20 Jul 2010

Cyber-Physical Codesign of Distributed Structural Health Monitoring with Wireless Sensor Networks

Gregory Hackmann

Weijun Guo

Guirong Yan
Missouri University of Science and Technology, yang@mst.edu

Chenyang Lu

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/civarc_enveng_facwork/2475

Recommended Citation

G. Hackmann et al., "Cyber-Physical Codesign of Distributed Structural Health Monitoring with Wireless Sensor Networks," *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS '10*, pp. 119 - 128, Association of Computing Machinery, Jul 2010.
The definitive version is available at https://doi.org/10.1145/1795194.1795211

Cyber-Physical Codesign of Distributed Structural Health Monitoring With Wireless Sensor Networks

Gregory Hackmann*, Weijun Guo*, Guirong Yan†, Chenyang Lu*, Shirley Dyke†
* Department of Computer Science and Engineering, Washington University in St. Louis
† School of Mechanical Engineering, Purdue University

ABSTRACT

Our deteriorating civil infrastructure faces the critical challenge of long-term structural health monitoring for damage detection and localization. In contrast to existing research that often separates the designs of wireless sensor networks and structural engineering algorithms, this paper proposes a cyber-physical co-design approach to structural health monitoring based on wireless sensor networks. Our approach closely integrates (1) flexibility-based damage localization methods that allow a tradeoff between the number of sensors and the resolution of damage localization, and (2) an energy-efficient, multi-level computing architecture specifically designed to leverage the multi-resolution feature of the flexibility-based approach. The proposed approach has been implemented on the Intel Imote2 platform. Experiments on a physical beam and simulations of a truss structure demonstrate the system's efficacy in damage localization and energy efficiency.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications; C.2.1 [**Network Architecture and Design**]: Wireless communication

1. INTRODUCTION

The deterioration of our civil infrastructure is a growing problem both in the US and around the world. For example, during their lifetimes, bridges suffer from environmental corrosion, persistent traffic and wind loading, extreme earthquake events, material aging, etc., which inevitably result in structural deficiencies. According to the American Society for Civil Engineers 2009 Report Card for America's Infrastructure, "more than 26%, or one in four, of the nation's bridges are either structurally deficient or functionally obsolete" [3]. Due to the expense of retrofitting a structure with a wired sensor infrastructure, most of these structures are not currently being continuously monitored.

ICCPS'10 April 13–15, 2010, Stockholm, Sweden

Recent years have seen growing interest in SHM based on wireless sensor networks (WSNs) due to their low installation and maintenance expenses. WSNs permit a dense deployment of measurement points on an existing structure, facilitating accurate and fault-tolerant damage identification techniques without the need to install a fixed wired infrastructure [20]. Indeed, numerous SHM systems have been proposed in literature which leverage WSNs to collect raw sensor data [6, 7, 16, 26]. These systems are generally designed to support traditional centralized SHM methods, with special consideration to the limited bandwidth and energy supplies that are not present under a traditional system of wired sensors.

However, by treating SHMs as a simple data collection devices for supporting centralized SHM methods, the resulting systems inherently suffer from high energy consumption and prolonged detection latencies. For example, a state-of-art system deployed at the Golden Gate Bridge required 9 hours to collect a single round of data from 64 sensors, resulting in a system lifetime of 10 weeks when using four 6V lantern batteries as a power source [21]. This system's high latency and relatively short lifetime arose from the fact that the underlying SHM method was designed separately from the WSN system. Specifically, the SHM method required the WSN to reliably deliver the entire raw sensor dataset to the base station for centralized processing, inherently placing a high network burden on the WSN system.

What is needed is a fundamentally different *cyber-physical* approach which considers both the constraints of the underlying WSN system (the *cyber* components) and the SHM requirements (the *physical* components) in its numerical approach. This can be achieved by leveraging the increasingly powerful processing capability of wireless sensor "motes" to partially process locally-collected data, extracting (and subsequently exchanging) only the important features relevant for SHM. Several recent studies demonstrate the potential for distributed SHM approaches to significantly reduce energy cost through localized data processing [5, 15, 20, 29].

