of processor inheritance can continue until no more sensing nodes are available.

As the WPSA search continues, information regarding newly found, increasingly optimal states is passed downwards through the network. In this way, all sensors are aware of search progress that has been made at higher temperature steps, maximizing the effectiveness of the WPSA search at a given temperature step and maintaining the continuity of the serial SA process. When a sensor detects a state, s, with a lower $E(s)$ 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, s, has a lower objective function value than the most optimal state the child has yet visited, s_c, then the child will then restart its N_{SA} 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, that child will merely restart its N_{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.

If a given sensor, n_i, detects an optimal solution, (*i.e.*, an objective function value equal to zero), it will order the rest of the network to discontinue the WPSA search and will alert the network end-user of the discovered results. However, if sensor n_i finishes its part of the WPSA search without having converged on a solution (*i.e.*, new states are still being accepted), it will alert its successor, sensor 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 WPSA search at a lower temperature step. If it has no successor, sensor n_i will automatically begin computation at temperature step T_{i+1}.

The WPSA implementation naturally parallelizes the SA search process without incurring hefty communication overhead. While it drastically reduces communication between nodes, this reduction comes at the cost of computation. In other words, the serial search displayed in Figure 4.3a will search over a deterministic number of states. If a total of Q temperature steps are searched, the number of examined states is $Q \cdot N_{SA}$. This does not change for some pure parallel implementations of SA. However, in the proposed WPSA, searches at a given annealing temperature can be restarted when a parent node locates a state corresponding to a new, lower $E(s)$ value, meaning that the total number of states searched at a temperature step will be greater than or equal to N_{SA}. Therefore, the

total number of states selected will likely be greater than $Q \cdot N_{SA}$. It should be noted, however, that performing more searches than $Q \cdot N_{SA}$ will often result in the identification of a more optimal state.

4.4 Market-Based Task Assignment

With an application scenario in place, it is now possible to outline the decentralized market-based approach used in this study to optimally distribute scarce WSN resources across several competing computational objectives (namely, four *n*-Queens problems). The ideas proposed herein are drawn from free-market economies, which are incredibly complex systems that are optimally controlled in a decentralized manner. In a free-market economy, 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 the allocation of system resources such as memory or network bandwidth (Clearwater 1996). Perhaps the greatest benefit of market-based optimization is that it yields a Pareto-optimal solution. A Pareto-optimal market is one in which no market participant can reap the benefits of higher utility or profits without causing harm to other participants when a resource allocation is changed (Mas-Colell*, et al.* 1995).

Conceptually, it would be somewhat trivial to develop a simple auction-based system which could be used in a WSN to crudely assign scarce computational resources (such as CPU cycles or data storage) to various computational tasks while attempting to

optimize a single computational objective (such as minimizing time to task completion). However, it is significantly more valuable to consider a more robust market-based scheme that can optimally allocate resources in the midst of several additional competing and resource-related objectives, such as wireless bandwidth usage and battery consumption. In this study, we attempt to create this type of system through the use of buyer and seller utilities. By embedding within each market agent (*i.e.*, wireless sensor) the desire to maximize an individual utility function, competing goals can be settled through market means (supply and demand functions, price, etc). The result is a Pareto-optimal allocation of scare system resources.

In this chapter, we are particularly interested in three distinct (but possibly competing) performance objectives: (O_1) completing all required computational tasks as quickly as possible, (O_2) minimizing power consumed by the sensor network, and (O_3) functioning as robustly and as reliably as possible in the wake of limited communication bandwidth and uncertain sensor performance. In order to measure the ability of the market-based resource allocation framework to address these three objectives, four performance metrics are created and utilized: (M_1) the time required to complete each task, (M_2) the number of wireless transmissions required to complete each task (which would be directly correlated to overall energy usage by the wireless network), (M_3) the number of sensor failures encountered during each task, and (M_4) the number of communication failures encountered during each task.

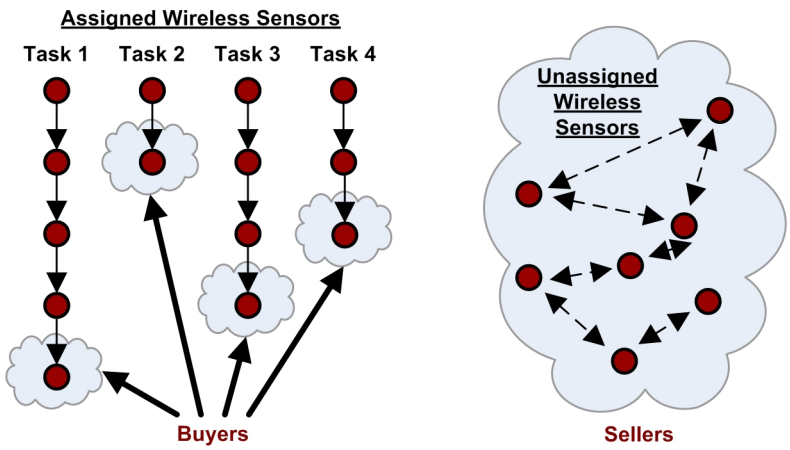


Figure 4.4: Buyer/seller distinction for market-based task assignment.

4.4.1 Buyer/Seller Framework

As seen in Figure 4.4, the sellers in this market-based allocation technique can be defined as the set of sensors in the wireless network not currently working on any computational task. These WSUs 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 (in this study, each *n*-Queen search problem). In order to simultaneously address all three performance objectives (O_1, O_2, and O_3) in a streamlined manner, buyers and sellers focus on different goals. In this market, sellers work to minimize network power consumption (O_2). Because the wireless radio consumes significantly more power than any other WSU hardware component (Lynch*, et al.* 2004), sellers gain utility by minimizing the number of wireless communications required to complete each task. Buyers, on the other hand, work both to minimize the overall time spent computing (O_1) and to maximize sensor and communication reliability (O_3). Thus, buyers gain utility by minimizing CPU time required to complete each task and minimizing the risk of lost CPU time due to sensor or communication failure.

4.4.2 Formulation Buyer-Side Utility Functions for WPSA

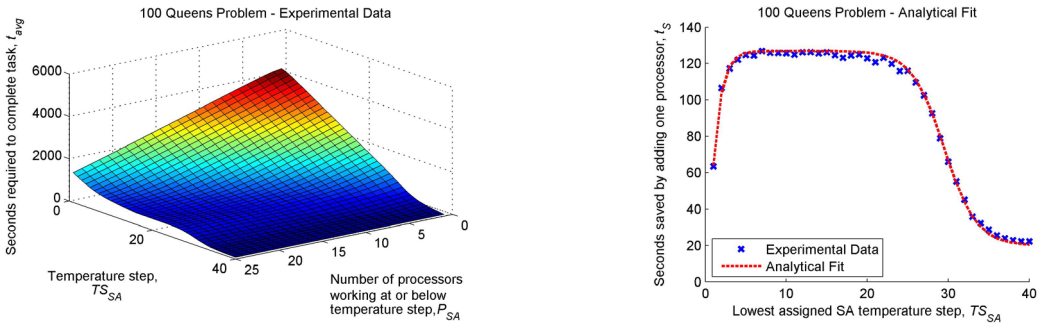
In light of this framework, it is now necessary to explicitly derive utility functions associated with both buyers and sellers engaged in solving multiple combinatorial optimization (CO) problems by WPSA. These utility functions will govern whether or not a buyer for a given CO problem will place a bid on the services of a seller and which buyer, if any, a seller will sell its computing services to. On the buyer side, a utility function, U_B, can be intuitively thought of as the total amount of time a computational task saves by adding an additional processing node, and can therefore be defined as follows:

$$U_B = t_S - \alpha_B \cdot t_{SF} - \beta_B \cdot t_{CF} \tag{4.3}$$

where t_S, t_{SF}, t_{CF}, are time values and α_B and β_B are weighting factors, as defined in detail below.

4.4.2.1 Formulation of t_S

For any computational task, the value of t_S represents the expected decrease in computation time required to complete the task brought about by the addition of one processor to those processors currently working on the CO problem. While there is often no way to directly formulate an analytical expression for this value, a trend can be established by looking at the average amount of time it takes a task to complete from a given point in its computation while utilizing a given number of processors. In the case of a combinatorial optimization task like *n*-Queens being solved using the WPSA algorithm,



(a) (b)

Figure 4.5: For the 100-Queens problem, (a) experimentally collected time to completion data and (b) analytical fit for t_S.

t_S can be expressed as the difference between the average time, $t_{avg}(P_{SA}, TS_{SA})$, required to complete a CO problem where P_{SA} nodes are currently computing at temperature steps up to TS_{SA}, and the average time, $t_{avg}(P_{SA}+1, TS_{SA}+1)$, required to complete the same task where $P_{SA}+1$ nodes are computing at temperature steps up to $TS_{SA}+1$:

$$t_S = t_{avg}(P_{SA}, TS_{SA}) - t_{avg}(P_{SA}+1, TS_{SA}+1) \tag{4.4}$$

Using data gathered over a large number of experimental trials run on a given WSU platform, Figure 4.5a shows the amount of time, $t_{avg}(P_{SA}, TS_{SA})$, required for a given number of processors to solve the 100-Queens problem when the first node in the WPSA chain is at a given temperature step and no processors are allowed to rejoin the task once they have completed their assigned search. This data can be used to empirically determine a relationship between the number of processors currently working on a 100-Queen problem (P_{SA}), the lowest SA temperature step being searched (TS_{SA}), and the amount of time saved from the addition of a processor (t_S), as seen in Figure 4.5b. It is found that the relationship between t_S and TS_{SA} is independent of P_{SA}, and thus can be approximated by an easily computable algebraic function:

Table 4.1: Coefficients for calculating t_S.

	Number of Queens (C_{SA})			
	25	50	75	100
a	0.0	1.0	8.0	20.0
b	12.3	35.9	73.5	126.9
c	13.0	23.0	27.0	29.5
d	8.3	19.7	39.5	63.4

$$t_S\left(TS_{SA}\right) = a + \frac{b-a}{1+e^{0.5\cdot\left(TS_{SA}-c\right)}} - \left(b-d\right)\cdot e^{1-TS_{SA}} \tag{4.5}$$

where the values for *a*, *b*, *c*, and *d* are specific to each task complexity, C_{SA}, and are tabulated in Table 4.1. The quality of the analytical fit provided by this function for the 100-Queens problem can also be seen in Figure 4.5b. Fits of similar quality can be found for all other problem complexities considered in this study (namely, 25-Queen, 50-Queen, and 75-Queen).

4.4.2.2 Formulation of t_{SF}

The failure of a WSU could occur during the execution of a computational task. For example, if a WSU fully depletes its battery, it will cease to operate. In the WPSA computational method, if a wireless sensor fails, the continuity of the WPSA search would be lost at and below the failed node. Therefore, the sensors below the failed node would be reassigned starting with an assignment at the failed node's temperature step. Hence, the buyer must account for its exposure to the risks associated with a failed WSU. Clearly, as the number of nodes working on a given CO problem increase, the buyer's exposure to the risk of a failed node increases.

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

$$t_{SF}\left(P_{SA}, N_{SA}\right) = \left(1 - \prod_{c=1}^{P_{SA}+1} p_{SS}\big|_{environment}\right) \cdot t\left(N_{SA}\right) \tag{4.6}$$

where $t(N_{SA})$ is the average time required for one sensor to complete N_{SA} search iterations and p_{SS} 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, but is typically quite high (>0.95).

The probability of a failed sensor should reflect the real-time state of the WSU. For example, if a battery source is getting low, the probability that the sensor node will complete its tasks reduces. Hence, p_{SS} could vary during the execution of the computational task. In this study, p_{SS}, is assumed fixed in order to simplify the analysis. As such, we can write:

$$\prod_{c=1}^{P_{SA}+1} p_{SS}\big|_{environment} = {p_{SS}}^{P_{SA}+1} \tag{4.7}$$

4.4.2.3 Formulation of t_{CF}

For any computational task, t_{CF} represents the expected processing time lost due to communication failure brought about by the addition of one processor. Like t_{SF}, this quantity can also be derived analytically. If any sensor loses communication with its parent for a prolonged time while it is involved in a WPSA task (for example, if it becomes blocked by a physical impediment), any work done by the failed node and all nodes below it would be lost. As such, t_{CF} 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 chain permanently loses parental communication. The probability of failure of any chain of parent-child communication links is dependent on the signal strength (RSSI) of each respective wireless communication link, c. Clearly, as the RSSI goes down, the probability of a prolonged loss of communications goes up. As such, an analytical value for t_{CF} can be expressed as:

$$t_{CF}\left(P_{SA}, N_{SA}, RSSI\right) = \left(1 - \prod_{c=1}^{P_{SA}} \frac{p_{CS}}{1 + e^{-0.4\left(40.0 + RSSI_c\right)}}\right) \cdot t\left(N_{SA}\right) \tag{4.8}$$

where $t(N_{SA})$ is as before and p_{CS} is the probability that a given communication link of perfect signal strength is not permanently destroyed during N_{SA} search iterations. Again, this value is dependent on the wireless sensor platform being used and the environment in which it is deployed, but is usually also quite high (>0.9).

Having examined in more detail the derivation of t_S, t_{SF}, and t_{CF}, it can now be seen from Equation 4.3 that α_B and β_B are weighting parameters that allow a WSN to prioritize between speedup (O_1), communication reliability (O_3), and sensor reliability (O_3). This type of weighting creates an extremely adaptable computing environment that can change, in real-time, to shifting computing needs within a WSN.

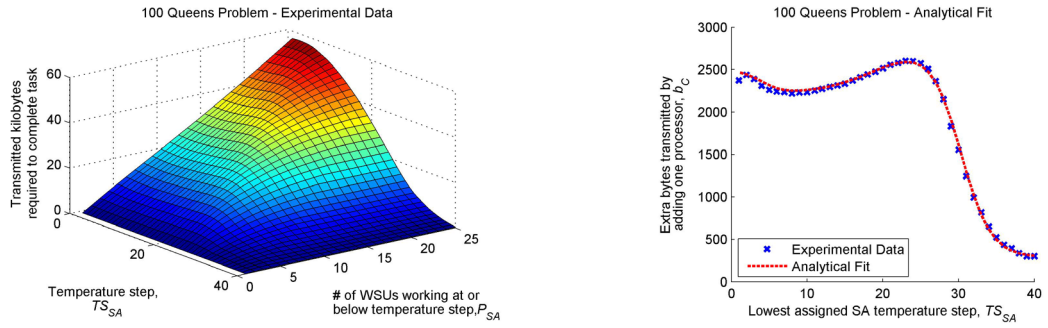
4.4.3 Formulation of Seller-Side Utility Functions for WPSA

On the seller side of this market-based allocation procedure, a somewhat simpler utility function, U_S, can be developed in a similar fashion to U_B. 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. Since the majority of power consumption in a wireless sensing device comes from the wireless radio (which, as stated before, consumes significantly 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_S can be defined as follows:

$$U_S = - b_C \tag{4.9}$$

4.4.3.1 Formulation of b_C

For any CO problem, the value of b_C represents the expected increase in communicated bytes required to complete the task brought about by the addition of one processor. Much like t_S, there is often no way of directly formulating an analytical expression for this value. As such, a trend can be established for any computational task



(a) (b)

Figure 4.6: For the 100-Queens problem, (a) experimentally collected communication data and (b) analytical fit for b_C.

by looking at the average number of bytes communicated when a task of complexity C_{SA} converges 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, b_C can be expressed as the difference between the average number of communicated bytes, $b_{avg}(P_{SA}, TS_{SA})$, required to complete a search where P_{SA} nodes are currently searching up to temperature step TS_{SA} and the average number of communicated bytes, $b_{avg}(P_{SA}+1, TS_{SA}+1)$, required to complete a search where $P_{SA}+1$ nodes are currently searching up to temperature step $TS_{SA}+1$:

$$b_C = b_{avg}(P_{SA}, TS_{SA}) - b_{avg}(P_{SA}+1, TS_{SA}+1) \tag{4.10}$$

Using experimentally gathered data, Figure 4.6a shows the amount of wireless communication (in bytes), $b_{avg}(P_{SA}, TS_{SA})$, required for a given number of processors to solve the 100-Queens problem when the first node in the WPSA chain is at a given temperature step and no processors are allowed to rejoin the task once they have completed their assigned search. This data can be used to determine a relationship between the number of processors currently working on a 100-Queen problem (P_{SA}), the

Table 4.2: Coefficients for calculating b_C.

	Number of Queens (C_{SA})			
	25	50	75	100
m_1	5.00	10.20	15.30	22.25
n_1	0.40	0.50	0.55	0.50
r_1	13.00	24.40	28.00	30.25
m_2	0.00	-0.60	-2.00	-5.75
n_2	0.00	0.40	0.40	0.25
r_2	0.00	13.00	15.00	20.00
m_3	1.75	1.75	1.80	2.50
n_3	0.80	0.80	1.00	0.70
r_3	4.00	4.00	4.00	4.00
q	0.00	0.20	1.20	2.50

lowest SA temperature step being searched (TS_{SA}), and the increase in wirelessly communicated bytes associated with the addition of one processor (b_C). It is found that the relationship between b_C and TS_{SA}, as seen in Figure 4.6b, is independent of P_{SA}, and thus can be approximated by an easily computable algebraic function:

$$b_C\left(TS_{SA}\right) = \sum_{i=1}^{3}\left[m_i - \frac{m_i}{1+e^{n_i\cdot\left(r_i - TS_{SA}\right)}} \right] + q \tag{4.11}$$

where values for m, n, r, and q are specific to each task complexity, C_{SA}, and are tabulated in Table 4.2. The quality of the analytical fit provided by this function for the 100-Queens problem can also be seen in Figure 4.6b. Fits of similar quality can be found for all other problem complexities.

