

A Parallel Simulated Annealing Architecture for Model Updating in Wireless Sensor Networks

Andrew T. Zimmerman, *Student Member, IEEE*, and Jerome P. Lynch, *Member, IEEE*

Abstract—In recent years, wireless sensing technologies have provided a much sought-after alternative to expensive cabled monitoring systems. Wireless sensing networks forego the high data transfer rates associated with cabled sensors in exchange for low-cost and low-power communication between a large number of sensing devices, each of which features embedded data processing capabilities. As such, a new paradigm in large-scale data processing has emerged; one where communication bandwidth is somewhat limited but distributed data processing centers are abundant. By taking advantage of this grid of computational resources, data processing tasks once performed independently by a central processing unit can now be parallelized, automated, and carried out within a wireless sensor network. By utilizing the intelligent organization and self-healing properties of many wireless networks, an extremely scalable multiprocessor computational framework can be developed to perform advanced engineering analyses. In this study, a novel parallelization of the simulated annealing stochastic search algorithm is presented and used to update structural models by comparing model predictions to experimental results. The resulting distributed model updating algorithm is validated within a network of wireless sensors by identifying the mass, stiffness, and damping properties of a three-story steel structure subjected to seismic base motion.

Index Terms—In-network data processing, parallel algorithms, structural monitoring, wireless sensors.

I. INTRODUCTION

COMPLEX structural systems such as buildings, bridges, pipelines, and ships, among many others, 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 to evaluate system performance either in the wake of time-based deterioration (such as cracking or corrosion) or after extreme loading scenarios (such as earthquakes). 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. While tethered SHM systems have been successfully installed on a number of large structures around the world [1], the SHM community has begun to investigate the use of wireless communication as a way of increasing the affordability of large-scale installations. Specifically, wireless sensors can eliminate the high costs associated with running cable from spatially distant sensors to a central data repository.

The idea of integrating traditional sensing components with a wireless radio for the purpose of structural monitoring was first proposed in 1998 [2]. Since then, many wireless sensing platforms have been developed for both commercial and academic use. These devices typically integrate a low-power microprocessor, an A/D converter, and a wireless transceiver and can be manufactured for a few hundred dollars per sensing node [3]. This price point represents considerable savings when compared with traditional cabled SHM systems, which have been reported to incur installation costs of several thousand dollars per sensing channel in large structures [4]. 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. This type of local data processing has several advantages within a wireless SHM system. Primarily, by processing data locally, a WSN can convert traditionally high bandwidth streams of raw data into relatively low bandwidth streams of processed data. In the wireless domain, this transition results in a decrease in communication demand that can help maximize power efficiency (as the radio is typically the most power-intensive element in the wireless system) and can improve network utilization (which is important because of the limited wireless bandwidth available for shared use) [5].

Because of these benefits, many different architectures for embedded data processing using wireless sensors have been proposed in the literature. Early on, researchers focused primarily on centralized implementations of engineering algorithms that required no communication between sensors. For example, algorithms involving autoregressive model fitting [6], wavelet transforms [7], and fast Fourier transforms [6] were all modified for local execution by a wireless sensor. Each of these embedded algorithms collects and processes raw measurement data available at each sensor with processed results transmitted one-by-one back to a central data repository for further analysis. These embedded data processing methods are relatively power efficient when compared to the transfer of raw time history data to a central location [5]. However, there is little to no sharing of sensor data (raw or processed) between nodes, preventing these centralized algorithms from autonomously producing

Manuscript received September 12, 2008; revised November 27, 2008; accepted January 05, 2009. Current version published October 02, 2009. This work was supported in part by the U.S. Office of Naval Research under Grant N00014-05-1-0596 administered by Program Manager A. Seman, and in part by the National Science Foundation under Grant CMMI-0726812. The work of A. T. Zimmerman was supported by the National Defense Science and Engineering Graduate Fellowship Program. The associate editor coordinating the review of this paper and approving it for publication was Prof. Norbert Meyendorf.