In this paper, we present a hierarchical decentralized SHM system that implements a *flexibility-based* damage identification and localization method. In contrast to previous decentralized algorithms like DLAC [19], flexibility-based methods explicitly correlate data across multiple sensors, allowing them to accurately identify and localize damage on a wider range of structures. Our hierarchical system organizes nodes into clusters using a novel *multi-level search* approach that incrementally activates sensors in the damaged regions, allowing much of the network to remain asleep. We take advantage of the Intel Imote2 [12] platform's computational power to perform in-network processing wherever possible; thus, nodes further save energy and bandwidth by only transmitting the intermediate results related to the flexibility calculation.

In this paper, we make the following contributions. (1) We propose a *cyber-physical architecture* which efficiently maps flexibility-based damage identification and localization methods onto a distributed WSN. (2) We describe an implementation of this architecture on top of the TinyOS operating system [1] and ISHM services toolsuite [2]. (3) We evaluate this implementation on representative beam and simulated truss structures, demonstrating that our approach can successfully localize damage on both structures to the resolution of a single element. Latency and power consumption data collected during these experiments also demonstrate the energy efficiency of our approach.

The remainder of this paper is organized as follows. Section 2 describes related SHM systems in literature. In Section 3, we discuss the basic numerical methods used by our flexibility-based damage localization. Section 4 presents our mapping of these methods into an efficient distributed architecture. Section 5 describes our implementation of this distributed architecture on top of the Intel Imote2 platform. Section 6 provides an empirical evaluation of our system, demonstrating that it can efficiently localize damage to two representative structures. Finally, we conclude in Section 7.

2. RELATED WORK

A UC Berkeley project to monitor the Golden Gate Bridge [16] represents one of the first large-scale deployments of smart sensor networks for SHM purposes. Vibration data is collected and aggregated at a base station under a centralized network architecture, where frequency domain analysis is used to perform modal content extraction. However, it took nearly a full day to transmit sufficient data for such computations. Similarly, researchers at Clarkson University have implemented a wireless sensor system for modal identification of a full-scale bridge structure in New York [14]. Battery-powered wireless sensor nodes equipped with accelerometers and strain transducers are used, having a high wireless data transmission rate. The entire network is polled by a master computer that collects acceleration and strain data. Both modal identification and quantification of static responses are performed using a centralized network architecture. Wisden [26] provides services for reliable multi-hop transmission of raw sensor data, using run-length encoding to compress the data before transmission. These centralized approaches suffer from two fundamental limitations. First, data may only be collected from a limited number of nodes in a reasonable time frame, which would allow the system to only detect the most severe (and probably visually apparent) damages. Second, such systems are inadequate for timely detection of structural failures resulting from extreme events (e.g., earthquakes) due to the prolonged time needed for collecting and analyzing data.

BriMon [7] partially addresses the communication bottleneck by sampling data at 400 Hz and averaging this data over 40 Hz windows. The data resolution and network size (a maximum of 12 nodes per span) supported by BriMon may not be fine-grained enough for damage detection and localization on complex structures. A deployment in the Torre Aquila heritage building [6] uses lossless compression to deliver heterogeneous sensor data to sink node. The network burden of this deployment was eased by the specific kinds of data needed to monitor the building's health: only three acceleration sensors were required, while the environmental and deformation sensors produced only 1–10 readings every 10 minutes.

The above limitations motivate the need for a *co-design* approach which addresses both the SHM and WSN concerns in a holistic manner. An integral part of such a solution is the adoption of distributed SHM solutions [17, 24]. Researchers at the University of Illinois at Urbana-Champaign have experimentally validated a SHM system that employs a smart sensor network deployed on a scale three-dimensional truss model [20, 23]. Results demonstrate that the adopted SHM system is effective for damage identification and localization; however, significant communication is involved in performing data cross-correlation, which results in significant energy consumption.

Lynch et al. [25] implemented a low-cost and rapid-to-deploy wireless structural monitoring system on a long-span cable-stayed bridge in Taiwan. The full-scale test was conducted by collecting ambient vibration data of the bridge and analyzing it in situ by two modal identification methodologies, the stochastic subspace identification method (SSI) and frequency domain decomposition method (FDD). Modal ID results led to the determination of a total of 10 modal frequencies and corresponding mode shapes within a frequency range of 0–7 Hz. Lynch et al. [28] also implemented an automated modal identification by optimizing output-only modal methods (FDD with peak-picking) for a distributed wireless sensor network. The distributed implementation, tested in a balcony of a theater, used a parallel data processing and reduced communication scheme to ensure scalability and power efficiency in the WSN. In their implementation, three network topologies are proposed to yield a two-node based data sharing chain. This implies the partial mode shape identified from each pair of nodes has to be recombined to recreate the complete mode shape necessary for damage detection. However, this strategy would potentially amplify the recombination error, if any one of the sensor nodes is unreliable.