4.4.4 Wireless Task Assignment Algorithm

Having developed utility functions associated with both buyers and sellers, it is now possible to create a methodology with which sensors in a WSN can buy and sell processing time in order to create an optimal distribution of resources while successfully

completing multiple computational objectives (*i.e.*, multiple *n*-Queen problems). By expanding on the fundamental principles of an auction, the following procedure is developed:

1) All sensing units not currently computing will broadcast their availability to the network (as market sellers).
2) The wireless sensors having most recently joined each existing computational task (market buyers) will calculate U_B based on the computational task they are working on, and submit a bid of U_B to each available market seller if $U_B > 0$.
3) Market sellers will calculate U_S based on each proposed computational job offer (bid) they receive, and will wait for a short period of time for other bids to be received.
4) Once all bids have been received, market sellers will calculate their expected profit from each proposed job using a market power / speed exchange rate (γ_M) that represents the minimum number of seconds of computational speedup that must be gained in order to warrant an additional byte of communication:

$$profit = U_B - \gamma_M \cdot U_S \tag{4.12}$$

5) Market sellers will choose the bid that generates the greatest non-negative profit, and will join the corresponding computational task.

Using this algorithm, computational assignments will be distributed throughout the network in such a way that the overall utility of the market is maximized. By default, this methodology works to maximize the speed with which a set of computational tasks can be completed. But because of the addition of the weighting parameters, α_B, β_B, and γ_M, the resulting framework is also capable of optimally adapting, in real-time, to shifting computing needs or resource limitations 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_B value in order to reflect the added emphasis on avoiding communication failure. Similarly, α_B can be used to emphasize sensor reliability and γ_M to stress power savings.

4.5 *n*-Queens Testbed and Results

In order to validate the market-based task assignment methodology proposed in this study, the four performance metrics (M_1 through M_4) outlined in Section 4.4 are evaluated using a network of wireless sensing prototypes. To this end, both the WPSA algorithm (Section 4.3.2.2) and the market-based task assignment algorithm (Section 4.4.4) are embedded within a network of 20 *Narada* wireless sensors, seen in Figure 4.7. The *Narada* wireless platform is described in greater detail in Section 1.6.2.

4.5.1 Performance Evaluation – Computational Speed

The first performance metric evaluated, M_1 (time to completion), involves the ability of the proposed market-based resource allocation method to improve the speed with which multiple computational objectives can be completed within a wireless

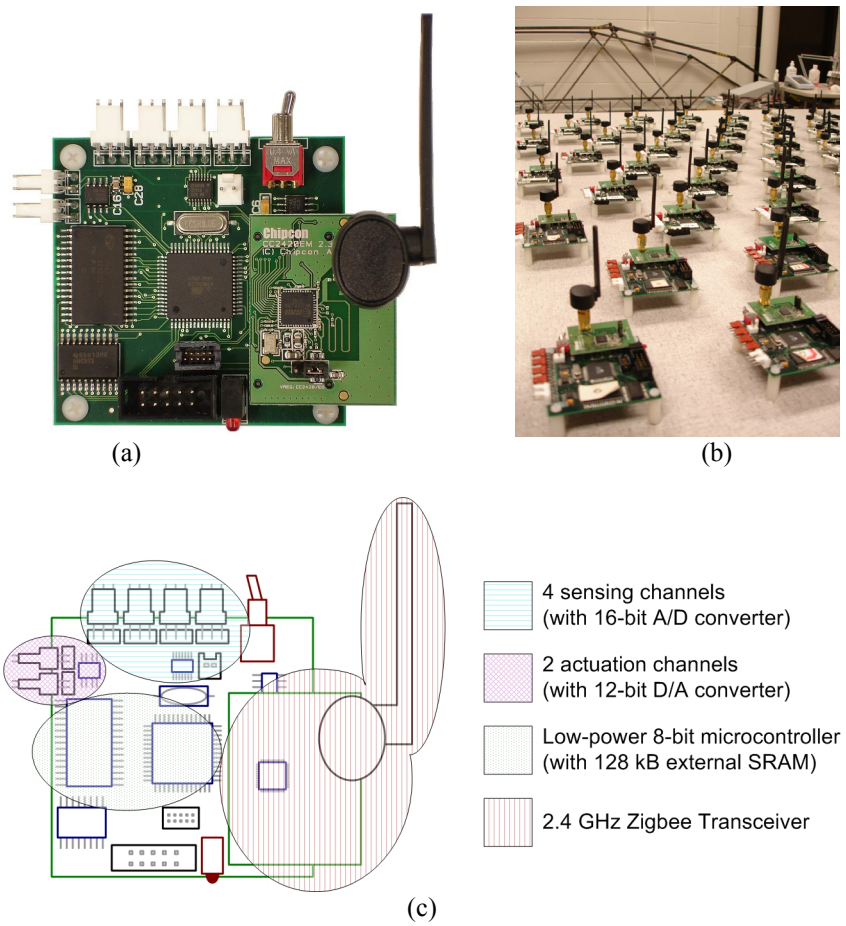


Figure 4.7: (a) Close-up of a *Narada* wireless sensing prototype, (b) a network of *Narada* wireless sensors, and (c) a schematic representation of *Narada*'s core components.

network. Specifically, we will show that for a given wireless network size (between 4 and 20 nodes), the market-based allocation method can optimally assign available processors to four competing computational tasks.

In order to evaluate this metric, four combinatorial optimization tasks of varying complexity (*i.e.*, 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 search chain associated with their given *n*-Queens task (performing a WPSA search at temperature T_0). After these initial assignments have been made, a pool of additional processing nodes (containing between 0 and 16 Narada

wireless sensors) is made readily available for computational use. At this point, the market-based bidding process begins with each of the four master nodes "bidding" on the computational services of the additional sensing nodes, and resource allocation proceeds as described in Section 4.4.4. If a "master" node finishes the WPSA search at its assigned temperature step without finding a global minimum, it will pass its "master" status on to its child, making itself once again available for computation on any of the four computational tasks. Similarly, if a global minimum is reached, all nodes will be released to join computation on any of the remaining tasks. Because we are strictly evaluating computational speedup, α_B, β_B, and γ_M are all set to zero in this test setup. As seen in Equation 4.3 and Equation 4.10, this allows us to negate the impact of wireless bandwidth (b_C) and communication/sensor reliability (t_{CF} and t_{SF}) by isolating computational speed (t_S) in the utility function calculations.

In order to begin evaluating the speedup performance of the proposed market-based task distribution methodology, it is first necessary to establish a benchmark against which to compare timing results. In order for the market-based method to be proven effective, it must be shown that a WSN utilizing the proposed method is capable of completing the four assigned tasks at least as quickly as if an optimal number of processors had been assigned *a priori* to each task at the outset of computation. In the *a priori* case, a static subset of processors remain with a given task throughout the entirety of its computation. Even a certain amount of degradation in computing speed with respect to this type of *a priori* optimization may serve to validate the market-based method, as the scalability and failure tolerance of real-time task assignment greatly outweighs any small time savings when dealing with full-scale deployments in harsh field settings;

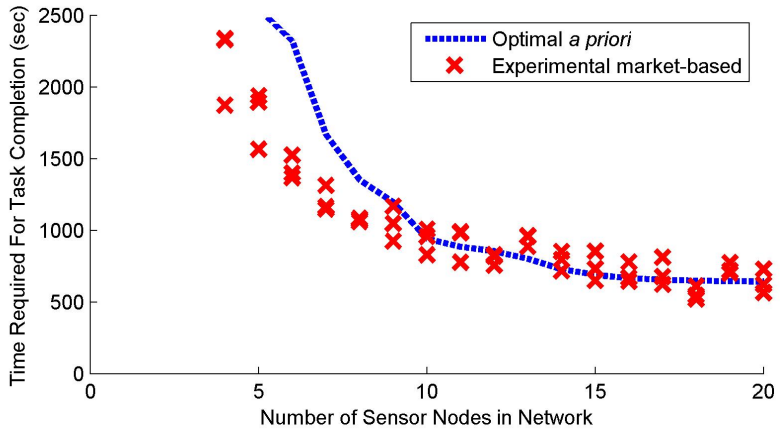


Figure 4.8: Time required to complete four distinct *n*-Queens problems using both market-based and optimal *a priori* resource assignment methods versus number of WSU nodes in sensing network.

specifically, *a priori* assignment of tasks can quickly become suboptimal in the wake of sensor failure. Note also that optimal *a priori* task distributions become exponentially more difficult to calculate as additional tasks or processors are added.

Experimental data is gathered using Narada networks ranging in size from 4 to 20 sensors. In each experimental instance, the WSN is asked to solve all four *n*-Queens problems. In total, each experimental instance is run three times. Figure 4.8 compares the experimental market-based performance against the performance of an *a priori* resource allocation scheme with respect to the total time required for each sensor network to complete all assigned tasks. It can be seen from this plot that the market-based task distribution method performs as well, if not better than an optimal *a priori* assignment of tasks. Note that there is inherent scatter in the market-based results, as the SA algorithm itself fluctuates somewhat in its speed to convergence. But on average, it can be seen that the proposed market-based method actually performs better than an *a priori* optimal distribution.

4.5.2 Performance Evaluation – Wireless Bandwidth Usage

Having confirmed the ability of the proposed resource allocation algorithm to optimize the speed with which multiple computational tasks can be completed within a WSN of a given size, it becomes necessary to evaluate the second of Section 4.4's performance metrics, M_2 (number of transmissions). This metric involves the ability of the proposed method to create a Pareto-optimal resource distribution which allows for a controlled balance between computational speed and wireless bandwidth usage. In order to evaluate this metric, the same four *n*-Queens problems are assigned to a network of 20 Narada wireless sensors, with parameters α_B and β_B set to zero. As seen in Equation 4.3 and Equation 4.10, these parameter settings allow us to isolate wireless bandwidth (b_C) and computational speed (t_S) while negating communication/sensor reliability (t_{CF} and t_{SF}) in the utility function calculations. As before, the WSN is asked to solve all four *n*-Queens problems in a large number of experimental trials, each of which is conducted with a different value of γ_M (ranging from 0.00 to 0.07 and representing increasing emphasis on wireless bandwidth consumption).

As seen in Figure 4.9, a distinct tradeoff can be observed between the amount of time required to complete all four tasks (Figure 4.9a) and the total amount of data transmitted during the completion of these tasks (Figure 4.9b) as the value of γ_M is increased. This is evidence that the market-based methodology proposed herein is sufficiently expressive that competing computational objectives such as computing speed and power consumption can be effectively prioritized through the market exchange rate γ_M.

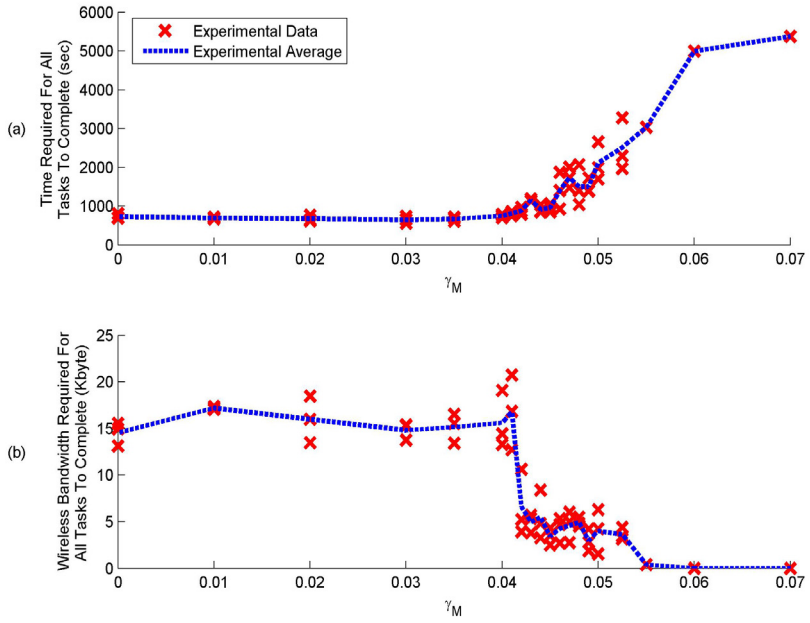


Figure 4.9: (a) Time and (b) amount of wireless communication required to complete four computational tasks using market-based resource assignment vs. weighting parameter γ_M.

4.5.3 Performance Evaluation – Sensor Reliability

The third performance metric to be evaluated, M_3 (number of sensor failures), involves the ability of the proposed market-based method to create a distribution of resources which allows for a Pareto-optimal balance between computational speed and risk of sensor failure. Because the risk of sensor failure in the WPSA algorithm is directly correlated to the size of the WPSA computational chains, this metric can be evaluated by viewing the tradeoff between time to completion and WPSA computational chain size. As such, the same four *n*-Queens problems are assigned to the network of 20 Narada wireless sensors, with parameters γ_M and β_B set to zero. As seen in Equation 4.3 and Equation 4.10, these parameter settings allow us to isolate sensor reliability (t_{SF}) and computational speed (t_S) while negating communication reliability and wireless bandwidth (t_{CF} and b_C) in the utility function calculations. Then, in a large number of

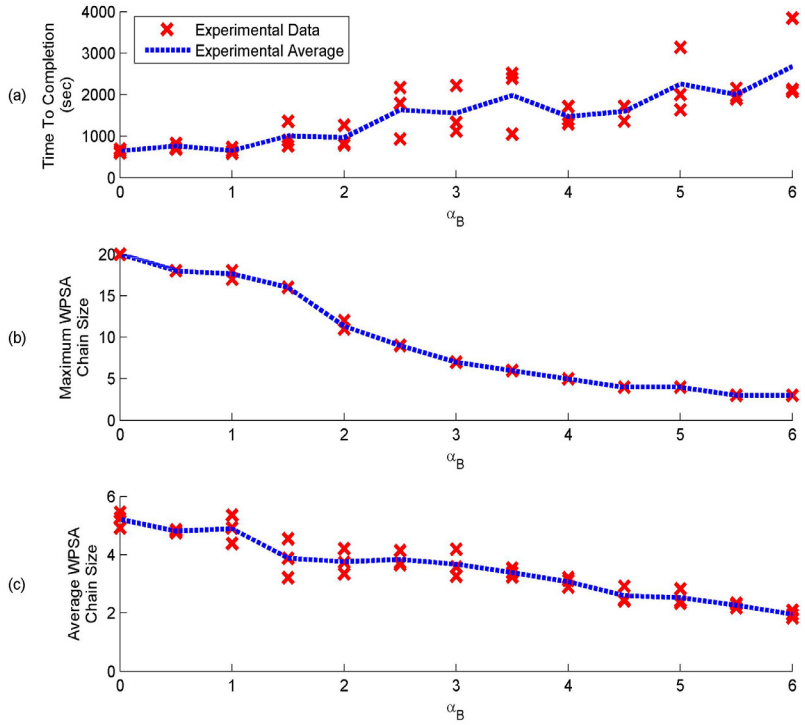


Figure 4.10: (a) Time required, (b) maximum WPSA chain size reached, and average WPSA chain size required to complete four computational tasks on 20 WSUs using market-based resource assignment while varying weighting parameter α_B.

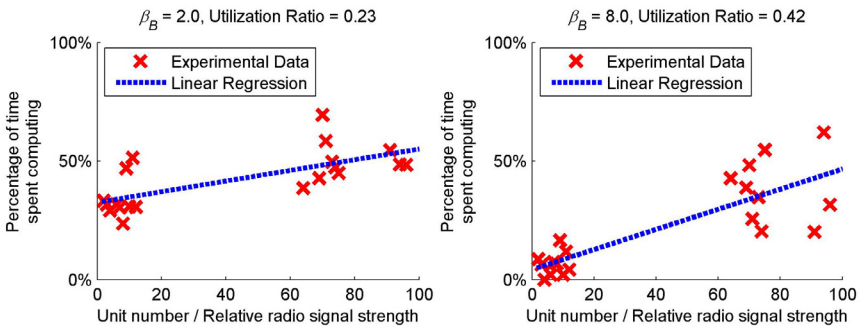
experimental trials, the WSN is asked to solve all four *n*-Queens problems with values of α_B varying between 0 and 6, representing increasing emphasis on time lost due to sensor failure.

As can be seen in Figure 4.10, a distinct tradeoff can be observed between the amount of time required to complete all four tasks (Figure 4.10a) and both the maximum length (Figure 4.10b) and the average length (Figure4.10c) of the WPSA computational chains formed to solve the *n*-Queens problems, as the value of α_B is increased. The fact that the chain size decreases and the time to completion increases with higher values of α_B is evidence that the market-based methodology is effectively and autonomously prioritizing between computing speed and risk of sensor failure.

4.5.4 Performance Evaluation – Communication Reliability

The last performance metric to be evaluated, M_4 (number of communication failures), involves the ability of the proposed market-based method to create a distribution of resources which allows for a Pareto-optimal balance between computational speed and risk of communication failure. As with metric M_3, because the risk of communication failure in the WPSA algorithm is directly correlated to the size of the WPSA computational chains, this metric can be evaluated by viewing the tradeoff between time to completion and WPSA computational chain size. Again, the same four *n*-Queens problems are assigned to a network of 20 Narada wireless sensors, with parameters γ_M and α_B set to zero. As seen in Equation 4.3 and Equation 4.10, these parameter settings allow us to isolate communication reliability (t_{CF}) and computational speed (t_S) while negating sensor reliability and wireless bandwidth (t_{SF} and b_C) in the utility function calculations. Then, a large number of experimental trials are executed with the WSN being asked to solve all four *n*-Queens problems with values of β_B varying between 0 and 10, representing increasing emphasis on time lost due to communication failure.