The authors are with the University of Michigan, Ann Arbor, MI 48109 USA (e-mail: atzimmer@umich.edu; jerlynch@umich.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSEN.2009.2019323

system-wide spatial properties (e.g., mode shapes). Additionally, because of their inherently centralized communication architecture, these methods are not scalable to large or spatially diverse sensor networks.

As such, several hierarchical data processing methods were later developed in order to improve network scalability and to allow for spatial in-network reasoning through data sharing. For example, by presenting a tiered network architecture where computational power is largely consolidated within a set of powerful gateway nodes, Chintalapudi *et al.* [8] created a wireless framework where sets of sensor data can be processed together and communication can be streamlined through the use of the high-powered nodes. This approach allows for a very flexible and abstracted user interface but underutilizes the computational capabilities of the majority of a network's nodes and presents an uneven balance of power consumption throughout the network. Similarly, other researchers have used hierarchical network architectures to aggregate and compress local sensor data [9], [10]. Such approaches improve network scalability by limiting data transmission and minimizing data loss but rely on a stark tradeoff between data size and computational accuracy. More recently, Zimmerman *et al.* [11] leveraged the peer-to-peer communications capability inherent to many wireless sensor platforms to create a parallel data processing environment for use within dense WSNs. By distributing computational tasks in a parallel fashion, 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. However, this parallel architecture relies on fixed network topologies that are incapable of producing correct or coherent results in the wake of communication and/or sensor failure.

In this study, a novel data processing architecture that builds on this previous work is created for use within a 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. This novel architecture functions by allowing a network of sensors to autonomously detect and utilize the computing resources of any available wireless node on the fly. This "ad hoc" capability allows for increases in the parallelism and efficiency of the architecture in real-time and can be used to reform or "self-heal" the network in the wake of any communication and/or sensor failures.

In order to examine the data processing capabilities of this novel architecture from an 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 study, the wireless parallel SA (WPSA) method is used within a WSN for the updating of structural models. This type of model updating can be used for many purposes in SHM, including the detection of damage or degradation within a

structure. For validation, 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.

II. BACKGROUND ON COMBINATORIAL OPTIMIZATION BY SA

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 1980s, several algorithms derived from physical and biological systems were developed for finding near-global optima in functions containing many local optima [12]. One of these methods is the SA optimization technique, first presented by Kirkpatrick *et al.* [13] in 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 among 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 "energy" of the physical system and then introducing an "effective" annealing temperature which will simulate the material cooling process. The Metropolis procedure is then used to avoid premature convergence on local optima, which is the key to the effectiveness of the generalized annealing process.

In 1953, Metropolis *et al.* [14] 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 by 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:

$$\Pr(\text{accept}) = e^{\frac{-\Delta E}{k_B \cdot T}} \quad (1)$$

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, "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 suboptimal 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, $E(s_{\text{new}})$, is then compared with the objective function value of the old state, $E(s_{\text{old}})$, and the new state is probabilistically accepted or rejected based on the criterion presented in (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:

$$E(s_{\text{new}}) \leq E(s_{\text{old}}) + T \cdot |\ln(U)| \quad (2)$$

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 suboptimal 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 pseudorandomly 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 (2). After N iterations, the temperature of the system is reduced by a factor of ρ , such that $T_{\text{new}} = \rho \cdot T_{\text{old}}$, and N additional iterations will be run at the new, lower temperature. 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 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

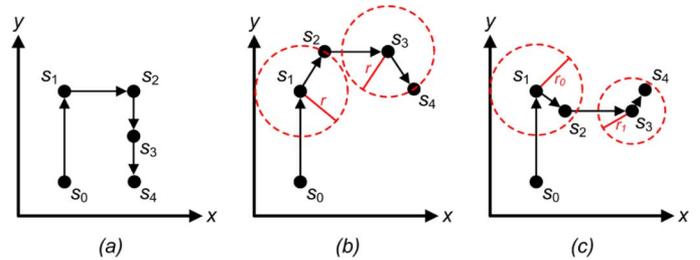


Fig. 1. Random state generation for a 2-D search problem using (a) standard SA, (b) BSA, and (c) WPSA.