In our own prior work, we designed and experimentally validated a distributed approach based on the Damage Location Assurance Criterion (DLAC) method [5, 15]. However, DLAC has several intrinsic limitations in its SHM capabilities. First, DLAC requires the user to pre-specify the damage patterns that it should try to identify and localize. Second, DLAC is not sensitive to small damages in a structure because it only monitors the structure's natural frequencies, and because it does not correlate readings across sensors. Finally, DLAC can only properly localize damage to asymmetric structures. These limitations occur because there is effectively no collaboration among sensors under DLAC: each sensor's readings are handled independently, and are only combined at the very end to compensate for node failures and sensor noise. Alleviating these limitations requires a fundamentally new architecture which leverages collaboration among sensors to enhance the damage identification and localization results.

3. DAMAGE LOCALIZATION APPROACH

In this section, we introduce the *physical* (structural engineering) aspects of our decentralized damage localization

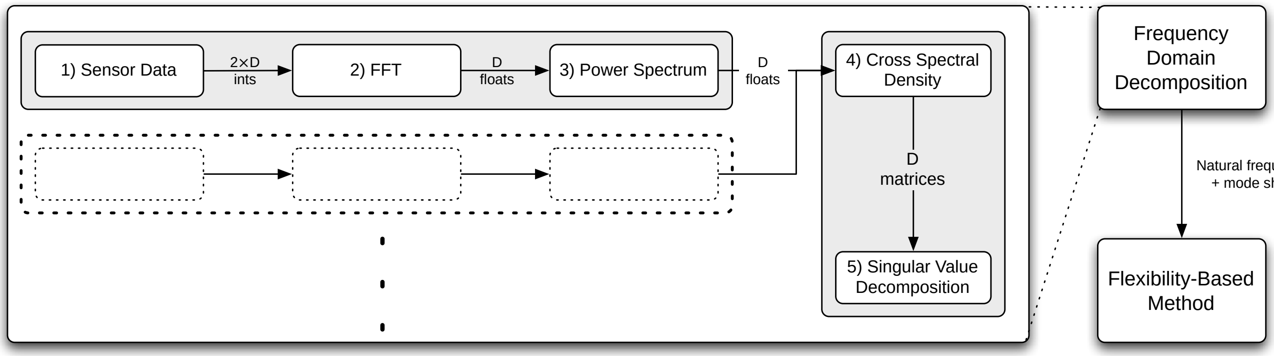


Figure 1: The data flow of a traditional flexibility-based method

system. Our system is based on a family of damage localization techniques collectively known as *flexibility based* algorithms. The intuition behind these methods is that structures will flex slightly when a force is applied, as shown in Figure 2. As a structure weakens, its stiffness decreases, and thus its flexibility changes. Changes in structural flexibility over a structure's lifetime can be used to identify and localize damage [22]. We have chosen this family of methods because they address the aforementioned limitations in DLAC. Moreover, as we discuss in Section 4, they enable us to develop a multi-level system architecture specifically optimized for this approach.

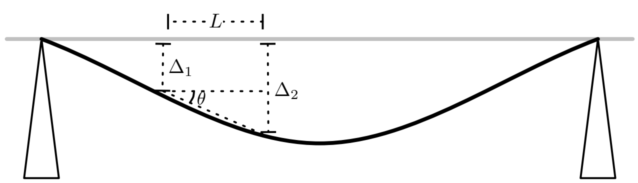


Figure 2: Structural deflection

We will provide here a brief background on two particular flexibility-based methods used within our decentralized system. While flexibility-based methods are well-known in structural engineering literature, the existing research generally deals with algorithmic issues (i.e., selecting the best numerical methods for damage identification and localization) rather than efficiently deploying these methods on a distributed architecture for WSNs. We will focus here on the details of these algorithms that are most relevant to our system design; more mathematical details can be found in [13, 27].