In the communication case, however, the risk of failure is also directly correlated with the signal strength of a given wireless connection (quantified by the RSSI value of the sensor-to-sensor link). As such, metric M_4 can also be evaluated by looking at the tradeoff between the overall computational speed of the network and any bias that is placed on communication between pairs of wireless nodes with strong wireless connections. Therefore, a separate measure, or "utilization ratio," is defined and calculated for each experimental trial. Intuitively, this ratio can be thought of as a



(a) (b)

Figure 4.11: Plots showing example utilization ratio calculation (via linear regression) for experimental cases where (a) $\beta_B = 2.0$ and (b) $\beta_B = 8.0$.

measure of the relationship between a wireless node's signal strength and how often it is utilized in computation. In running this set of experimental trails, the radio on each *Narada* wireless sensor is programmed to output with a signal strength proportional to its unit number (between 1 and 99). This way, a measure of signal strength bias can be calculated for each experimental trial by plotting the amount of computation a given WSU performs (in percentage of time) versus the unit number of that WSU (*i.e.*, its relative radio signal strength). Using this plot, a linear regression can be drawn through the resulting points, and the slope of this line can be used to quantify the "utilization ratio", or the change in WSU utilization divided by the change in RSSI. An example of this concept is illustrated graphically in Figure 4.11. Note that in order to more clearly show the linear regression, 10 sensors are used with low unit numbers (between 1 and 15), and 10 sensors are used with high unit numbers (between 60 and 100).

As can be seen in Figure 4.12, a distinct tradeoff can be observed between the amount of time required to complete all four tasks (Figure 4.12a) and both the maximum length (Figure 4.12b) and the average length (Figure 4.12c) of the corresponding WPSA computational chains, as the value of β_B is increased. Additionally, as β_B is increased, it

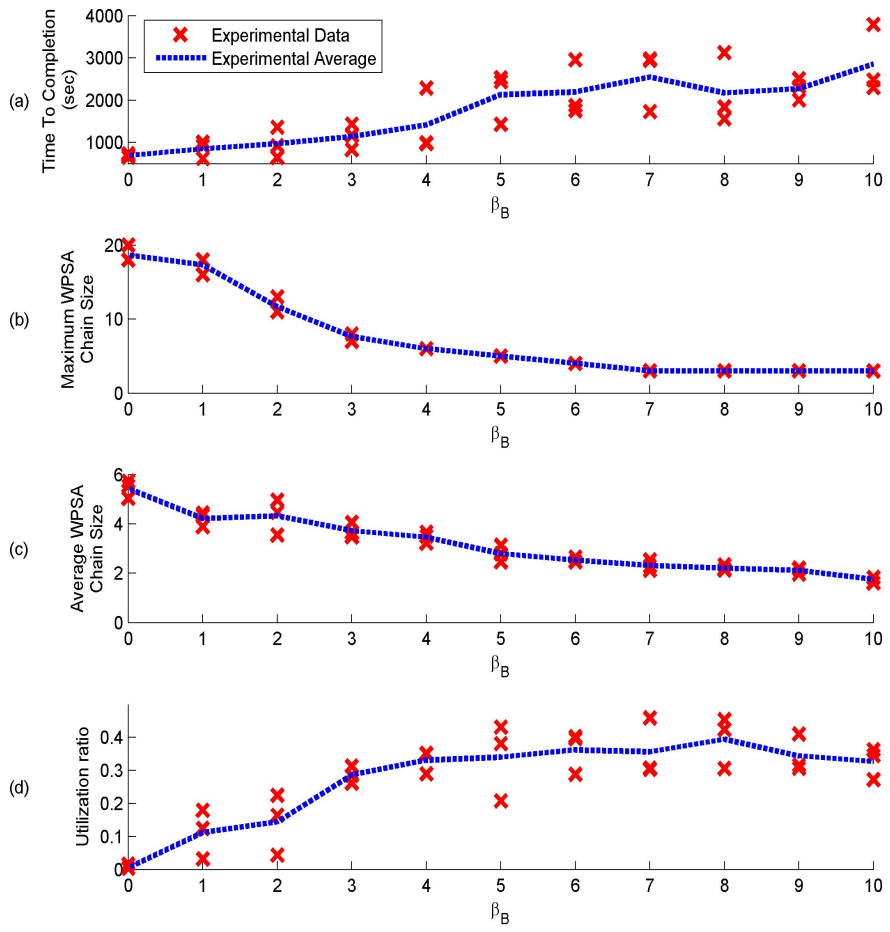


Figure 4.12: (a) Time required, (b) maximum WPSA chain size reached, average WPSA chain size required, and utilization ratio observed while completing four co mputational tasks on 20 WSUs using market-based resource assignment while varying weighting parameter βB.

can also be seen through the utilization ratio that increased preference is placed on sensors with better, more reliable communications channels (Figure 4.12d). This is evidence that the market-based methodology is effectively and autonomously prioritizing between computing speed and risk of communication failure.

4.6 Model Updating Testbed and Results

In order to demonstrate the applicability of the market-based resource allocation methodology developed in this chapter to civil structures, an additional benchmark problem is chosen that can be easily scaled to represent a set of computational tasks, each of which places increasing demand on the computing resources of a WSN. This problem, which has far-reaching implications in SHM, is vibration-based finite element model updating (FEMU) (Doebling*, et al.* 1998). The idea behind FEMU is that by fitting the parameters of a structural model to match experimentally collected sensor data, one can calibrate analytical models, validate design assumptions, and even detect the onset of structural damage or degradation using model-based information as a guide. In this testbed, we consider three finite element (FE) models of a single cantilevered beam, using 5, 10, and 20 beam elements, respectively. In the absence of experimental data, a simple eigenvalue analysis is used to determine the modal frequencies, ω_{Bi}, and mode shapes, ϕ_{Bi}, of the simple cantilever. Then, each of these three FE models is updated using FEMU and the stiffness value (EI_x) of each beam element is found such that the modal properties of each model (ω_{MUi} and ϕ_{MUi}) match the baseline properties (ω_{Bi} and ϕ_{Bi}) as closely as possible. The objective function, E, used to assess the closeness of the model to the true system properties, is based on the modal frequencies, ω_i, and the modal assurance criterion, MAC_i, of the first four modes (Allemang and Brown 1982). The FEMU process is illustrated in Figure 4.13.

As was the case with the *n*-Queens testbed, the wireless parallel simulated annealing (WPSA) algorithm (Zimmerman and Lynch 2009) is adopted as a technique capable of

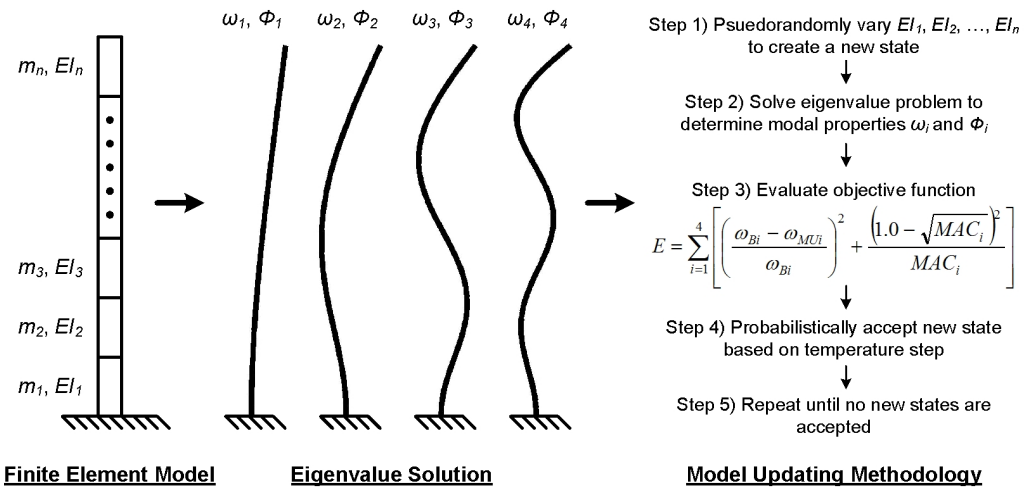


Figure 4.13: Finite element model updating procedure.

generating approximate solutions to FEMU problems using a WSN. As such, both buyer and seller utilities can be defined as before:

$$U_B = t_S - \alpha_B \cdot t_{SF} - \beta_B \cdot t_{CF} \tag{4.12}$$

$$U_S = - b_C \tag{4.13}$$

where

$$t_S\left(TS_{SA}\right) = \frac{\alpha}{1+e^{\lambda_2\cdot\left(TS_{SA}-\beta\right)}} - \left(\alpha-\gamma\right)\cdot e^{\lambda_1\cdot\left(1-TS_{SA}\right)} \tag{4.14}$$

$$t_{CF}\left(P_{SA},N_{SA},RSSI\right) = \left(1-\prod_{c=1}^{P_{SA}}\frac{p_{CS}}{1+e^{-0.4\left(40.0+RSSI_c\right)}}\right)\cdot t\left(N_{SA}\right) \tag{4.15}$$

$$t_{SF}\left(P_{SA},N_{SA}\right) = \left(1-\prod_{c=1}^{P_{SA}+1} p_{SS}\big|_{environment}\right)\cdot t\left(N_{SA}\right) \tag{4.16}$$

$$b_C\left(TS_{SA}\right) = \frac{\alpha}{1+e^{\lambda_2\cdot\left(TS_{SA}-\beta\right)}} - \left(\alpha-\gamma\right)\cdot e^{\lambda_1\cdot\left(1-TS_{SA}\right)} \tag{4.17}$$

Table 4.3: Coefficients for calculating t_S and b_C, tabulated for each finite element model updating problem considered (5 elements, 10 elements, and 20 elements).

t_S

C_{SA}	α	β	γ	λ_1	λ_2
5	235	20	130	1	0.22
10	800	30	460	0.85	0.3
20	3200	25	1750	0.6	0.5

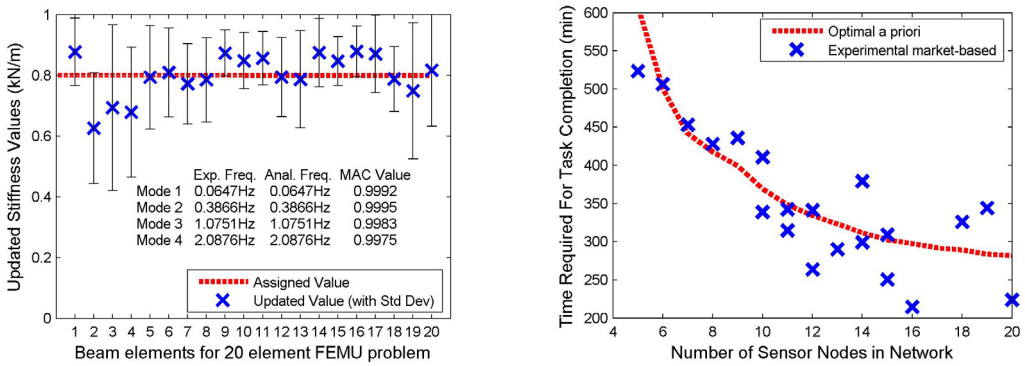
b_C

C_{SA}	α	β	γ	λ_1	λ_2
5	5.5	6	3.9	1	0.6
10	4.6	16.5	2.9	0.75	0.35
20	7	28	4.2	0.15	0.6

Note that the equations modeling t_S and b_C have changed because we are dealing with a different application space (FEMU vs. *n*-Queens). As such, it is necessary to redefine the parameters for each of the above functions. Using a set of experiments similar to those used to create the parameters in Table 4.1 and Table 4.2, a new set of parameter values are calculated. These new parameters are tabulated in Table 4.3.

4.6.1 Performance Evaluation – Computational Speed

Having developed a set of utility functions applicable to the FEMU problem posed in this testbed, it is now possible to validate the market-based methodology proposed in this study. As such, both the WPSA algorithm and the market-based task assignment algorithm are embedded within a network of *Narada* wireless sensor prototypes. For the market-based method to be proven effective from a speed perspective, it must be shown that a WSN utilizing the proposed method is capable of completing the three assigned FEMU tasks (5, 10, and 20 elements) at least as quickly as if an optimal number of processors had been assigned *a priori* to each task at the outset of computation. As such, experimental data is gathered using Narada networks ranging in size from 5 to 20 sensors, with the weighting parameters α_B, β_B, and γ_B all being set to zero. Figure 4.14a shows the ability of the WPSA method to find approximate solutions to the 20 element FEMU problem, and Figure 4.14b compares experimental market-



(a) (b)

Figure 4.14: (a) Averaged 20 element WPSA results and (b) time to completion for market-based method.

based performance against the performance of an *a priori* resource allocation scheme with respect to the total time required for each method to complete all assigned tasks. It can be seen that the market-based task distribution method performs as well as an optimal *a priori* assignment of tasks.

In addition to confirming the ability of the proposed resource allocation algorithm to optimize the speed with which multiple computational tasks can be completed within a WSN of a given size, it is also found that the method is capable of creating a balance between computational speed and other performance objectives by varying the market weighting parameters. For example, a distinct tradeoff can be observed between the amount of time required to complete all three tasks and the total amount of data transmitted during the completion of these tasks as the value of γ_B is increased. This is evidence that the market-based methodology proposed herein can effectively and autonomously prioritize between competing computational objectives such as computing speed and power consumption. Similar tradeoff relationships were found between speed and network reliability by varying the parameters α_B and β_B.

4.7 Chapter Summary

This chapter outlines 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 WSN. 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. Additionally, through the use of three weighting parameters (α_B, β_B, and γ_M), this market-based method is shown to be capable of effectively and autonomously shifting network priority from one performance objective to another, thereby offering a flexible framework where scarce resources can be optimally consumed in the midst of competing resource-based objectives.

When combined with the decentralized modal analysis techniques developed in Chapter 2 of this dissertation and the distributed model updating method created in Chapter 3, the market-based resource allocation technique presented in this chapter

represents the completion of a novel agent-based computational paradigm in which sensor data can be autonomously collected from a structure and processed in a multitude of ways without the need for a centralized processing center. In contrast to traditional centralized methods of data processing within WSNs, this agent-based approach is incredibly scalable, power efficient, and robust to communication or sensor failure. While the market-based resource allocation techniques presented herein close the loop on the development of a flexible architecture appropriate for many SHM applications, the ideas developed in this chapter can also be leveraged in order to improve upon the modal estimation techniques of Chapter 2. As such, a novel market-based frequency domain decomposition technique based on this work is presented in detail in Chapter 5.

CHAPTER 5

MARKET-BASED FREQUENCY DOMAIN DECOMPOSITION FOR AUTOMATED MODE SHAPE ESTIMATION IN WIRELESS SENSOR NETWORKS

5.1 Introduction

In this chapter, the market-based resource allocation techniques developed in Chapter 4 are applied to the distributed modal analysis methods developed in Chapter 2. Specifically, these market-based methods are used to overcome the topology-related restrictions of the decentralized FDD method (these restrictions are outlined in Section 2.3.2.3), thereby creating a true agent-based architecture for automated modal estimation in wireless sensor networks. Simultaneously, the resulting method leads to improvements in the quality of the mode shapes estimated by the decentralized FDD technique. Quality mode shape estimates are important in many SHM applications, including computer-aided design (Sinha and Friswell 2002), model-validation (Mottershead and Friswell 1993), and damage detection (Doebling*, et al.* 1998; Ismail*, et al.* 2006; Fang and Perera 2009).

Recent advances in microprocessor and wireless communication hardware have paved the way for the development of automated modal estimation techniques

specifically for use within dense wireless SHM systems (Lynch 2007). While there has been a sizable amount of effort spent creating centralized algorithms for modal estimation within WSNs (Lynch*, et al.* 2004; Lynch*, et al.* 2006), recent work by Zimmerman*, et al.* (2008a) represents the first successful attempt to create a fully decentralized approach to in-network modal identification. In this work (presented in Chapter 2 of this dissertation), three traditionally serial modal estimation methods (peak-picking, random decrement, and frequency domain decomposition) are decentralized and embedded within a network of wireless sensing prototypes. It is shown experimentally that a wireless monitoring system is able to autonomously determine modal frequencies using a distributed peak picking (PP) algorithm, mode shapes using a distributed frequency domain decomposition (FDD) method, and modal damping ratios using a distributed random decrement (RD) technique. It is also seen that these embedded techniques yield modal parameters comparable to those obtained using traditional offline analyses.

However, while these methodologies have proven successful as applied to two separate structural systems (Zimmerman*, et al.* 2008a; Zimmerman*, et al.* 2008b), it is noted that the frequency domain decomposition method for mode shape estimation requires a linear network topology that may result in the accumulation of error within global mode shape estimates. Additionally, because the FDD topology in this approach must be set by the user upon deployment of the sensing system, this method is somewhat constrained in environments where sensor failure or communication loss is a possibility. In this chapter, the market-based resource allocation techniques previously employed for optimizing model updating calculations are leveraged in order to improve mode shape estimation within wireless sensor networks by increasing the robustness of the

decentralized FDD method while simultaneously helping balance the tradeoff between mode shape accuracy and computational resource consumption (*i.e.*, storage capacity, CPU cycles, communication bandwidth, etc) in the wireless domain.

5.2 Mode Shape Estimation using the Decentralized FDD Method

The frequency domain decomposition (FDD) technique, which was developed by Brincker*, et al*. (2001b), improves upon other methods of mode shape estimation, such as peak-picking (Ewins 1986), by allowing closely spaced modes to be identified with great accuracy. This method works by approximately decomposing the spectral density matrix into a set of single degree of freedom (SDOF) systems. In order to accomplish this, an estimate of the output power spectral density (PSD) matrix, $\hat{\boldsymbol{G}}yy(j\omega)$, is first obtained for each discrete frequency $\omega = \omega_i$ by creating an array of frequency response functions (FRFs) using Fast Fourier Transform (FFT) information from each degree of freedom in a system. Then, by taking the singular value decomposition (SVD) of the matrix $\hat{\boldsymbol{G}}yy(j\omega)$, singular values and singular vectors can be extracted from the PSD. If an SVD is performed near a modal peak in the PSD function, the first singular vector, u_{i1}, can be interpreted as an accurate estimate of the mode shape, ϕ_i.