studied in this paper, a modification on the blended SA (BSA) algorithm proposed by Levin and Lieven is exclusively utilized [15]. 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 [Fig. 1(a)]. In the BSA algorithm, this standard type of state generation is alternated every other step with a "radius adjustment" approach, where all annealing parameters are changed by choosing a random point on a hypersphere that is a fixed radius away from the previous annealing state [Fig. 1(b)]. This method requires two 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 [Fig. 1(c)]. Then, the radius itself is treated as a variable in the SA process much like the annealing temperature. It starts with a high value near 1.0 (such that the entirety of each parameter's valid range can be searched) and over time is reduced such that the SA search focuses increasingly on values that are close to the currently optimal state. By focusing on a smaller portion of the SA search space, this change may decrease the chances of finding a global minimum. However, in most cases, it 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.

III. 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

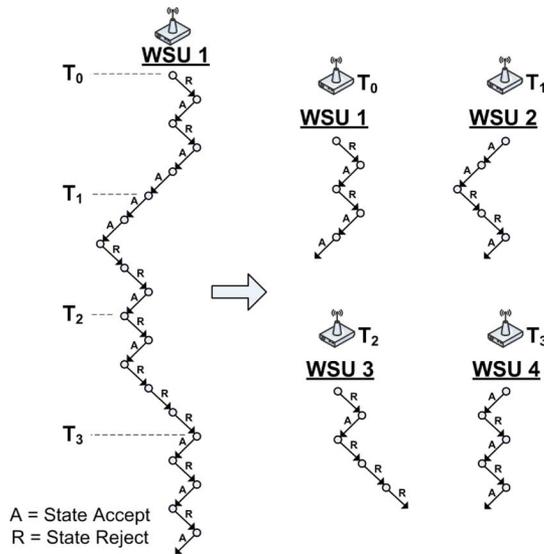


Fig. 2. Simple serial SA search tree, shown up to the fourth temperature step along with its corresponding WPSA search trees, assigned to wireless sensors.

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 [16]. 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 WSNs while minimizing the communication demands of the parallel algorithm. This is done by taking advantage of the fact that the SA process typically rejects more states than it accepts, especially as the annealing temperature is lowered and the algorithm converges on a solution. Specifically, the traditionally serial SA search problem (which is continuous across all temperature steps) can be broken into a set of smaller search trees, each of which corresponds to a given temperature step and begins with the globally optimal state assignment so far detected at the preceding temperature step. Each smaller search problem can then be assigned individually to any available sensor in the network, and thus, multiple temperature steps can be searched concurrently. This concept is displayed graphically in Fig. 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.

A. Wireless Implementation of the Parallel SA Algorithm

In the parallel SA implementation used in this study (WPSA), a computational task requiring SA optimization is first assigned to any one available sensing unit, along with a user-defined ini-

tial 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 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 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.

B. Wireless Sensing Hardware

While the parallel data processing architecture developed herein can be embedded in almost any wireless sensing hardware, the Narada wireless sensor (Fig. 3) developed at the

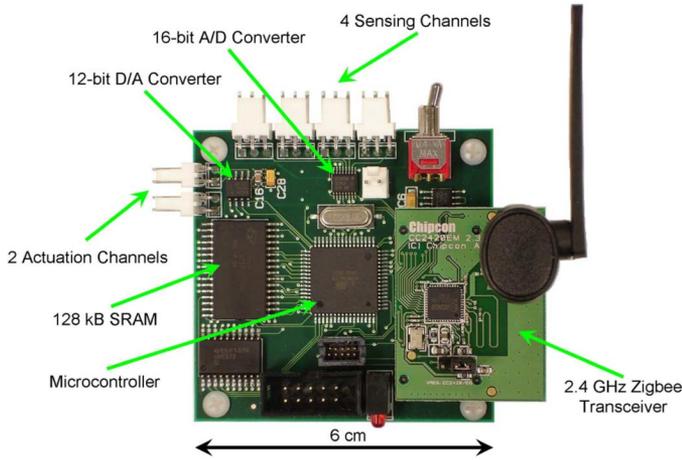


Fig. 3. Narada wireless sensing prototype.