Flexibility-based methods are executed in two stages. When the system is first turned on, a baseline structural modal identification is performed. The sensors simultaneously collect vibration data. Multiple sensors' data are correlated to identify the structure's *modal parameters* (natural frequencies and mode shapes). The modal parameters are then further processed to compute the structure's *flexibility matrix*.

Online, the data collection and processing phases above are repeated, and the base station produces a new flexibility matrix. By subtracting the new flexibility matrix from the stored one, the base station can determine if the structure is damaged (and if so, identify the damaged region).

We will now summarize the main components of flexibility-based methods, as shown in Figure 1. The structure's modal parameters are identified using Frequency Domain Decomposition (FDD), an existing structural engineering technique which can be decomposed into several stages. Traditionally, FDD is executed as follows. (1) All the nodes in a cluster simultaneously collect D vibration samples using their onboard accelerometers. The size of D depends on structural properties (like its complexity and material) as well as the modes we are interested in, and is typically hundreds or thousands of samples. (2–3) Each node independently performs an FFT and power spectrum analysis on the vibration data, transforming it into magnitudes in the frequency domain. (4) D magnitudes collected from each node are correlated to compute a Cross Spectral Density (CSD) matrix. (5) A Singular Value Decomposition (SVD) is performed on the CSD matrix at each of D discrete frequencies. The singular value in each singular value matrix is collected to form a vector, and the structure's P lowest natural frequencies are identified as the peaks in this vector. The mode shapes corresponding to the natural frequencies can be estimated from the first column of the corresponding left SVD matrix.

The FDD output is then input into a flexibility-based method. Our system uses two specific flexibility-based methods: the Angles-Between-String-and-Horizon flexibility-based method (ASHFM) [13] and the Axial Strain flexibility-based method (ASFM) [27]. We are particularly interested in these two methods because they can localize damage down to a resolution of a specific element on beam-like and truss-like structures, respectively. Most other flexibility-based methods localize damage only to less specific regions of the structure, while [4] achieves similar damage localization resolution at a much higher computational cost.

ASHFM measures the flexibility of a beam-like structure as the angle θ in Figure 2. The FDD output data is used to calculate θ at each of the structure's modes, producing a flexibility matrix F. The difference $\Delta F = \left|F^b - F\right|$ can be used to localize the changes in flexibility to locations along the beam, where F^b is the flexibility matrix calculated during the baseline phase. Specifically, the maximum absolute values of the components in each column or diagonal of ΔF are extracted as *damage indicators*. When a location on the structure is damaged, it will appear as as "step and jump" in the plot of damage indicators, as shown in Figure 3. In this example, the large jumps in the ASHF damage indicator surrounding the shaded points reflect damage in the

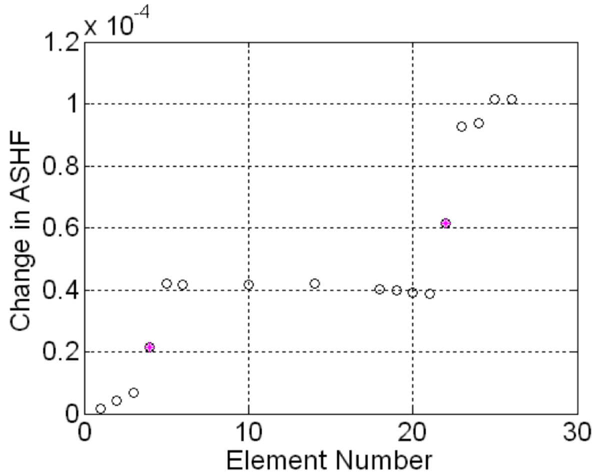


Figure 3: Example ASHFM damage indicator output; shaded dots correspond to damaged elements

corresponding structural elements.

ASFM achieves damage localization at a similar resolution to ASHFM, except on truss-like rather than beam-like structures. At a high level, ASFM localizes damage in much the same way as ASHFM. The main differences are that ASFM requires vibration data taken simultaneously in multiple directions, and that the two methods have different formulations for computing F.