In its serial implementation, the centralized FDD (CFDD) method requires that a processing element have a significant amount of memory in order to store and manipulate the output PSD matrix for each degree of freedom in the system. If there are 100 sensing nodes in a network, for example, the CFDD method requires complex matrix operations (an SVD, in particular), to be performed on a 100x100 PSD matrix. Within a wireless sensing network, where memory availability is scarce and processing power is limited, an

alternative decentralized FDD (DCFDD) method can be used to create independent mode shapes between sensing node pairs; a central node can then be used to combine these two-node mode shapes into global properties after computation is complete (Zimmerman*, et al.* 2008a). In this decentralized approach, each wireless sensor first collects a consistent set of time history acceleration data that is converted to an FRF using an embedded FFT algorithm. Then, an in-network decentralized peak picking algorithm is employed to look for system-wide consensus in identified modal frequencies. Once the entire network is apprised of the global modal frequencies, each node can transmit its individual FFT results at each of these frequencies to the next unit in a pre-determined chain (except the last node in the chain, which has no successor). Using this data, each receiver node can construct a two-degree of freedom output PSD matrix for each picked frequency using the two sets of FFT results in its possession. Then, each receiver node performs a SVD on the resulting 2x2 PSD matrices, extracting a set of two-node mode shapes from the singular values corresponding to each modal frequency. Finally, all of these two-node mode shapes can be sent to a central node where they can be stitched together to form global mode shapes of the structural system.

5.2.1 Limitations of the Decentralized FDD Method

While networks of wireless sensors employing the DCFDD method have been shown to be capable of creating accurate mode shape estimates while monitoring both a theatre balcony (Zimmerman*, et al.* 2008a) and a steel pedestrian bridge (Zimmerman*, et al.* 2008b), there are two major drawbacks to this method as it applies to autonomous in-network execution. The first disadvantage is that all global mode shapes that are

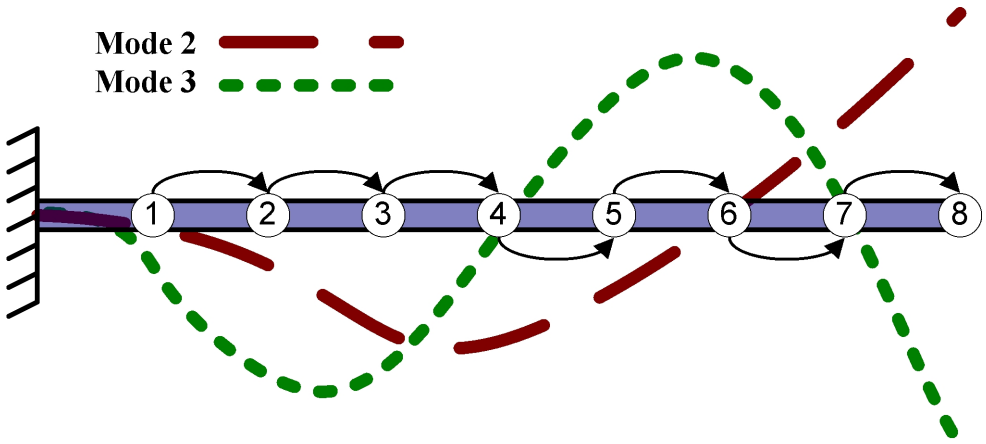


Figure 5.1: An example network topology for two-node FDD data sharing (arrows and shading indicate transmission of Fourier spectra for 2-point mode determination).

determined using the DCFDD method are merely linear combinations of two-node mode shapes calculated locally between each pair of wireless nodes. As such, the network topology (Figure 5.1 shows one possible topology for an 8 node wireless network monitoring a simple cantilevered beam) used to create the DCFDD computational chain can have a potentially large impact on the accuracy of the mode shape estimate. For example, let us assume that one accelerometer in a wireless network is deployed at a "node" of the second bending mode of interest, and that the wireless unit collecting data from this sensor is also somewhere in the middle of the computational chain (*i.e.*, WSU 6 in Figure 5.1). Because of its geographic location (on a modal node), this sensor will have a near-zero FRF value at the natural frequency of the third mode. As such, when global mode shapes are being assembled, this near-zero value will have the potential to propagate through the entire estimated shape, causing numerical instability between the partial mode shape generated by the sensing units above it in the computational chain (*i.e.*, WSUs 1 to 5 in Figure 5.1) and the partial mode shape generated by those below (*i.e.*, WSUs 7 and 8). It is important to note that this effect is largely dependent on the noise floor of the sensors being used. If there is no noise present in the system, the

DCFDD mode shape estimates will match CFDD estimates exactly. However, as the amount of noise in the system increases, the negative impact of the decentralization will increase as well. This is due to the fact that in a centralized implementation, a least-squares effect minimizes the error due to noise across the entire mode shape, whereas a decentralized implementation allows this noise error to accumulate through each combination of multiple 2-node mode shapes.

The second major limitation of the DCFDD method is that a fixed topology must be decided upon before a sensing network can be deployed on a physical structure. As such, DCFDD cannot be applied to a given monitoring scenario without *a priori* knowledge of both the monitored structure and the monitoring system to be deployed. Additionally, this restriction means that the DCFDD method is not robust in situations where wireless nodes can either fail or temporarily lose communication.

5.2.2 Possible Improvements to the Decentralized FDD Method

One obvious way in which the aforementioned limitations of the DCFDD method could be mitigated is by decreasing the degree of computational decentralization inherent to the technique. In other words, instead of forming global mode shapes out of a sequence of two-node mode shapes, we can increase the size of the local mode shapes to three wireless nodes or higher. This type of change could be implemented within the WSN by requiring two or more (say, n-1) wireless nodes to transmit their FRF information to a single node, on which an nxn SVD would be performed (instead of the standard 2x2 SVD proposed in the original DCFDD method). These local n-node mode shapes could then be combined by forcing one or more WSUs in a local mode shape to be redundant to another

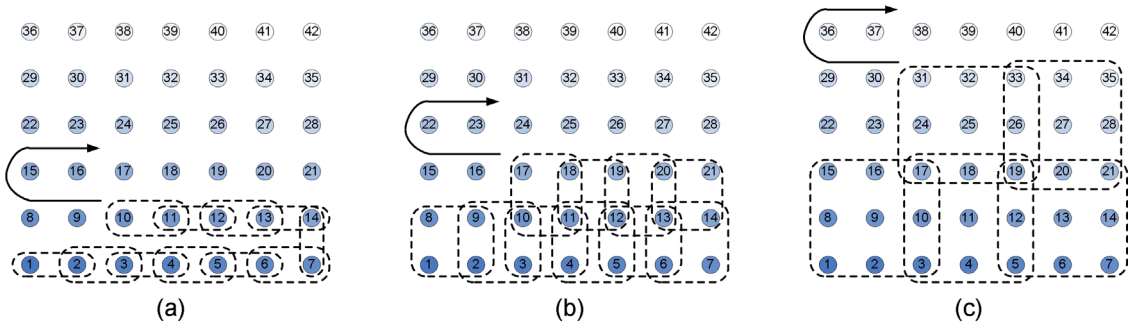


Figure 5.2: (a) DCFDD topology from Zimmerman *et al.* (2008), with two-node mode shapes and (b,c) DCFDD topologies from Sim *et al.* (2009), with overlapping four and nine-node mode shapes, respectively.

local mode shape. Note that if more than one wireless node overlaps between local mode shapes, some higher-level mode-stitching methodology (a least square approach, for example) must be utilized to average out any differences in the global mode shapes resulting from joining local shapes generated relative to each overlapping WSU. From the FDD perspective, this increased computational centrality could improve the quality of decentralized mode shape estimation by eliminating situations like the one described above, where a wireless sensor with near-zero frequency content at a mode of interest creates numerical instability during the mode stitching step. In the improved technique, this type of near-zero sensor data could be buffered between two other nodes in a 3x3 or greater SVD calculation. Then, modes could be stitched using the buffer nodes instead of the node with the non-zero value, leading to a decrease in estimation error.

A similar strategy for improving the DCFDD method was recently proposed and implemented in simulation by Sim*, et al.* (2009). In this work, global mode shapes are created from a set of local mode shapes by leveraging topologies with increasingly large sets of overlapping nodes (Figure 5.2). By minimizing the error between stitched and reference global mode shapes, the authors of this study found that sufficiently large local groups and multiple overlapping nodes contribute to more reliable mode shape estimates

than were possible with the two-node mode shapes utilized in the original DCFDD method (Figure 5.2a). However, this new approach still requires an *a priori* topology assignment, making it fragile in networks prone to communication or sensor failure. Also, this method requires either significantly more communication or significantly more computation than the original method, as a subset of wireless nodes will have to either transmit their data more than once or compute more than one SVD.

It should be noted, however, that adopting *any* strategy that generates local modes of size 3 or more will require an immediate and obvious trade-off relative to the original DCFDD methodology: as the size, n, of the local mode shape estimates increase, the amount of time required to complete the necessary nxn SVD computations will grow exponentially. Figure 5.3 shows experimentally determined SVD computation times for a variety of SVD sizes, as run on an Atmel ATmega128 microprocessor (the same microcontroller used in the *Narada* wireless sensor node). This fixed point, 8-bit microprocessor, which in this case is running on an 8MHz externally generated clock signal, is widely used in the wireless sensing community and is fairly comparable from a computational standpoint to other low-power 8-bit microprocessors available for embedded computing. As can be seen in Figure 5.3, the gathered experimental time, t, can be easily modeled with the following second order regression:

$$t\,(\text{sec}) = \left(2.087x10^{-3}\right)\cdot n^3 - \left(1.60x10^{-3}\right)\cdot n^2 + \left(15.109x10^{-3}\right)\cdot n \qquad (5.1)$$

where n is the size of the SVD decomposition.

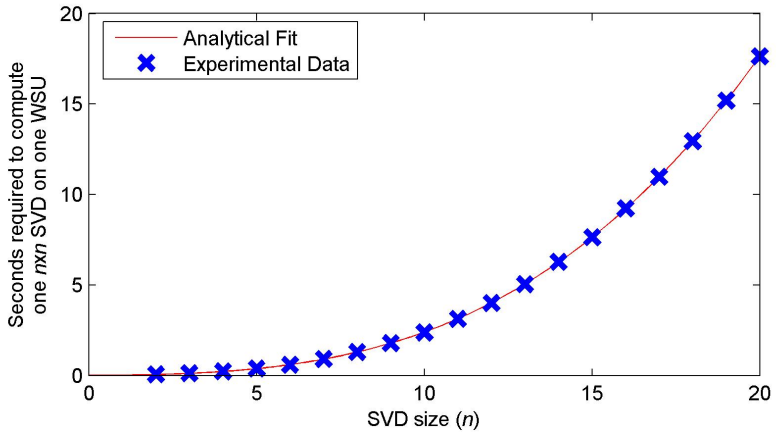


Figure 5.3: Time required for a single wireless sensor (using an 8-bit microcontroller) to complete an *n*x*n* SVD calculation, and the associated second order regression (Equation 5.1).

It is important to note from this figure that while the 17.5 seconds of computation required to complete a 20x20 SVD on the ATmega128 may not seem significant, in some FDD applications there may be hundreds or thousands of frequencies of interest, each one of which requires a separate SVD computation. In these cases, the time saved by further decentralizing the mode shape calculations could be drastic.

It is also important to note that some wireless sensing platforms are designed with access to large amounts of external SRAM for computation. As such, the maximum possible SVD size for certain wireless platforms is far greater than the 20x20 SVD shown in Figure 5.3. The *Narada* wireless sensor (see Section 1.6.2), for example, has 128kB of supplementary SRAM, and is theoretically capable of computing SVDs up to 128x128 in size (each entry in this matrix is a complex single precision floating point value requiring 8 bytes of storage). The Crossbow iMote2 has 256kB of SRAM, and can, in theory, compute an SVD up to 181x181 in size. In this study, however, we will consider a wireless framework utilizing an ATmega128 where no external storage is available,

restricting the maximum SVD size to 20x20 (requiring 3,200 bytes of storage – or about 80% of the total 4kB of available SRAM).

5.3 Background on Market-Based Resource Allocation

As outlined above, it is clear that improvements to the DCFDD method can be made by exploring the effects of fewer (and larger) local mode shape estimates. However, it has also been shown that there is a distinct tradeoff between improved mode shape estimates and the amount of scarce resources (specifically, computing time, storage space, *etc.*) required to calculate local mode shapes using increasingly large clusters of wireless sensing units. As such, the objective of the work in this chapter is to develop a robust methodology in which this tradeoff between mode shape accuracy and computational requirements can be optimally managed in an autonomous and ad-hoc manner.

While there are several possible approaches to this optimization problem, it is decided to leverage the workings of another complex system that is optimally controlled in a decentralized manner: the free-market economy. This free-market approach was also utilized for resource allocation in Chapter 4. In a free market economy, scarce societal resources are distributed based on the local interactions of buyers and sellers who obey the laws of supply and demand. In the context of this chapter, wireless nodes can be modeled as buyers and sellers who are looking to trade an optimal amount of scarce system resources (in this case, storage space and processing time) in exchange for a measure of gained utility (in this case, improved mode shape estimates). Having described the DCFDD method and its potential limitations, it can now be seen that

market-based methods may be capable of helping manage the conflicts created between mode shape accuracy and the limited computational resources available within a WSN.

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 (Clearwater 1996). Perhaps the greatest benefit of market-based optimization is that it can often yield a Pareto optimal solution; a Pareto optimal market is one in which no market participant can reap the benefits of higher utility or profits without causing harm to other participants when a resource allocation is changed (Mas-Colell*, et al.* 1995).

Recently, Zimmerman*, et al.* (2009) developed a WSN-oriented resource distribution framework based on free-market economics that can be used to autonomously allocate scarce system resources (such as battery power, data storage capacity, CPU time, wireless bandwidth, *etc.*) for the simultaneous processing of multiple computational tasks within a WSN. This architecture is explained in great detail in Chapter 4 of this dissertation, and is leveraged to improve the decentralized mode shape estimation method.

5.3.1 Market-Based Resource Allocation in WSNs

Conceptually, it would be somewhat trivial to develop a simple market-based system which could be used to crudely optimize a WSN-based resource allocation system based on only one goal, such as maximizing computational speed. However, it is significantly more valuable to consider a more robust market-based scheme that can

optimally allocate resources in the midst of several additional competing objectives, such as computational speed, wireless bandwidth usage and battery consumption. Zimmerman*, et al.* (2009) create this type of system through the use of buyer and seller "utilities". By embedding within each market agent (*i.e.*, wireless sensor) the desire to maximize an individual utility function, it has been shown that competing goals can be settled through market means (supply and demand equilibrium). The result also represents a Pareto optimal allocation of scare system resources.

In previous studies, focus is placed on three distinct (but possibly competing) performance objectives: completing all required computational tasks as quickly as possible, minimizing power consumed by the sensor network, and functioning as robustly and as reliably as possible. In order to measure the ability of the market-based technique to address these three objectives, three performance metrics are created and utilized: the time required to complete a given task, the number of wireless transmissions required to complete a given task, and the number of sensor and communication failures encountered during a given task.

In the market-based resource allocation method referenced above (Zimmerman*, et al.* 2009), market sellers are defined as the set of sensors in the wireless network not currently working on any computational task. These sensing units "sell" their computational abilities to a number of buyers, represented by the set of sensors most recently added to each existing computational task. In order to simultaneously address all three performance objectives in a streamlined manner, buyers and sellers focus on different goals. In this market, sellers work to minimize network power consumption and buyers work to minimize the overall time spent computing while simultaneously

maximizing network reliability. In order to quantify the degree to which a given resource allocation benefits the network as a whole, separate utility functions (U_B and U_S) are assigned to buyers and sellers, respectively.

Having developed utility functions associated with both buyers and sellers, a methodology is created 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:

1. All sensing units not currently computing will broadcast their availability to the network (as market sellers).
2. The wireless sensors having most recently joined each existing computational task (market buyers) will calculate U_B based on the computational task they are working on, and place a bid of U_B if $U_B > 0$.
3. Market sellers will calculate U_S based on each proposed computational job offer they receive.
4. Once all bids have been received, market sellers will calculate their expected profit from each proposed job using a market power / speed exchange rate (γ_M) that represents the minimum number of seconds of computational speedup that must be gained in order to warrant an additional byte of communication:

$$profit = U_B - \gamma_M \cdot U_S \tag{5.2}$$

5. Market sellers will choose the bid that generates the greatest non-negative profit, and join the corresponding computational task.

It is shown that this market-based method of optimally allocating scarce system resources within a WSN 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 task at the outset of computation. Additionally, it is found that the utility functions developed in this market-based method allow the network to shift priority from one scarce resource to another and from one computational objective to another, providing a flexible framework where scarce resources can be optimally consumed in the midst of competing resource-based objectives and constraints. Given the effectiveness of the proposed market-based methods, they will be explored to improve the accuracy and robustness of the DCFDD mode shape estimation method.