University of Michigan is adopted for use in this study [17]. This unit's computational capabilities are provided by an Atmel Atmega128 microcontroller that is supplemented with 128 kB of external SRAM for additional data storage capacity. The Narada is also equipped with a 4-channel 16-bit Texas Instruments ADS8341 ADC capable of high-resolution data acquisition at sampling rates up to 100 kHz and a Chipcon CC2420 IEEE 802.15.4 compliant wireless radio for scalable wireless communication between sensors.

IV. EXPERIMENTAL EVALUATION OF THE WPSA ALGORITHM

A. Overview of Model Updating in SHM

In SHM, CO techniques have been successfully used for many purposes, including determining optimal sensor placement [18], establishing decision boundaries for damage identification [19], and updating model parameters to fit experimental sensor data [15]. 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 [20]. These results can then be used to track structural performance over time or to look for signs of long-term structural degradation [21]. This approach has been used to effectively detect and locate damage in a variety of real structures [22], [23].

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

$$E = \alpha \cdot \sum_{i=1}^q \left(\frac{\omega_{ai} - \omega_{ei}}{\omega_{ei}} \right)^2 + \beta \cdot \sum_{i=1}^q \left(\frac{1 - \sqrt{MAC_i}}{MAC_i} \right)^2 + \gamma \cdot \sum_{i=1}^q \left(\frac{\zeta_{ai} - \zeta_{ei}}{\zeta_{ei}} \right)^2 + f(t) \quad (3)$$

where ω_{ai} and ω_{ei} are the i th analytical and experimental modal frequencies, respectively, ζ_{ai} and ζ_{ei} are the i th analytical and experimental modal damping ratios, respectively, and MAC_i is the modal assurance criteria, a measure of correlation between the i th analytical and experimental mode shapes [24]. It is assumed that there are q modes in the system. Finally, α , β , and γ are weighting constants determined experimentally to account for differences in the magnitudes of the three modal parameters. In this paper, α is taken to be 0.6, β is taken to be 0.1, and γ is taken to be 0.005. A optional function $f(t)$ can be added in order to match experimental and analytical data characterized in the time domain.

Once formulated, the objective function in (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.

B. Experimental Validation on a Seismically Excited Steel Building

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 test bed. This structure sits on a 5 m \times 5 m triaxial seismic shaking table. Each floor of this structure [seen in Fig. 4(a)] is 3 m (9.84 ft) wide by 2 m (6.56 ft) deep and 3 m (9.84 ft) tall. Seismic ground motion is applied parallel to the longer floor dimension. A mass of approximately 6000 kg (13 228 lb) is supported by each floor. Four H150 \times 150 \times 7 \times 10 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 2000 kN/m (11 420 lb/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 is also placed at ground level to measure the ground acceleration driving the system.

C. Analytical Model of the Dynamic Structural System

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 [Fig. 4(b)] to analytically describe the dynamic response of the steel structure. This model is characterized by the following equation of motion:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = -\mathbf{M}\mathbf{1}\ddot{u}_g(t) \quad (4)$$

where $\mathbf{u} \in \mathbb{R}^{9 \times 1}$ is a vector of displacements (relative to the ground) and rotations for each of the m degrees of freedom, $\mathbf{M} \in \mathbb{R}^{9 \times 9}$, $\mathbf{C} \in \mathbb{R}^{9 \times 9}$, and $\mathbf{K} \in \mathbb{R}^{9 \times 9}$ are the structure's mass, damping, and stiffness matrices, respectively, $\mathbf{1} \in \mathbb{R}^{9 \times 1}$