5.4 Market-Based Frequency Domain Decomposition in WSNs

In order to create an environment in which the WSN-based DCFDD method can improve its mode shape estimation capability while optimizing its required consumption of scarce resources, the market-based buyer-seller framework outlined in Section 5.3 is applied to the modal estimation problem. In this problem, we are interested in optimizing over four distinct (but possibly competing) performance objectives: (O_1) estimating mode shapes as accurately as possible from dynamic sensor data; (O_2) calculating mode shape estimates as quickly as possible; (O_3) utilizing as little memory as possible; (O_4) maintaining communications that are as reliable as possible. In order to measure the ability of the market-based DCFDD to address these objectives, four performance metrics are created and utilized: (M_1) the accuracy of each estimated mode shape; (M_2) the amount of time required to calculate each estimated mode shape; (M_3) the amount of

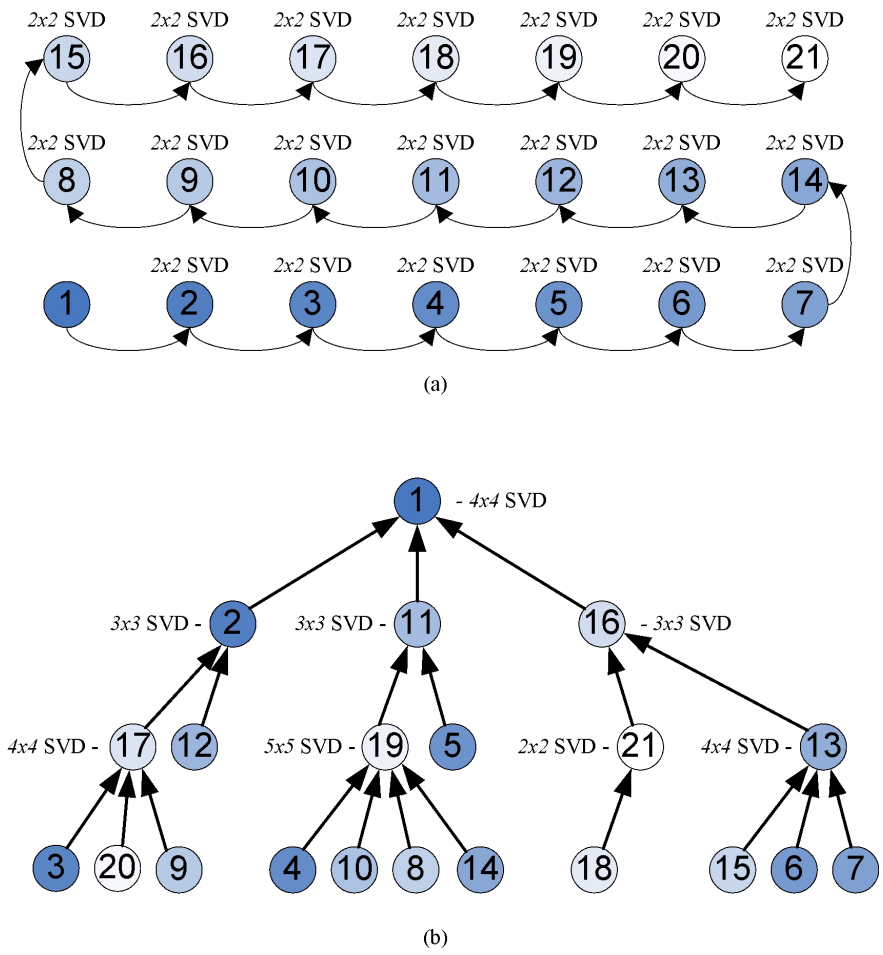


Figure 5.4: (a) Example DCFDD network topology and computational requirements vs. (b) Example MBFDD network topology and computational requirements.

storage per computing sensor required to calculate each estimated mode shape; (M_4) the strength the communication links created between buyer and seller nodes.

In contrast to the DCFDD method, which functions by creating a used-defined chain-like topology through which computational tasks (specifically, 2x2 SVD calculations) can be distributed amongst a network of wireless sensors (Figure 5.4a), the Market-Based Frequency Domain Decomposition (MBFDD) technique creates an ad-hoc tree-like topology through which a set of *n*x*n* SVD calculations can be similarly distributed (Figure 5.4b). This ad-hoc approach has numerous advantages over the chain-like DCFDD topology formations presented by Zimmerman*, et al.* (2008a). First, by

expanding the potential size of each local mode shape (*i.e.,* SVD dimension), problems associated with stitching together many two-node modes can be greatly mitigated and the accuracy of the global mode shape estimation will improve. Second, this type of optimal ad-hoc tree creation is dependent only on the ability to analytically model the monitored structure before sensors are deployed. As such, the MBFDD method can create an optimal tree for mode shape estimation even in the midst of unknown sensor placement, sensor failure, network communication loss, *etc.*

5.4.1 Buyer/Seller Framework for MBFDD

In the MBFDD method developed in this study, wireless sensors available for computation will either be required to transmit their frequency domain data (at estimated modal frequencies) to another wireless sensor (*i.e.,* WSU 18 in Figure 5.4b), or receive frequency domain data (at estimated modal frequencies) from *n*-1 additional wireless sensors and compute an *n*x*n* SVD (*i.e.,* WSU 1 in Figure 5.4b), or (c) transmit, receive, *and* compute (*i.e.,* WSU 11 in Figure 5.4b). The purpose of the MBFDD technique, then, is to create an optimal topology for sending, receiving, and computing dynamic sensor data so as to optimize between objectives O_1, O_2, O_3, and O_4.

In contrast to the market-based resource allocation methodology discussed in Section 5.3 (Zimmerman*, et al.* 2009), market sellers in the MBFDD method can be defined as the set of sensors in the wireless network not currently assigned to an action (*i.e.*, send, receive-compute, or both). In a way, these WSUs will be "selling" their sensor data to one of a number of buyers. Buyers in this market are represented by the set of sensors currently assigned to either send data, receive-compute, or both.

In order to simultaneously address all four performance objectives (O_1, O_2, O_3, and O_4) in a streamlined manner, buyers and sellers focus on different goals. In this market, buyers work to maximize mode shape accuracy (O_1), to minimize the overall time spent computing (O_2), and to minimize the amount of required storage (O_3). As such, buyers gain utility by creating a topology that minimizes the difference between the MBFDD and CFDD mode shape estimations, as calculated using the modal assurance criteria (MAC) (Allemang and Brown 1982), and by limiting the size of the local SVD clusters in the MBFDD computational tree, (since smaller SVD clusters require less computation and less storage capacity). Sellers, on the other hand, work to improve the reliability of the wireless connections in the MBFDD tree structure (O_4). Thus, sellers gain utility by forming quality (*i.e.*, reliable) communication links between nodes.

5.4.2 Formulation of Buyer-Side Utility Functions for MBFDD

In light of this framework, it is now necessary to explicitly derive utility functions associated with both buyers and sellers as they create a tree-like topology for MBFDD mode shape estimation. These utility functions will govern which computational SVD cluster an unassigned sensor will join, and will limit the size of these clusters in order to conserve processor time and memory consumption. On the buyer side, the utility, U_B, gained by adding an additional wireless sensor to a given computational cluster represents a weighted combination of the expected improvement in MAC value, M_B, increase in computational time, T_B, and increase in storage capacity, S_B, brought about by a move from SVD cluster size n-1 to SVD cluster size n. As such, U_B can be defined as follows:

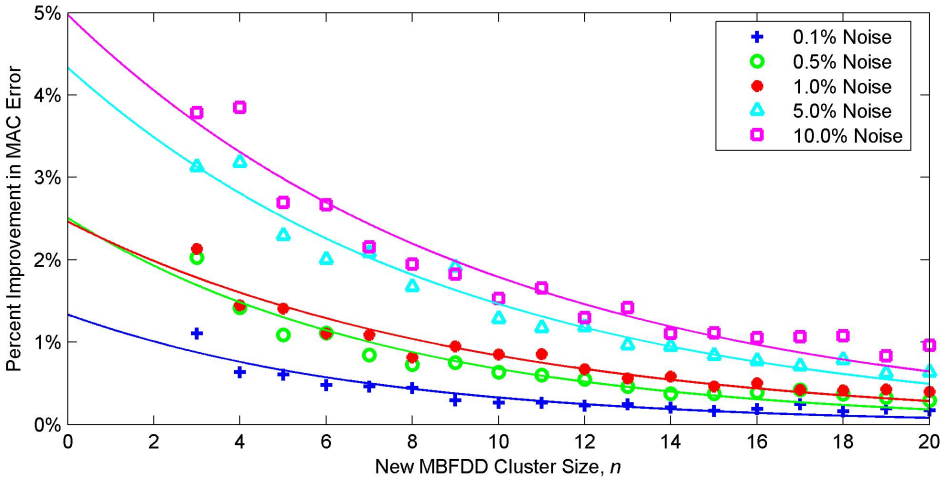


Figure 5.5: Percent improvement in MAC error brought about by increasing computational cluster size, for sensor data with varying noise levels.

$$U_B = M_B - \frac{1}{\alpha_B} \cdot T_B - \frac{1}{\beta_B} \cdot S_B \tag{5.3}$$

where α_B, and β_B are weighting parameters that allow for a shift of focus between mode shape accuracy, processing time, and storage capacity.

In any FDD calculation, M_B can be thought of as the expected improvement in MAC value brought about by a move from cluster size n-1 to cluster size n. While it is very difficult to directly formulate an analytical expression for this value, a trend can be established by looking at the average improvement (decrease) in MAC value over a large number of experimental trials where one WSU is moved from a 2-node cluster (we will call this Cluster A) into an existing n-1 node cluster (Cluster B), thus creating an n node cluster (Cluster C). Using a simple analytical model of a cantilevered beam to generated simulated experimental data, Figure 5.5 shows the experimentally generated average improvement in MAC value error gained by estimating a mode shape directly using an

nxn SVD (from Cluster C) rather than stitching together mode shapes estimated using both an (*n*-1)x(*n*-1) SVD (from Cluster B) and a 2x2 SVD (from Cluster A). Since the value of M_B is an application-specific quantity (*i.e.,* some structures and/or sensor configurations may show greater sensitivity to decentralized mode shape estimates), a simulated cantilever beam testbed (described in greater detail in Section 5.5) on which 20 wireless sensors are deployed is leveraged to create this figure. Specifically, Figure 5.5 utilizes data gathered over a large number of experimental trials run using computational characteristics measured from the *Narada* WSU platform (see Section 1.6.2) and simulated vibration data with noise levels varying between 0.1% RMS and 10% RMS. It is found that each of the curves in this figure (representing varying noise levels) can be modeled by an easily computable algebraic function, which represents the average improvement (decrease) in MAC value, M_B:

$$M_B(A,\lambda,n) = A \cdot e^{-\lambda \cdot n} \tag{5.4}$$

where the values for A and λ are specific to each noise level, and are tabulated in Table 5.1. These analytical regressions can also be seen in Figure 5.5. Note that the benefit of increasing cluster size is experimentally determined to be independent of the mode of interest. As such, Figure 5.5 is an average over the first four modes of the simulated cantilever beam used to develop the regressions for Equation 5.4.

Just like M_B represents the expected improvement in mode shape estimate, the value of T_B represents the expected increase in processing time required to compute an *n*x*n* SVD instead of an (*n*-1)x(*n*-1) SVD. Much like M_B, it is very difficult to directly

Table 5.1: A and λ regression parameters for MAC improvement at varying noise levels.

Noise Level	A	λ
0.0%	0.000E+00	0.000E+00
0.1%	1.334E-02	1.406E-01
0.5%	2.505E-02	1.311E-01
1.0%	2.463E-02	1.079E-01
5.0%	4.333E-02	1.087E-01
10.0%	4.972E-02	1.022E-01
20.0%	4.367E-02	7.244E-02

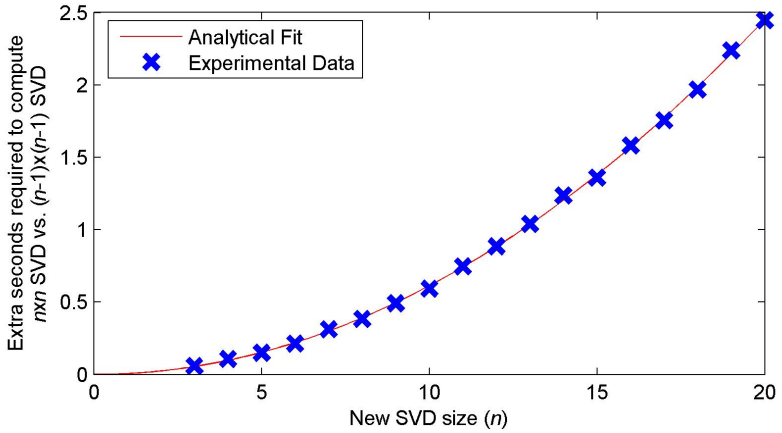


Figure 5.6: Extra time required for a single wireless sensor to complete an nxn SVD calculation instead of an (n-1)x(n-1) SVD, and the associated second order regression (Equation 5.5).

formulating an analytical expression for this value, but an empirical trend can be established by looking at the average amount of time required to complete an SVD of a given size on a given microprocessor. Figure 5.6 shows average experimental values collected using an ATmega128 processor running SVD calculations of various sizes, as well as a second order regression which can be used to easily model T_B:

$$T_B(n) = \left(6.14x10^{-3}\right)\cdot n^2 - \left(0.14x10^{-3}\right)\cdot n \tag{5.5}$$

Correlated tightly to T_B, the value of S_B in any FDD calculation represents the increased number of bytes of storage required to compute an *n*x*n* SVD instead of an (*n*-1)x(*n*-1) SVD. Unlike M_B and T_B, this value is deterministic, as any *n*x*n* SVD computation requires storage of an *n*x*n* matrix of complex single precision floating point values. Both the real and imaginary components of a complex single precision floating point value require 4 bytes of storage. As such, we can easily model S_B as:

$$S_B(n) = 8 \cdot n^2 \tag{5.6}$$

5.4.3 Formulation of Seller-Side Utility Functions for MBFDD

On the seller side of this market-based allocation procedure, a somewhat simpler utility function, U_S, can be developed in a similar fashion to U_B. Intuitively, seller utility can be thought of as the reliability of the wireless communication link required to transmit frequency domain data from seller to buyer. Since the majority of power consumption in a wireless sensing device comes from the wireless radio (which consumes significantly more power than a microcontroller), the seller-side utility is also tied to both minimization of wireless bandwidth as well as minimization of power expenditure (communication links with lower reliability will require increased amounts of retransmission). As such, U_S can be defined as follows:

$$U_S = \gamma_S \cdot C_S \tag{5.7}$$

where γ_S is a weighting parameter allowing the MBFDD algorithm to scale the influence of wireless communication, and C_S represents the probability of failure of a given buyer-seller communication link.

Unlike M_B and T_B, it is possible to model C_S using an analytical expression. This expression is dependent on two parameters which are correlated to the specific wireless platform used in the WSN: the radio signal strength indicator (RSSI), which conveys the strength of a given wireless connection, and the probability of a communication link with a perfect RSSI failing due to unforeseen circumstances, p_{CF}. Within the MBFDD framework, the RSSI parameter can be gathered directly from the radio interface for a given buyer-seller connection, while p_{CF} is a platform specific quantity that is usually quite high (it is taken as 0.9 in this study). As such, C_S can be defined as follows:

$$C_S\left(p_{CS}, RSSI\right) = 1 - \frac{p_{CS}}{1 + e^{-0.4\cdot\left(40.0 + RSSI\right)}} \tag{5.8}$$

It is important to note that while this study focuses only on processing time, storage capacity, and communication reliability, other limited WSN resources (such as battery power, wireless bandwidth, etc), can be similarly included in the utility function U_S, and can be coupled with a new set of weighting parameters. As such, the proposed MBFDD algorithm can create computational trees for mode shape estimation while optimizing over any number of scarce network resources.

5.4.4 MBFDD Algorithm

Having developed utility functions associated with both buyers and sellers, it is now possible to create a methodology with which sensors in a WSN can buy and sell processing time and storage space in order to create optimal mode shape estimates using the MBFDD method. By expanding on the fundamental principles of an auction, the following procedure for MBFDD is developed for implementation within a sensing network:

1. The MBFDD algorithm is initialized by assigning the generic FDD task (at a subset of chosen frequencies) to one available wireless sensing unit (chosen at random). This WSU becomes the root of the MBFDD tree. An example MBFDD tree and its resulting buyer/seller delineation for a small WSN can be seen in Figure 5.7a. Note that in this figure, two nodes (WSU 5 and WSU 4) have already been assigned to the root (WSU 2). This allows all steps of the MBFDD algorithm to be expressed clearly in Figure 5.7.
2. Each WSU already assigned to a position in the MBFDD tree (*i.e.*, each buyer) broadcasts its current cluster size, n-1, to whichever sellers are within communication range. These buyer broadcasts occur one at a time, traversing the MBFDD tree in a depth-first fashion starting with the root. In this way, each seller has a chance to hear bids from each buyer within their communication range. Figure 5.7b demonstrates this process, with sequential broadcasts from WSU 2, WSU 5, and WSU 4. Note that no wireless node is capable of communication with all other nodes in the network.

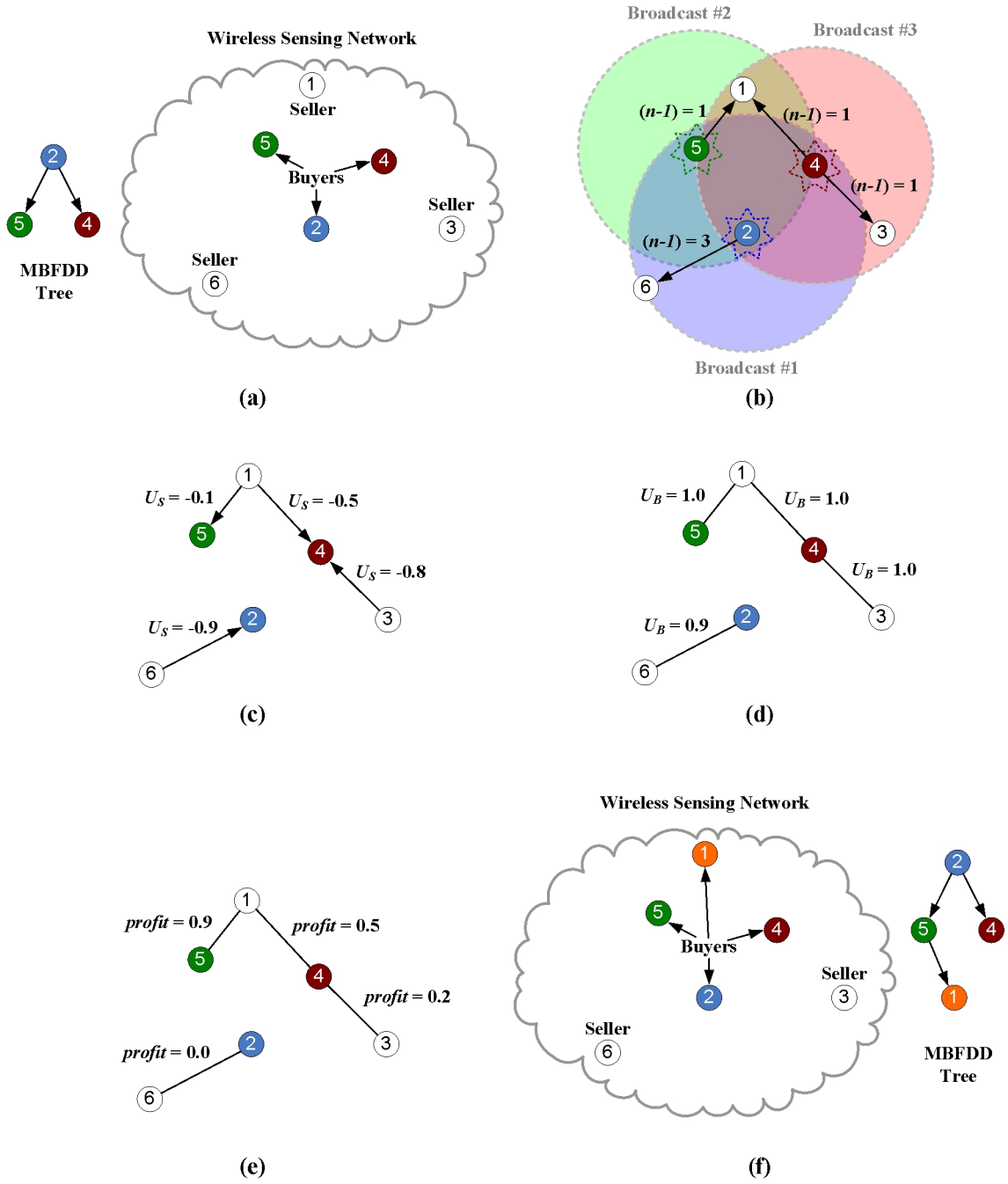


Figure 5.7: MBFDD algorithm: (a) Example MBFDD tree (mid-creation) and buyer/seller delineation. (b) Buyer broadcast of current SVD cluster size. (c) Seller utility determination. (d) Buyer utility determination. (e) Total market utility (profit). (f) Updated MBFDD tree and buyer/seller delineation.

3. Upon receiving a bid from a buyer, each seller will calculate the utility, U_S, that it would gain by accepting the buyer's bid. It will then transmit this utility

information to the buyer, using a randomized backoff to avoid packet collision. Example seller utilities for each buyer-seller combination can be found in Figure 5.7c. Note a correlation between U_S and communication distance.

4. After a buyer node broadcasts a bid request (step 2), it will wait for a period of time to ensure that all utility values, U_S, have arrived from the sellers within its communication range. A buyer will then calculate the utility, U_B, it will gain from adding an additional processor to its SVD cluster. Example buyer utilities for each buyer-seller combination can be found in Figure 5.7d. Note that buyer utilities are correlated to the size of the buyer's SVD cluster.
5. For each seller utility it receives, a buyer will then calculate the expected total market utility (or profit) gained from moving forward with that buyer-seller relationship:

$$profit = U_B + U_S \tag{5.9}$$

Figure 5.7e shows the profit calculated from U_S and U_B. Note that the highest utility is generated between WSU 5 and WSU 1.

6. After all bids have been received, the buyer will determine the bid that generates the greatest profit, and will pass that bid information (including buyer identification number, seller identification number, and total profit) to the next buyer in the depth-first traversal of the MBFDD tree. If, however, the maximum profit generated by a given buyer is less than the maximum profit generated by a previous buyer, the given buyer will pass along the bid information relating to

maximum profit generation instead of its own. In this way, information regarding the buyer-seller combination that generates the greatest profit at a given time step will propagate through the MBFDD tree, eventually ending up back at the root node.

7. After all buyers have completed the broadcast/bid process, the root node will command the buyer involved in the maximum bid (using the MBFDD tree structure for communication) to add the seller node associated with the maximum bid to its computational cluster. This step is visualized in Figure 5.7f, where the maximum utility pair of nodes (WSU 5 and WSU 1) are paired in the tree.
8. Steps 2 through 7 are repeated until no unassigned (seller) nodes remain in the network. At this point, the topology is complete and FDD computation can begin.

Using this algorithm, a MBFDD computational tree can be created such that the overall utility of the market is maximized. Because of the addition of the weighting parameters, α_B, β_B, and γ_M, the resulting framework is capable of optimally adapting, in real-time, to shifting computing needs or resource limitations within a wireless network. For example, assume that it is absolutely essential that a particular mode shape estimate be as accurate as possible. Without any reprogramming of the sensing network, the network can simply assign near-zero values to α_B, β_B, and γ_B in order to reflect the added emphasis on improving computational accuracy. Similarly, increasing values of α_B can be used to emphasize computational speed, β_B to stress storage restrictions, and γ_B to stress communication reliability.

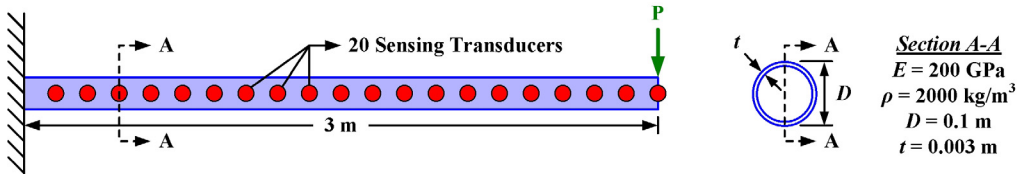


Figure 5.8: Simple cantilevered beam used for experimental validation of the MBFDD method.

5.5 Experimental MBFDD Testbed and Results

In order to validate the market-based task assignment methodology proposed in this study, the four performance metrics (M_1 through M_4) outlined in Section 5.4 are evaluated using a simple cantilever beam testbed. This beam is 3m long, and has a circular cross-section with the properties outlined in Figure 5.8. It is monitored by twenty *Narada* wireless sensors (see Section 1.6.2), each monitoring vertical acceleration and positioned at equal spacing across the length of the beam. This beam can be excited with a broadband input by impulsing it at its tip, and it yields its first four vibrational modes at 2.93Hz, 18.37Hz, 51.44Hz, and 100.81Hz, respectively. However, because it is necessary to examine the performance of the MBFDD over a statistically large number of runs and under a variety of different sensor noise conditions (0.0%, 0.1%, 0.5%, 1.0%, 5.0%, 10.0%, and 20.0% RMS noise) it is decided to utilize a simulated environment to validate the abilities of the MBFDD method.

5.5.1 Evaluation – Mode Shape Accuracy vs. Computation Time

In the first experimental instance performed in this study, the tuning parameter α_B is examined for its ability to shift the priority of the MBFDD methodology from computational speed (at low values of α_B) to mode shape accuracy (at high values of α_B). In other words, the ability of the MBFDD method to optimize between objective O_1 (high

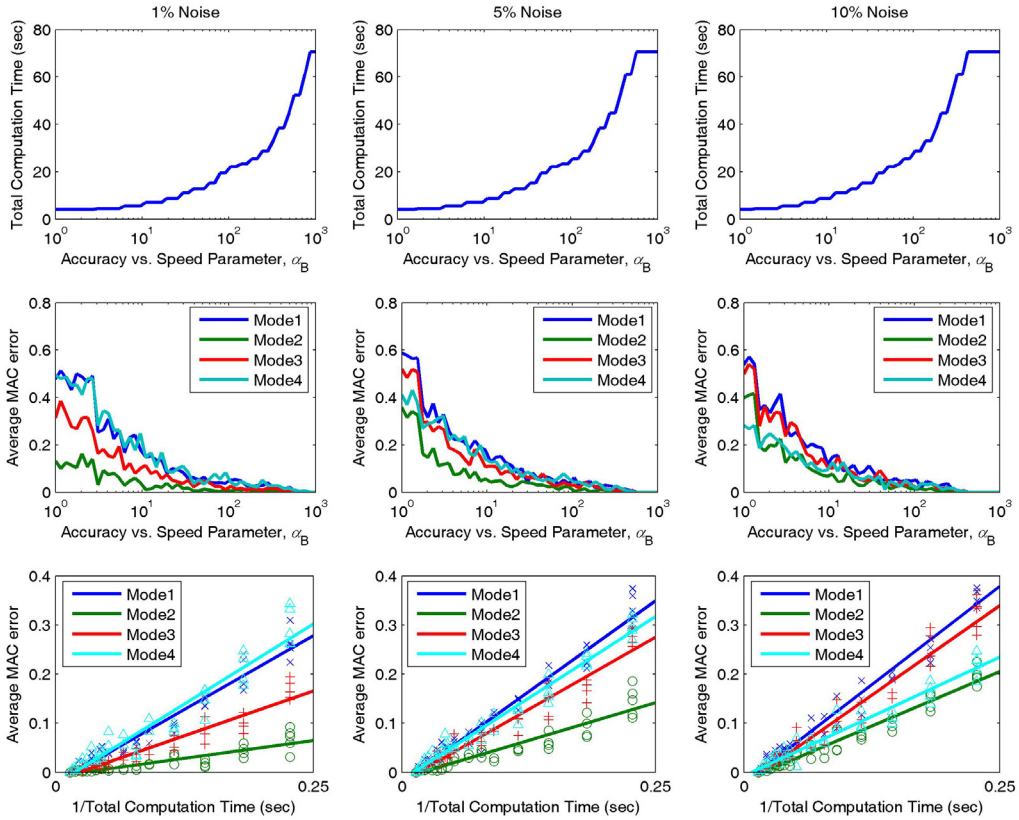


Figure 5.9: Total computation time and average MAC error for the MBFDD method applied to the cantilever testbed for three different levels of noise and with increasing values of α_B.

mode shape accuracy) and objective O_2 (low computation time) is validated. In order to achieve this goal, the network of 20 wireless sensors are repeatedly asked to estimate the first four vibrational modes using the MBFDD method with varying values of α_B. For each of the 7 noise levels of interest (0.0%, 0.1%, 0.5%, 1.0%, 5.0%, 10.0%, and 20.0% RMS noise), 50 different values of α_B (varying between 0 and 1000) are investigated using 100 different sets of sensor data. Figure 5.9 shows the results gathered from this large number of simulated experiments.

While it is fairly easy to quantify the amount of computation required in a given MBFDD topology (simply determine how much time was spent processing), it is less

obvious how to quantify the quality of a mode shape estimate. In this study, MBFDD estimates are compared with CFDD estimates in the following way: for each set of sensor data, a set of mode shapes are estimated using both the MBFDD and CFDD methods. Then, each of these sets of mode shape estimates are compared to analytically derived mode shapes for the cantilever beam using the corresponding MAC value. By taking the difference between the average MBFDD MAC value and the average CFDD MAC value for a given mode shape and noise level, we can generate the aforementioned MAC error between the MBFDD estimates and the CFDD estimates.

As can be seen in Figure 5.9, not only is the MBFDD method capable of creating a very accurate mode shape estimate, but it can effectively utilize the tuning parameter α_B to discriminate between an emphasis on mode shape accuracy and an emphasis on computational speed. The exact relationship between mode shape accuracy and computational speed can be seen in the bottom plots in this figure, where the value of α_B increases from the bottom left to the top right of the plot.

5.5.2 Evaluation – Mode Shape Accuracy vs. Storage Requirement

In the second experimental instance performed in this study, the tuning parameter β_B is examined for its ability to shift the priority of the MBFDD methodology from low storage requirements (at low values of β_B) to mode shape accuracy (at high values of β_B). In other words, the ability of the MBFDD method to optimize between objective O_1 (high mode shape accuracy) and objective O_3 (low storage requirements) is validated. In order to achieve this goal, the network of 20 wireless sensors are repeatedly asked to estimate the first four vibrational modes using the MBFDD method with varying values of β_B. For

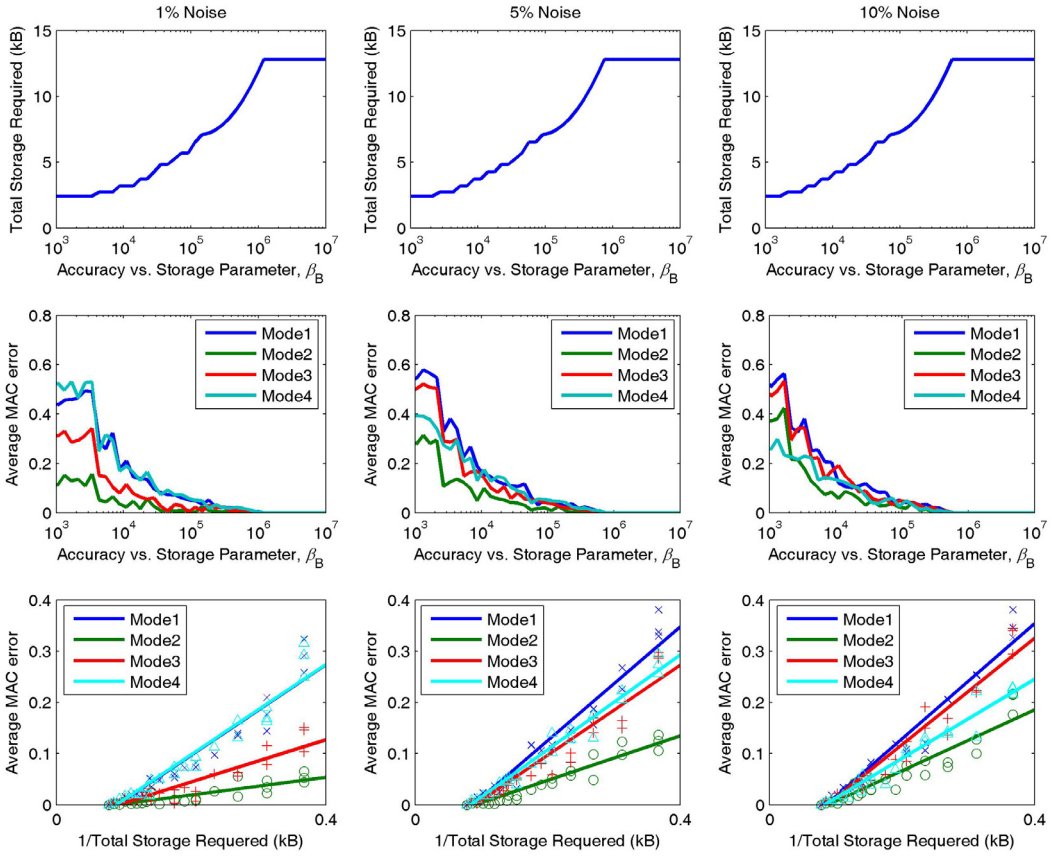


Figure 5.10: Total storage requirement and average MAC error for the MBFDD method applied to the cantilever testbed for three different levels of noise and with increasing values of β_B.

each of the 7 noise levels of interest (0.0%, 0.1%, 0.5%, 1.0%, 5.0%, 10.0%, and 20.0% RMS noise), 50 different values of β_B (varying between 0 and 1000) are investigated using 100 different sets of sensor data. Figure 5.10 shows the results gathered from this large number of simulated experiments. It can be seen from this figure that the MBFDD method is capable of creating very accurate mode shape estimates, and can effectively utilize the tuning parameter β_B to discriminate between an emphasis on mode shape accuracy and an emphasis on required data storage. The exact relationship between mode shape accuracy and required data storage can be seen in the bottom plots in this figure, where the value of β_B increases from the bottom left to the top right of the plot.

5.5.3 Evaluation – Mode Shape Accuracy vs. Communication Reliability

In the third experimental instance performed in this study, the ability of tuning parameter γ_B to shift the priority of the MBFDD methodology from communication quality (at low values of γ_B) to mode shape accuracy (at high values of γ_B) is investigated. In other words, it is necessary to validate the ability of the MBFDD method to optimize between objective O_1 (high mode shape accuracy) and objective O_4 (low communication requirements). In order to achieve this goal, a network of 20 wireless sensors are repeatedly asked to estimate the first four vibrational modes using the MBFDD method with varying values of γ_B. For each of the 7 noise levels of interest (0.0%, 0.1%, 0.5%, 1.0%, 5.0%, 10.0%, and 20.0% RMS noise), 50 different values of γ_B (varying between 0.1 and 1000) are investigated using 100 different sets of sensor data. Figure 5.11 shows the results gathered from this large number of simulated experiments.

In this investigation, the effect of communication quality on the MBFDD topology is investigated by looking at the average (geometric) distance between communication links in the MBFDD tree. Because RSSI is correlated to distance between transmitter and receiver (assuming line-of-sight), it follows that sensors that are farther from one another on the cantilever will have lower mutual RSSI values, indicated decreased communication reliability. It can be seen that this correlation holds in the experimental setting. Mode shape accuracy is determined exactly as described in Section 5.4.1.