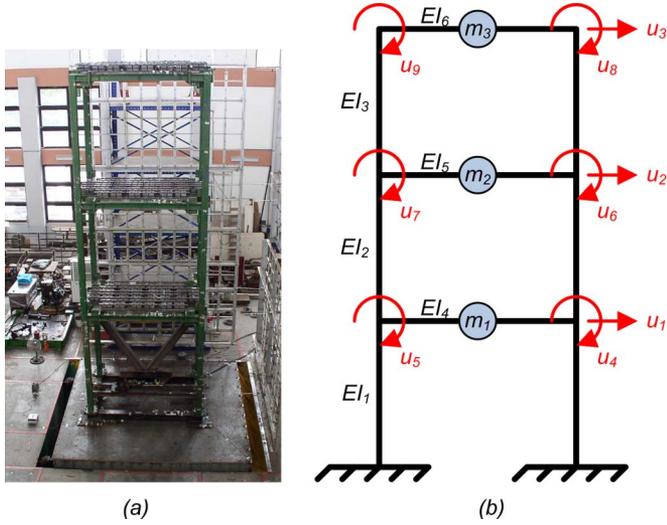


Fig. 4. Three-story (a) structure and (b) model used in this study.

is the 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 $\mathbf{M}_R \in \mathbb{R}^{3 \times 3}$ and $\mathbf{K}_R \in \mathbb{R}^{3 \times 3}$ matrices. A Rayleigh damping matrix, $\mathbf{C}_R \in \mathbb{R}^{3 \times 3}$, is constructed given the modal damping ratios in two lateral modes [25]. 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:

$$\dot{\mathbf{z}}(t) = \mathbf{A} \mathbf{z}(t) + \mathbf{B} \ddot{u}_g(t) \quad (5)$$

$$\mathbf{y}(t) = \mathbf{C} \mathbf{z}(t) + \mathbf{D} \ddot{u}_g(t) \quad (6)$$

where $\mathbf{z} \in \mathbb{R}^{6 \times 1}$ is the system state vector ($\mathbf{z}^T = \{\mathbf{u}^T \dot{\mathbf{u}}^T\}$), $\mathbf{y} \in \mathbb{R}^{3 \times 1}$ 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 \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} can be expressed as follows:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ -\mathbf{M}_R^{-1} \mathbf{K}_R & -\mathbf{M}_R^{-1} \mathbf{C}_R \end{bmatrix} \in \mathbb{R}^{6 \times 6} \\ \mathbf{B} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \in \mathbb{R}^{6 \times 1} \\ \mathbf{C} &= [-\mathbf{M}_R^{-1} \mathbf{K}_R \quad -\mathbf{M}_R^{-1} \mathbf{C}_R] \in \mathbb{R}^{3 \times 6} \\ \mathbf{D} &= [\mathbf{1}] \in \mathbb{R}^{3 \times 1}. \end{aligned} \quad (7)$$

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 \mathbf{A} . Specifically:

$$\omega_i = |\lambda_i|, \phi_i = \psi_i, \zeta_i = \cos[\tan^{-1}(\text{Im}(\lambda_i)/\text{Re}(\lambda_i))] \quad (8)$$

where ω_i is expressed in radians per second.

D. Model Updating of the Three-Story Steel Test Structure

In this study, three unknown mass (m_i) values, six unknown stiffness (EI_j) values, and two unknown modal damping (ζ_k) ratios are treated as updating parameters. The

WPSA algorithm is used to stochastically search the range of each of these eleven model parameters for a system state ($s = \{m_1 \dots m_3, EI_1 \dots EI_6, \zeta_1 \dots \zeta_2\}$), 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 (3). 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 ($\omega_i, \phi_i, \zeta_i$) used in the objective function. By employing peak picking, frequency domain decomposition, and random decrement methods already embedded in the wireless sensors [11], modal frequency, damping ratio, and mode shape 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.

E. 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 (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 I and graphically in Fig. 5(c), the WPSA algorithm is capable of producing a good match between analytical and experimental frequency domain properties in the three-story structure. Additionally, Fig. 5(a) and (b) show the similarity between experimental and analytical acceleration time history responses to the same random ground excitation.

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. Fig. 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

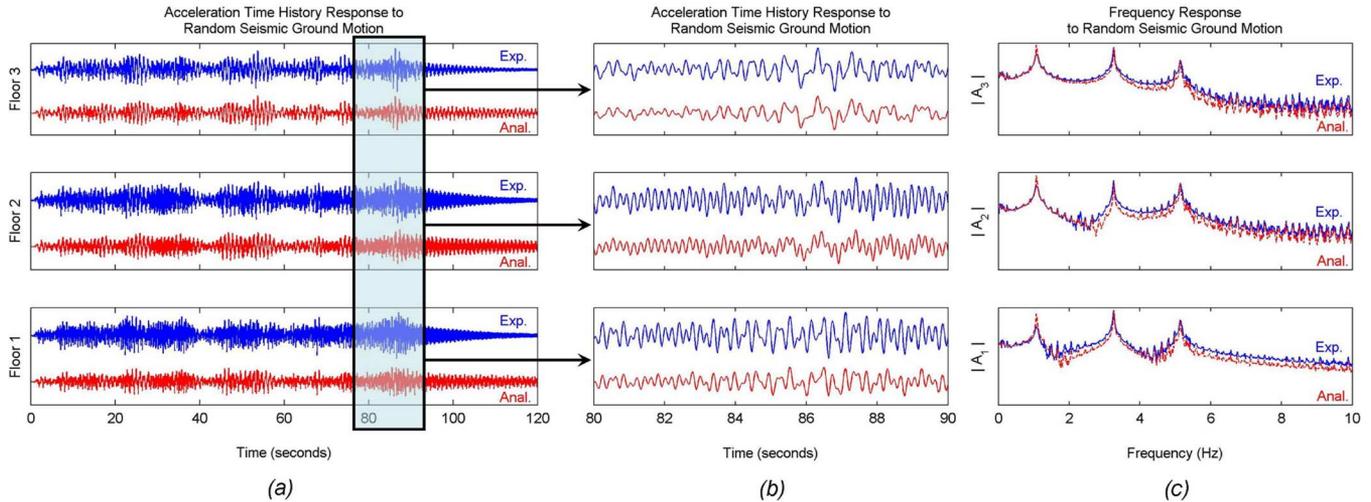


Fig. 5. Comparison of experimental and updated analytical model response to seismic base motion in the (a) and (b) time and (c) frequency domains.

TABLE I
COMPARISON OF EXPERIMENTALLY SENSED AND ANALYTICALLY DERIVED MODAL PROPERTIES FOR THREE-STORY STRUCTURE

Data Source	Floor Mass (kg)			Column Stiffness (kN/m)			Floor Stiffness (MN/m)			Modal Frequency (Hz)			MAC Value			Damping Ratio (%)		
	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

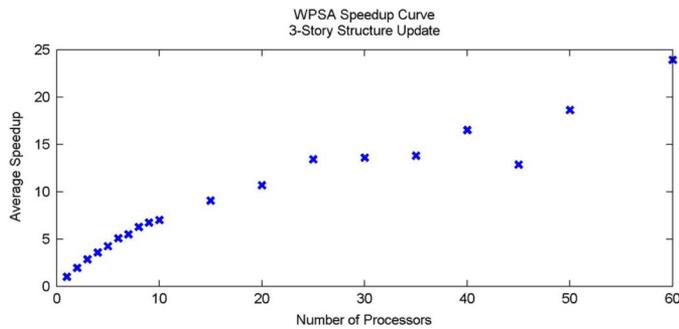


Fig. 6. Experimental speedup curve for structural model updating using WPSA.

the three-story model presented in this study is decreased from 8.5 h to 23 min. While this is significantly longer than the amount of time a PC would require to perform the same task, 23 min 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 (3), the Atmel Atmega128 used in this study took 350 ms plus 500 ms for every 100 time history points projected using the Newmark numerical integration scheme. The algorithm consumed 35% of the 128 kB of available internal flash ROM, 57% of the 4 kB of available internal SRAM, and 2.33%

of the 128 kB 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.