As seen in Figure 5.11, the MBFDD method is capable of discriminating between an emphasis on mode shape accuracy and an emphasis on communication reliability by utilizing the tuning parameter γ_B. The exact relationship between mode shape accuracy

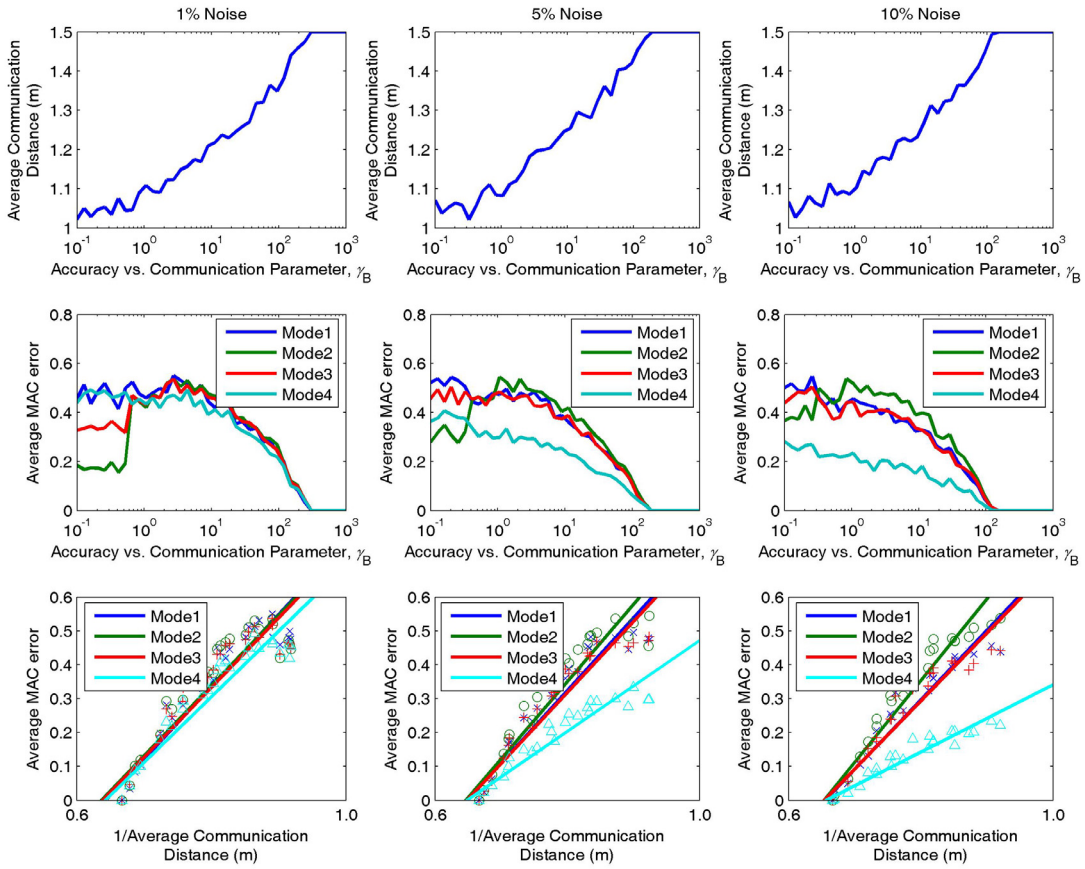


Figure 5.11: Average communication distance and average MAC error for the MBFDD method applied to the cantilever testbed for three different levels of noise and with increasing values of γ_B.

and communication reliability can be seen in the bottom plots in this figure, where the value of γ_B increases from the bottom left to the top right of the plot. It is interesting to note that at values of γ_B less than 1.0, we actually see a decrease in mode shape error. This is because at γ_B values less than 1.0, the MBFDD algorithm is approaching the DCFDD topology (*i.e.,* the MBFDD tree is merely a sequential chain of sensors running the length of the cantilever). Because of the simplicity of this beam example, this sequential ordering actually happens to produce improved results at low noise levels for certain mode shapes. However, this phenomena would not extend to more complex examples.

5.6 Chapter Summary

In this chapter, the decentralized frequency domain decomposition (DCFDD) method for autonomous mode shape estimation presented in Chapter 2 is drastically improved in two distinct ways. First, by creating a tree-like topology in place of the chain-like topology presented in Chapter 2, the resulting mode shape estimates can be calculated by stitching together local mode shapes of varying size, instead of focusing solely on the two-node mode shapes of Chapter 2. In addition (and perhaps more importantly), the flexibility limitations of the original DCFDD algorithm (with respect to topology formation and distribution of scarce network resources) are greatly mitigated through the application of market-based resource allocation techniques. Specifically, the framework developed in Chapter 4 for market-based resource allocation in WSNs is leveraged in order to provide an agent-based architecture for autonomously estimating mode shapes using DCFDD.

The resulting market-based frequency domain decomposition methodology (MBFDD) is successfully applied to mode shape estimation in a simple cantilever beam. It is shown that the MBFDD method is capable of autonomously forming a computational topology that allows a network of wireless sensors to not only improve on the mode shape estimates of the DCFDD technique, but to optimally distinguish between multiple resource constraints or objectives. Using the weighting parameters α_B, β_B, and γ_B, the MBFDD method is experimentally shown to be capable of managing changing emphasis between mode shape accuracy, computational speed, storage requirements, and wireless communication reliability.

CHAPTER 6

CONCLUSIONS

6.1 Summary of Results and Contributions

As sensing technologies and data processing techniques continue to improve in capability and decrease in price, the modern engineering community is becoming increasingly reliant on sensor data to provide an accurate assessment of structural system behavior and performance. Experimentally sensed data is vital to properly validating and calibrating analytical models, as well as detecting degradation and failure in engineered systems. Due to the high costs associated with the installation and maintenance of coaxial data cables in large engineered systems, wireless sensing technologies have recently been explored as a new interface between sensing transducer and data repository. By lowering installation and maintenance costs, wireless sensors have provided a cost-effective building block on which pervasive networks of sensing transducers can be deployed on large civil structures, such as bridges or buildings.

In addition to the cost savings generated by the elimination of unnecessary cables, wireless sensing networks (WSNs) have also shown great promise because of their ability to process sensor data locally at each wireless node. Local data processing is especially advantageous when confronted with the huge amounts of data commonly associated with

dense networks of sensors (such as those envisioned to be created using low-cost wireless technology). As such, different architectures have been developed for embedded data processing using wireless sensors. Early work in this area focused primarily on serial implementations of engineering algorithms, which allow a WSN to process sensor data locally, thereby minimizing the amount of information needing to be transmitted to a central data repository. These centralized embedded data processing methods are relatively power efficient when compared to the transfer of raw time history data to a central location, but do not share sensor data between wireless nodes, preventing these architectures from autonomously determining system-wide properties.

This dissertation focuses on the development of a set of distributed computing architectures for processing sensor data within a network of wireless sensors. By focusing on the ad-hoc capabilities of a WSN (characterized by autonomously created and self-healing communication links between sensors), several novel agent-based computational architectures for distributed in-network data processing are presented and evaluated in the context of structural health monitoring (SHM). These architectures allow dense wireless monitoring systems to collect, store, and autonomously process large amounts of sensor data. Data processing in-network eliminates some common monitoring problems such as power consumption (for battery operated nodes), and data glut. Each successive architecture developed herein moves farther away from the current reliance on a centralized architecture for in-network computing and towards an agent-based paradigm where network computing demands can be handled autonomously (and Pareto-optimally) without the need for human interaction.

In Chapter 2, a set of distributed computing techniques are developed for automated in-network estimation of modal parameters (modal frequencies using the peak picking approach, damping ratios using the random decrement method, and mode shapes using the frequency domain decomposition algorithm) given a set of wirelessly collected sensor data. These decentralized algorithms, which treat each wireless sensor as an equal-capability processor within a larger parallel computing system, represent the first fully parallel computing architectures applied to the wireless sensing paradigm. This set of distributed techniques is evaluated by deploying a network of wireless sensors on both the main balcony of a large theatre located in Detroit, Michigan and on a pedestrian bridge located in Ann Arbor, Michigan. In each case, the network of deployed sensors is shown to be capable of automatically and accurately estimating modal properties of a real world structure. While these distributed methodologies demonstrate the first fully parallel data processing architecture implemented within a WSN, and are shown to be capable of accurately estimating modal properties, they do not take full advantage of the distributed intelligence available within a network of wireless sensors. Specifically, because they rely on a predetermined network topology, these methods neglect the reliability-related advantages that come from autonomous network creation and self-healing, making them somewhat unreliable in the wake of sensor or communication failure.

As such, Chapter 3 builds upon the advancements of Chapter 2 in order to create the first fully parallel, fully ad-hoc architecture for distributed data processing in a WSN that is also robust to sensor or communication failure. This architecture views a network of wireless sensors as a parallel computing system with an unknown and possibly changing number of processing nodes. As such, it is capable of processing sensor data in

parallel even in the midst of wireless nodes being added to or removed from the network. In this chapter, a specific distributed architecture is developed for the ad-hoc in-network updating of structural models using a wireless parallel simulated annealing (WPSA) stochastic optimization technique. This distributed method can be used to solve any of the combinatorial optimization problems that arise across engineering disciplines, and could potentially be used to detect the onset of structural damage or degradation. The WPSA technique is validated on a wireless sensor network by successfully updating a 6-DOF dynamic structural model with unknown mass, stiffness, and damping properties.

The parallel algorithms for in-network data processing presented in Chapter 2 and Chapter 3 represent a significant step towards the automation of complex data processing tasks within agent-based WSNs. However, one of the key challenges associated with in-network data processing still needing to be overcome is that within the wireless environment, many 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. Especially in networks where multiple computational tasks may need to be executed simultaneously, it is important to devise an autonomous, optimal method of distributing and consuming these scarce system resources throughout the network. In Chapter 4, the ad-hoc distributed data processing architecture developed in Chapter 3 is expanded to include resource optimization capabilities through the application of market-based techniques. Specifically, utility functions are developed for wireless nodes designated as both "buyers" and "sellers", allowing for an optimal utility-based assignment of sensing nodes in a given wireless network to a set of multiple computational tasks so as to minimize the consumption of limited network resources such

as battery power, CPU time, and wireless bandwidth. Using the *n*-Queens problem as a basis for validation, it is shown that the use of market principles to assign computational resources to multiple, simultaneously processed computational tasks allows for computational optimization in the wake of competing objectives such as power consumption, memory usage, and time to completion.

Lastly, in Chapter 5, the market-based resource allocation architecture developed in Chapter 4 is applied to the distributed modal identification techniques of Chapter 2. Specifically, a Market-Based Frequency Domain Decomposition (MBFDD) method is created in order allow for the autonomous formation of computational FDD topologies and to improve the accuracy of the distributed frequency domain decomposition mode shape estimates. This approach represents the first truly agent-based, truly ad-hoc approach to modal estimation using networks of wireless sensors. Using a simple cantilever beam as a testbed, the agent-based MBFDD technique is shown to be capable of autonomously forming computational topologies that optimize mode shape estimates (relative to a centralized FDD approach) while adaptively shifting, in real time, to changing computational objectives within the network (*i.e.*, mode shape accuracy, computational time, memory requirements, communication reliability, *etc.*).

6.2 Future Trends

The agent-based computational architectures presented in this dissertation represent a significant step towards the deployment of wireless monitoring systems that are capable of autonomously determining system properties and associating system properties with an analytical model. These capabilities can empower a sensor network to

validate structural design assumptions, calibrate analytical models, and possibly detect the onset of damage within a structural system. However, before truly autonomous wireless monitoring systems can be effectively deployed on a wide variety of structures, several additional advances must be made in the field of agent-based WSN computation.

From the system identification standpoint, it is widely agreed upon that no one method for modal estimation can be effectively applied in all cases. While the decentralized peak picking, random decrement, and frequency domain decomposition techniques developed in this dissertation have been shown to be effective at estimating modal frequencies, damping ratios, and mode shapes (respectively) in several different physical structures, each has its own limitations and cannot be applied in every application space and with every type of loading scenario. As such, it is important to build a toolbox of interchangeable algorithms that are truly parallel, and can be deployed in an agent-based architecture similar to the ones seen in Chapter 2 and Chapter 5. Methods such as the Eigensystem Realization Algorithm (ERA) (Juang and Pappa 1985) and the Natural Excitation Technique (NExT) (James*, et al.* 1992), for example, have both been applied to distributed wireless systems (Nagayama and Spencer Jr. 2007), but are not developed from a completely agent-based perspective. Once a complete toolbox of system identification tools is developed, an agent-based methodology must be created that can intelligently decipher modal estimates generated from multiple identification methods. The market-based architecture presented in Chapter 4 and Chapter 5 is one possible framework in which this problem may be addressed.

From the model updating perspective, it is clear that high-fidelity finite element models are necessary in order to associate experimentally identified global properties

with extremely localized structural information. The distributed simulated annealing architecture developed in Chapter 3 was a first step towards implementing model updating techniques within an agent-based network of wireless sensors, but it can be clearly seen that the simplistic analytical representations that this method is capable of analyzing in-network must be supplemented with more sophisticated structural models. One way in which the method developed in Chapter 3 could be extended for online model validation or low-level damage detection is by introducing some sort of "change threshold", or a measure of change in the properties of the simplistic analytical representation that may warrant additional investigation using a more sophisticated model. Because many wireless monitoring systems, when deployed in practice, are connected to the internet for data storage or data access purposes, an internet-based grid of high performance computers, running high fidelity finite element models, could be leveraged to investigate possible signs of damage as detected by the wireless network.

Lastly, from the resource allocation standpoint, it is clear that the ability to autonomously optimize scarce network resources across multiple computational tasks (Chapter 4) or throughout one particular task (Chapter 5) is an incredibly valuable contribution to the field of distributed data processing in WSNs. However, one of the drawbacks of the agent-based methods presented in both Chapter 4 and Chapter 5 is that each of these methodologies is reliant on the wireless network having some level of *a priori* knowledge about the task (or tasks) that it will be charged with completing. In Chapter 4, each computational task that the market-based resource allocation algorithm is tasked with assigning processors to must first be analyzed in simulation with respect to its computational and communication requirements. In Chapter 5, the advantages of creating

various FDD cluster sizes are dependent somewhat on sensor placement and on an *a priori* analytical model of the monitored structure. In both of these cases, it would be nice if new computational tasks (or, for that matter, new structures) could be exposed to the wireless network without the need for any additional knowledge to be infused into the wireless system. One area of research that could be applied in order to achieve this type of objective is that of machine learning. By applying reinforcement learning techniques to repeating allocation tasks (*i.e.*, computational task assignment and/or FDD topology creation), optimal allocation methodologies could be learned (and re-learned) over time as the resource allocation techniques are repeatedly applied. As such, new computational objectives or new monitored structures could be introduced to a given SHM system without the need for extensive *a priori* data processing or even reprogramming of the wireless sensing network.

REFERENCES

AKKAYA, K., DEMIRBAS, M. and AYGUN, R.S. (2008). "The impact of data aggregation on the performance of wireless sensor networks." *Wireless Communications and Mobile Computing*, 8(2), 171-193.

ALAMPALLI, S. (2000). "Effects of testing, analysis, damage, and environment on modal parameters." *Mechanical Systems and Signal Processing*, 14(1), 63-74.

ALLEMANG, R.J. (1999). *Vibrations: Experimental Modal Analysis*. Course Notes (UC-SDRL-CN-20-263-663/664), Structural Dynamics Research Library, University of Cincinnati, Cincinnati, OH.

ALLEMANG, R.J. and BROWN, D.L. (1982). "A correlation coefficient for modal vector analysis." *International Modal Analysis Conference & Exhibit*, Orlando, FL 110-116.

ANUSSORNNITISARN, P., NOF, S.Y. and ETZION, O. (2005). "Decentralized control of cooperative and autonomous agents for solving the distributed resource allocation problem." *International Journal of Production Economics*, 98(2), 114-128.

ARROW, K.J. and HURWICZ, L. (1960). Decentralization and computation in resource allocation. *Essays in Economics and Econometrics*. R.W. Pfouts. Raleigh, North Carolina, University of North Carolia Press.

ASMUSSEN, J.C. (1997). "Modal Analysis Based on the Random Decrement Technique – Application to Civil Engineering Structures." Ph.D. dissertation, University of Aalborg, Aalborg, Denmark.

ATHANASIADIS, I.N. and MITKAS, P.A. (2004). "An agent-based intelligent environmental monitoring system." *Management of Environmental Quality*, 15(3), 238.

BOLONI, L., KHAN, M.A. and TURGUT, D. (2006). "Agent-based coalition formation in disaster response applications." *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, Los Alamitos, CA.

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

BRINCKER, R., VENTURA, C.E. and ANDERSON, P. (2001a). "Damping estimation by frequency domain decomposition." *International Modal Analysis Conference* 698-703.

BRINCKER, R., ZHANG, L. and ANDERSON, P. (2001b). "Modal identification of output-only systems using frequency domain decomposition." *Smart Materials and Structures*, 10(3), 441-445.

BRYSON, S. (1995). "Data glut revisited." *Computers in Physics*, 9(5), 525-525.

BYERS, J. and NASSER, G. (2000). "Utility-based decision-making in wireless sensor networks." *1st Annual Workshop on Mobile and Ad Hoc Networking and Computing*, Boston, Massachussetts 143-144.

CELEBI, M. (2002). *Seismic instrumentation of buildings (with emphasis on federal buildings)*. Technical Report No. 0-7460-68170, United States Geological Survey (USGS), Menlo Park, CA.

CHINTALAPUDI, K., PAEK, J., GNAWALI, O., FU, T.S., DANTU, K., CAFFREY, J., GOVINDAN, R., JOHNSON, E. and MASRI, S. (2006). "Structural damage detection and localization using NetSHM." *5th International Conference on Information Processing in Sensor Networks*, Nashville, TN 475-482.

CLEARWATER, S.H. (1996). *Market-Based Control: A Paradigm for Distributed Resource Allocation*, World Scientific Press, Singapore.

COCCHI, R., SHENKER, S., ESTRIN, D. and ZHANG, L. (1993). "Pricing in computer networks: motivation, formulation, and example." *IEEE/ACM Transactions on Networking*, 1(6), 614-627.

COLE, H.A. (1968). "On-the-line analysis of random vibrations." *9th AIAA/ASME Structural Dynamics and Materials Conference*, Palm Springs, CA.