V. CONCLUSION

This study builds upon previous work in the parallel processing of data on wireless sensor networks by proposing a parallel SA algorithm designed to efficiently utilize the distributed resources available in large networks of wireless devices. This algorithm gains efficiency as the number of sensors in a network grows, making it scalable to very large networks, and it can be applied to many of the large number of CO problems seen across many engineering disciplines. 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. By altering the model's structural properties (mass, stiffness, and damping) such that the analytical modal output (frequencies, damping ratios, and mode shapes) match experimentally sensed data, properties of the physical structure can be accurately estimated. Future work in this area will involve incorporating more sophisticated interaction between sensors in order to optimally distribute resources throughout the wireless network and will utilize more complex finite element models in order to improve the accuracy of the model updating method and push the computational limits of the embedded environment.

ACKNOWLEDGMENT

The authors thank Prof. C.-H. Loh (National Taiwan University) and Prof. K. H. Law (Stanford University) for their assistance with the experimental portion of this study.

REFERENCES

- [1] J. M. Ko and Y. Q. Ni, "Technologies developments in structural health monitoring of large-scale bridges," *Eng. Struct.*, vol. 27, no. 12, pp. 1715–1725, 2005.
- [2] E. G. Straser, A. S. Kiremidjian, T. H. Meng, and L. Redlfeisen, "A modular, wireless network platform for monitoring structures," in *Proc. Int. Modal Anal. Conf. Exhibit*, Santa Barbara, CA, 1998, pp. 450–456.
- [3] J. P. Lynch and K. Loh, "A summary review of wireless sensors and sensor networks for structural health monitoring," *Shock Vib. Dig.*, vol. 38, no. 2, pp. 91–128, 2006.
- [4] M. Celebi, "Seismic instrumentation of buildings (with emphasis on federal buildings)," USGS, Menlo Park, CA, Tech. Rep. No. 0-7460-68170, 2002.
- [5] J. P. Lynch, A. Sundararajan, K. H. Law, A. S. Kiremidjian, and E. Carryer, "Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency," *Smart Mater. Struct.*, vol. 13, pp. 800–810, 2004.
- [6] J. P. Lynch, "Decentralization of wireless monitoring and control technologies for smart civil structures," John A. Blume Earthquake Engineering Center, Stanford University, Stanford, CA, Tech. Rep. 140, 2002.
- [7] J. P. Lynch, A. Sundararajan, K. H. Law, A. S. Kiremidjian, and E. Carryer, "Power-efficient data management for a wireless structural monitoring system," in *Proc. 4th Int. Workshop Struct. Health Monit.*, Stanford, CA, 2003, pp. 1177–1184.
- [8] K. Chintalapudi, J. Paek, O. Gnawali, T. S. Fu, K. Dantu, J. Caffrey, R. Govindan, E. Johnson, and S. Masri, "Structural damage detection and localization using NetSHM," in *Proc. 5th Int. Conf. Inform. Process. Sens. Netw.*, Nashville, TN, 2006, pp. 475–482.
- [9] Y. Gao, "Structural health monitoring strategies for smart sensor networks," Ph.D. dissertation, Dept. Civil Environ. Eng., Univ. Illinois at Urbana-Champaign, Urbana, IL, 2005.
- [10] T. Nagayama, B. F. Spencer, G. A. Agha, and K. A. Mechtov, "Model-based data aggregation for structural health monitoring employing smart sensors," in *Proc. 3rd Int. Conf. Networked Sensing Syst.*, Chicago, IL, 2006, pp. 203–210.
- [11] A. T. Zimmerman, M. Shiraishi, R. A. Swartz, and J. P. Lynch, "Automated modal parameter estimation by parallel processing within wireless monitoring systems," *J. Infrastruct. Syst.*, vol. 14, no. 1, pp. 102–113, 2008.
- [12] D. G. Bounds, "New optimization methods from physics and biology," *Nature*, vol. 329, pp. 215–219, 1987.
- [13] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [14] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [15] R. I. Levin and N. A. J. Lieven, "Dynamic finite element model updating using simulated annealing and genetic algorithms," *Mech. Syst. Signal Process.*, vol. 12, no. 1, pp. 91–120, 1998.
- [16] D. R. Greening, "Parallel simulated annealing techniques," *Physica D*, vol. 42, pp. 293–306, 1990.
- [17] R. A. Swartz, D. Jun, J. P. Lynch, Y. Wang, D. Shi, and M. Flynn, "Design of a wireless sensor for scalable distributed in-network computation in a structural health monitoring system," in *Proc. Int. Workshop Struct. Health Monit.*, Stanford, CA, 2005, pp. 1570–1577.

- [18] A. R. M. Rao and G. Anandakumar, "Optimal placement of sensors for structural system identification and health monitoring using a hybrid swarm intelligence technique," *Smart. Mater. Struct.*, vol. 16, no. 6, pp. 2658–2672, 2007.
- [19] H. W. Park and H. Sohn, "Parameter estimation of the generalized extreme value distribution for structural health monitoring," *Probab. Eng. Mech.*, vol. 21, no. 4, pp. 366–376, 2006.
- [20] J. E. Mottershead and M. I. Friswell, "Model updating in structural dynamics: A survey," *J. Sound Vib.*, vol. 167, no. 2, pp. 347–375, 1993.
- [21] S. W. Doebling, C. R. Farrar, and M. B. Prime, "Summary review of vibration-based damage identification methods," *Shock Vib. Dig.*, vol. 30, no. 2, pp. 91–105, 1998.
- [22] J. R. Wu and Q. S. Li, "Structural parameter identification and damage detection for a steel structure using a two-stage finite element model updating method," *J. Constr. Steel Res.*, vol. 62, no. 3, pp. 231–239, 2006.
- [23] A. Teughels and G. DeRoeck, "Structural damage identification of the highway bridge Z24 by FE model updating," *J. Sound Vib.*, vol. 278, no. 3, pp. 589–610.
- [24] R. J. Allemang and D. L. Brown, "A correlation coefficient for modal vector analysis," in *Proc. 1st Int. Modal Anal. Conf. Exhibit*, Orlando, FL, 1982, pp. 110–116.
- [25] T. K. Caughey and M. E. J. O'Kelly, "Classical normal modes in damped linear dynamic systems," *J. Appl. Mech.*, vol. 32, no. 3, pp. 583–588, 1965, ASME.



Andrew T. Zimmerman (S'09) was born in Wheat Ridge, CO. He received the B.S. degree in engineering with a concentration in civil engineering from the Colorado School of Mines, Golden, in 2005, and the M.S.E. degree in civil engineering from the University of Michigan, Ann Arbor, in April 2006. He is currently working towards the Ph.D. degree in civil engineering and the M.S. degree in computer science at the University of Michigan.

His research interests include distributed computing architectures for in-network data processing in wireless sensing networks. He is supported by the U.S. Office of Naval Research through a National Defense Science and Engineering Graduate Fellowship (NDSEG).



Jerome P. Lynch (M'04) received the M.S. and Ph.D. degrees in civil and environmental engineering, in 1998 and 2002, respectively, and the M.S. degree in electrical engineering, in 2003, from Stanford University, Stanford.

Currently, he is an Assistant Professor of Civil and Environmental Engineering at the University of Michigan; he is also an Associate Professor with the Department of Electrical Engineering and Computer Science. His research interests include wireless structural monitoring, feedback control, and damage detection algorithms. Some of his more current research has been focused on the design of nanoengineered materials for smart structure applications including carbon nanotube-based thin-film wireless sensors for structural health monitoring.

Dr. Lynch was awarded the 2005 Office of Naval Research Young Investigator Award, the 2007 University of Michigan Henry Russel Award, and the 2008 College of Engineering (University of Michigan) 1938E Award.