CUNHA, A. and CAETANO, E. (2006). "Experimental modal analysis of civil engineering structures." *Journal of Sound and Vibration*, 40(6), 12-20.

CURESCU, C. and NADJM-TEHRANI, S. (2005). "Price/utility-based optimized resource allocation in wireless ad hoc networks." *2nd Annual IEEE Communications Society Conference on Sensor and AdHoc Communications Networks*, Santa Clara, California 85-95.

DEDMAN, B. (2007). "I-35 bridge was rated among the nation's worst." *www.msnbc.com*, August 3, 2007.

DOEBLING, S.W., FARRAR, C.R. and PRIME, M.B. (1998). "A summary review of vibration-based damage identification methods." *Shock and Vibration Digest*, 30(2), 91-105.

ESTES, A.C. and FRANGOPOL, D.M. (2003). "Updating Bridge Reliability Based on Bridge Management Systems Visual Inspection Results." *Journal of Bridge Engineering*, 8(6), 374-382.

ESWARAN, S., MISRA, A. and LA PORTA, T. (2008). "Utility-based adaptation in mission-oriented wireless sensor networks." *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, San Francisco, California 278-286.

EWINS, D.J. (1986). *Modal Testing: Theory and Practice*, Research Studies Press, Ltd., Saunton, Somerset, England.

FANG, S.-E. and PERERA, R. (2009). "Power mode shapes for early damage detection in linear structures." *Journal of Sound and Vibration*, 324(1-2), 40-56.

FARRAR, C.R. (2001). *Historical overview of structural health monitoring*, Los Alamos Dynamics, Los Alamos, NM.

FARRAR, C.R. and WORDEN, K. (2007). "An introduction to structural health monitoring." *Philosophical Transactions of the Royal Society of London, Series A*, 365(1851), 303-315.

FEMA (2007). *I-35W Bridge Collapse and Response*. USFA-TR-166, U.S. Department of Homeland Security (Federal Emergency Management Agency), Minneapolis, MN.

FHWA (2004). *National Bridge Inspection Standards*. FHWA-2001-8954, Federal Highway Administration (FHWA).

GAO, Y. (2005). "Structural health monitoring strategies for smart sensor networks." Ph.D. dissertation, Department of Civil and Environmental Engineering, University of Illinois, Urbana-Champaign, IL.

GARDNER, J.W. (1994). *Microsensors: Principles and Applications*, John Wiley & Sons, West Sussex, England.

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

HASHIMOTO, Y., MASUDA, A. and SONE, A. (2005). "Prototype of sensor network with embedded local data processing." *Smart Structures and Materials Conference*, San Diego, CA 245-252.

HIPLEY, P. (2000). "Caltrans' current state-of-practice." *Instrumental Systems for Diagnostics of Seismic Response of Bridges and Dams*, Berkeley, CA 3-7.

HO, Y.C., SERVI, L. and SURI, R. (1980). "A class of center-free resource allocation algorithms." *Large Scale Systems*, 1(1), 51-62.

HOLFORD, K.M. (2009). "Acoustic emission in structural health monitoring." *Key Engineering Materials*, 413-414, 15-28.

HOMAIFAR, A., TURNER, J. and ALI, S. (1992). "The N-queens problem and genetic algorithms." *IEEE SOUTHEASTCON*, Birmingham, Alabama 262-267.

HOU, T.C. and LYNCH, J.P. (2006). "Rapid-to-deploy wireless monitoring systems for static and dynamic load testing of bridges: validation on the Grove Street bridge." *13th International Symposium on Smart Structures and Materials*.

HOU, Z., HERA, A. and SHINDE, A. (2006). "Wavelet-based structural health monitoring of earthquake excited structures." *Computer-Aided Civil and Infrastructure Engineering*, 21(4), 268-279.

IBRAHIM, S.R. (1977). "Random decrement technique for modal identification of structures." *Journal of Spacecraft and Rockets*, 14(11), 696-700.

ISMAIL, Z., ABDUL RAZAK, H. and ABDUL RAHMAN, A.G. (2006). "Determination of damage location in RC beams using mode shape derivatives." *Engineering Structures*, 28(11), 1566-1573.

JAMES, G.H., CARNE, T.G., LAUFFER, J.P. and NORD, A.R. (1992). "Modal testing using natural excitation." *10th International Modal Analysis Conference*, San Diego, CA.

JIN, J., WANG, W.H. and PALANISWAMI, M. (2007). "Application-oriented flow control for wireless sensor networks." *3rd International Conference on Networking and Services*, Athens, Greece 423-429.

JUANG, J.N. (1994). *Applied System Identification*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

JUANG, J.N. and PAPPA, R.S. (1985). "An Eigensystem realization algorithm for modal parameter identification and model reduction." *Journal of Guidance Control and Dynamics*, 8, 620-627.

KAO, Y.F. and HUANG, J.H. (2008). "Price-based resource allocation for wireless ad hoc networks with multi-rate capability and energy constraints." *Computer Communications*, 31, 3613-3624.

KEVIN, K.T. (2004). "Smart sensor technology for infrastructural integrity assessment and monitoring for homeland security." *Sensors, and Command, Control, Communications, and Intelligence Technologies for Homeland Security and Homeland Defense*, Orlando, FL 22-32.

KIRKPATRICK, S., GELATT JR., C.D. and VECCHI, M.P. (1983). "Optimization by simulated annealing." *Science*, 220, 671-680.

KO, J.M. and NI, Y.Q. (2005). "Technology developments in structural health monitoring of large-scale bridges." *Engineering Structures*, 27(12), 1715-1725.

KUROSE, J.F. and SIMHA, R. (1989). "A microeconomic approach to optimal resource allocation in distributed computer systems." *IEEE Transactions on Computers*, 38(5), 705-717.

LEVIN, R.I. and LIEVEN, N.A.J. (1998). "Dynamic finite element model updating using simulated annealing and genetic algorithms." *Mechanical Systems and Signal Processing*, 12(1), 91-120.

LICHTENSTEIN, A.G. (1993). "The Silver Bridge collapse recounted." *Journal of Performance of Constructed Facilities*, 7(4), 249.

LJUNG, L. (1987). *System Identification: Theory for the User*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

LOUTAS, T.H., KALAITZOGLOU, J., SOTIRIADES, G. and KOSTOPOULOUS, V. (2008). "A novel approach for continuous acoustic emission monitoring on rotating machinery without the use of slip ring." *Journal of Vibration and Acoustics (ASME)*, 130(6), 1-6.

LU, K.C., WANG, Y., LYNCH, J.P., LOH, C.H., CHEN, Y.J., LIN, P.Y. and LEE, Z.K. (2006). "Ambient vibration study of the Gi-Lu cable-stay bridge: Application of wireless sensing units." *13th International Symposium on Smart Structures and Materials*.

LUDWIG, M.J. and CONRARDY, C. (2007). "Portable weld inspection management system." *Journal of Ship Production*, 23(3), 147-160.

LYNCH, J.P. (2002). "Decentralization of wireless monitoring and control techniques for smart civil structures." Ph.D. dissertation, John A. Blume Earthquake Engineering Center, Stanford University, Stanford, CA.

LYNCH, J.P. (2007). "An overview of wireless structural health monitoring for civil structures." *Philosophical Transactions of the Royal Society of London, Series A, Mathematical and Physical Sciences*, 365, 345-372.

LYNCH, J.P. and LOH, K.J. (2006). "A summary review of wireless sensors and sensor networks for structural health monitoring." *The Shock and Vibration Digest*, 38(2), 91-128.

LYNCH, J.P., SUNDARARAJAN, A., LAW, K.H., KIREMIDJIAN, A.S. and CARRYER, E. (2004). "Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency." *Smart Materials and Structures*, 13, 800-810.

LYNCH, J.P., WANG, Y., LOH, K., YI, J.H. and YUN, C.B. (2006). "Performance monitoring of the Geumdang Bridge using a dense network of high-resolution wireless sensors." *Smart Materials and Structures*, 15(6), 1561-1575.

MAHOON, A. (1988). "The role of non-destructive testing in the airworthiness certification of civil aircraft composite structures." *Composites*, 19(3), 229-235.

MALHOTRA, V.M. and CARINO, N.J. (2004). *Handbook on nondestructive testing of concrete*, CRC Press, LLC, Boca Raton, FL.

MAS-COLELL, A., WHINSTON, M.D. and GREEN, J.R. (1995). *Microeconomic Theory*, Oxford University Press, New York, New York.

MCCANN, D.M. and FORDE, M.C. (2001). "Review of NDT methods in the assessment of concrete and masonry structures." *NDT & E International*, 34(2), 71-84.

MCCAULEY, L. and FRANKLIN, S. (2002). "A large-scale multi-agent system for navy personnel distribution." *Connection Science*, 14(4), 371-85.

METROPOLIS, N., ROSENBLUTH, A.W., ROSENBLUTH, M.N. and TELLER, A.H. (1953). "Equation of state calculations by fast computing machines." *Journal of Chemical Physics*, 21(6), 1087-1092.

MOHAMED, S. and PETER, R. (2004). "Embedded MEMS for health monitoring and management of civil infrastructure." *Smart Structures and Materials - Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, San Diego, CA 331-343.

MOTTERSHEAD, J.E. and FRISWELL, M.I. (1993). "Model updating in structural dynamics: A survey." *Journal of Sound and Vibration*, 167(2), 347-375.

NAGAYAMA, T., SPENCER, B.F., AGHA, G.A. and MECHITOV, K.A. (2006). "Model-based data aggregation for structural monitoring employing smart sensors." *3rd International Conference on Networked Sensing Systems*, Chicago, IL.

NAGAYAMA, T. and SPENCER JR., B.F. (2007). *Structural health monitoring using smart sensors*. Newmark Structural Engineering Laboratory Report No. 1, University of Illinois at Urbana-Champaign, Urbana, IL.

NI, Y.Q., ZHOU, H.F., CHAN, K.C. and KO, J.M. (2008). "Modal flexibility analysis of cable-stayed Ting Kau bridge for damage identification." *Computer-Aided Civil and Infrastructure Engineering*, 23(3), 223-236.

NIST (2005). *Final report on the collapse of the World Trade Center towers*, National Institute of Standards and Technology (NIST), Gaithersburg, MD.

O'MALLEY, K. (2001). "Agents and automated online trading." *Dr. Dobb's Journal*, 26(5), 23-8.

PARJAJKA, J. and BLOSCHL, G. (2008). "The value of MODIS snow cover data in validating and calibrating conceptual hydrologic models." *Journal of Hydrology*, 358(3-4), 240-258.

PARK, H.W. and SOHN, H. (2006). "Parameter estimation of the generalized extreme value distribution for structural health monitoring." *Probabalistic Engineering Mechanics*, 21(4), 366-376.

RAGHAVAN, A. and CESNIK, C.E.S. (2007). "Review of guided-wave structural health monitoring." *The Shock and Vibration Digest*, 39(2), 91-114.

RANDALL, R.B. (2004a). "State of the art in monitoring rotating machinery - Part 1." *Sound and Vibration*, 38(3), 14-21.

RANDALL, R.B. (2004b). "State of the art in monitoring rotating machinery - Part 2." *Sound and Vibration*, 38(5), 10-17.

RAO, A.R.M. and ANANDAKUMAR, G. (2007). "Optimal placement of sensors for structural system identification and health monitoring using a hybrid swarm intelligence technique." *Smart Materials and Structures*, 16(6), 2658-2672.

REINA, P. (2004a). "Focus on construction of columns at airport–section of year-old concrete concourse at Paris airport dropped, killing at least four people." *Engineering News Record*, May 2004.

REINA, P. (2004b). "Questions raised over steel roof struts–probe says steel tubes, not columns, are likely trigger for Paris airport's concourse collapse." *Engineering News Record*, July 2004.

ROHL, J.S. (1983). "A faster lexicographical N queens algorithm." *Information Processing Letters*, 17(5), 231-233.

ROSEMARK, R. and LEE, W.C. (2005). "Decentralizing query processing in sensor networks." *2nd International Conference on Mobile and Ubiquitous Systems*, San Diego, California 270-280.

RUIZ-SANDOVAL, M. (2004). "Smart Sensors for Civil Infrastructure Systems." Ph.D. Dissertation, University of Notre Dame, South Bend, IN.

RUSSELL, S.J. and NORVIG, P. (2003). *Artificial Intelligence: A Modern Approach*, Pearson Education, Inc., Upper Saddle River, New Jersey.

SAMSONOV, P. (1995). "Nondestructive visual inspection of aging aircraft." *SPIE - Nondestructive Evaluation of Aging Aircraft, Airports, Aerospace Hardware, and Materials*, Oakland, CA 190-196.

SCHMITZ, T., RUTZEN, W. and JOKAT, W. (2007). "Cable-based geophysical measurement and monitoring systems, new possibilities for tsunami early-warnings." *2007 Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies* 301-304.

SETAREH, M. (1990). "Use of Tuned Mass Dampers for the Vibration Control of Floors Subjected to Human Movements." Ph.D. dissertation, University of Michigan, Ann Arbor, Michigan.

SHENKER, S. (1995). "Fundamental design issues for the future Internet." *IEEE Journal on Selected Areas in Communications*, 13(7), 1176-1188.

SIM, S.H., SPENCER, B.F., ZHANG, M. and XIE, H. (2009). "Automated decentralized smart sensor network for modal analysis." *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, San Diego, CA 72920W-1-72920W-1-12.

SIMMERS JR., G.E., SODANO, H.A., PARK, G. and INMAN, D.J. (2006). "Detection of corrosion using piezoelectric impedance-based structural health monitoring." *AIAA Journal*, 44(11), 2800-2803.

SINHA, J.K. and FRISWELL, M.I. (2002). "Model updating: A tool for reliable modelling, design modification and diagnosis." *The Shock and Vibration Digest*, 34(1), 27-35.

SLAUGHTER, S.B., CHEUNG, M.C., SUCHARSKI, D. and COWPER, B. (1997). *State of the Art in Hull Monitoring Systems*. Report No. SSC-401, Ship Structure Committee, Washington, DC.

SOSIC, R. and GU, J. (1991). "Fast search algorithms for the N-queens problem." *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6), 1572-1576.

SPENCER, B.F., RUIZ-SANDOVAL, M.E. and KURATA, N. (2004). "Smart sensing technology: opportunities and challenges." *Journal of Structural Control and Health Monitoring*, 11, 349-368.

STASZEWSKI, W.J., BOLLER, C. and TOMLINSON, G. (2004). *Health Monitoring of Aerospace Structures: Smart Sensor Technologies and Signal Processing*, John Wiley & Sons Ltd., West Sussex, England.

STASZEWSKI, W.J., MAHZAN, S. and TRAYNOR, R. (2009). "Health monitoring of aerospace composite structures – active and passive approach." *Composites Science and Technology*, 69(11), 1678-1685.

STRASER, E.G., KIREMIDJIAN, A.S., MENG, T.H. and REDLEFSEN, L. (1998). "Modular, wireless network platform for monitoring structures." *16th Interational Modal Analysis Conference (IMAC)*, Santa Barbara, CA 450-456.

SWARTZ, R.A., JUN, D., LYNCH, J.P., WANG, Y., SHI, D. and FLYNN, M. (2005). "Design of a wireless sensor for scalable distributed in-network computation in a structural health monitoring system." *International Workshop on Structural Health Monitoring*, Stanford, California.

TEUGHELS, A. and DEROECK, G. (2004). "Structural damage identification of the highway bridge Z24 by FE model updating." *Journal of Sound and Vibration*, 278(3), 589-610.

WANG, Y., LOH, K.J., LYNCH, J.P., FRASER, M., LAW, K. and ELGAMAL, A. (2006). "Vibration monitoring of the Voigt Bridge using wired and wireless monitoring systems." *4th China-Japan-US Symposium on Structural Control and Monitoring*.

WANG, Y., LYNCH, J.P. and LAW, K.H. (2005). "Wireless structural sensors using reliable communication protocols for data acquisition and interrogation." *23rd International Modal Analysis Conference*.

WASHER, G.A. (1998). "Developments for the non-destructive evaluation of highway bridges in the USA." *NDT & E International*, 31(4), 245-249.

WU, J.R. and LI, Q.S. (2006). "Structural parameter identification and damage detection for a steel structure using a two-stage finite element model updating method." *Journal of Constructional Steel Research*, 62(3), 231-239.

WU, W.-F. and HUANG, T.-H. (1993). "Prediction of fatigue damage and fatigue life under random loading." *The International Journal of Pressure Vessels and Piping*, 53(2), 273-298.

WU, Z.S. (2003). Structural health monitoring and intelligent infrastructures in Japan. *Structural Health Monitoring and Intelligent Infrastructure*. M. Abe and Z.S. Wu. Lisse, Balkema. 1: 153-167.

ZAGRAI, A.N. and GIURGIUTIU, V. (2001). "Electro-mechanical impedance method for crack detection in thin plates." *Journal of Intelligent Material Systems and Structures*, 12(10), 709-718.

ZIMMERMAN, A.T. and LYNCH, J.P. (2009). "A parallel simulated annealing architecture for model updating in wireless sensor networks." *IEEE Sensors Journal*, 9(11), 1503-1510.

ZIMMERMAN, A.T., LYNCH, J.P. and FERRESE, F.T. (2009). "Market-based computational task assignment within autonomous wireless sensor networks." *IEEE International Conference on Electro/Information Technology*, Windsor, Canada 23-28.

ZIMMERMAN, A.T., SHIRAISHI, M., SWARTZ, R.A. and LYNCH, J.P. (2008a). "Automated modal parameter estimation by parallel processing within wireless monitoring systems." *ASCE Journal of Infrastructure Systems*, 14(1), 102-113.

ZIMMERMAN, A.T., SWARTZ, R.A. and LYNCH, J.P. (2008b). "Automated identification of modal properties in a steel bridge instrumented with a dense wireless sensor network." *4th International Conference on Bridge Maintenance, Safety and Management*, Seoul, Korea.