4. DISTRIBUTED ARCHITECTURE

The numerical methods discussed above have been designed with centralized networks in mind, where sensors are used as simple data collection devices that can stream large data sets to a central server over a wired backbone. Under a WSN, this approach is inappropriate because of the nodes' limited network and energy resources. However, in order to design an efficient decentralized architecture, we can leverage a particularly powerful feature of these flexibility-based methods. Specifically, they enable a tradeoff between energy consumption and localization resolution: the more nodes that are activated, the finer-grained the damage localization.

We leverage this feature to construct an energy-efficient, *multi-level* damage localization system which selectively activates additional sensors at each level in order to more precisely localize structural damage. In the common case that the structure is not damaged at all, only a minimal subset of nodes are enabled, considerably reducing the system's energy and bandwidth consumption. This approach naturally maps to a hierarchical, cluster-based distributed network architecture. In addition, to promote a more efficient mapping onto our distributed system, we leverage an existing *peak picking* technique to reduce the data flow among sensors participating in each cluster.

4.1 Multi-Level Damage Localization

Although adding more sensors can improve a flexibility-based method's localization results, only a handful of sensors are needed to accurately *identify* damage. In the first stage of the multi-level search, this minimal number of sensors are enabled, forming a single cluster. Damage identification and localization is performed using this small subset of sensors. In the common case that no damage is identified, the search ends and all the nodes return to sleep.

In the event that damage is identified, the flexibility-based method will also output coarse-grained damage localization. For example, ASHFM will identify two adjacent sensors surrounding each damage location on the structure. In the next round of the multi-level search, the system activates additional sensors in the region of interest and repeats the entire procedure, including collecting new vibration data. This second round subsequently localizes the damage to a smaller region than the first round. The system may repeat this drill-down procedure to achieve even finer grained results until the desired resolution is reached.

The key feature of this approach is that it does not activate the entire sensor network at once. Instead, relatively few sensors are used to identified damage; and when damage is identified, only those sensors in the area of interest are incrementally added to the search. As a result, many nodes are able to remain asleep for part or all of the multi-level search. This approach will also scale to larger structures, since the cost of the search is no longer proportional to the size of the structure. As we discuss in Section 5.2, the reduced energy burden can also be distributed across the network by activating different subsets of the network at different times.

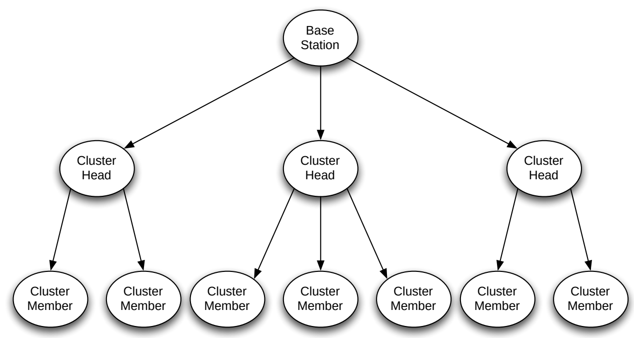


Figure 4: Sensor Roles in the System

4.2 Network Hierarchy

Once the nodes participating in this multi-level search are selected, they are each assigned one of three different roles: *cluster member*, *cluster head*, and *base station*. A node's role determines what data it handles as well as its level in the network hierarchy, as shown in Figure 4. To allow the system to better scale to large structures, the nodes may be organized into multiple independent clusters. Each cluster operates as an independent unit, with the cluster head coordinating nodes within its cluster and ultimately transmitting the cluster's (relatively small) mode shape data to the base station for final processing.

Based on these roles, the system operates as follows. The cluster members collect raw vibration samples from their onboard accelerometers. They then carry out an FFT to transform the vibration response into frequency domain data, followed by a power spectrum analysis.

The cluster head nodes aggregate the extracted power spectrum data from the cluster members beneath them in the hierarchy. There, the CSD and SVD are carried out to extract the structure's mode shape vector.

The cluster heads then transmit the mode shapes to a single base station node, which calculates the structure's flexibility. The flexibility is then used to identify and localize any structural damage.

4.3 Enhanced FDD

Efficiently implementing this architecture for a flexibility-based system is challenging because there are no obviously "best" places to introduce network communication: the CSD and SVD routines are necessarily computed on a single node with access to all the other cluster members' data, and the prior steps all have very large outputs (hundreds or thousands of points). In order to achieve truly energy-efficient behavior, we must optimize the FDD algorithm's data flow to promote an efficient mapping onto wireless sensor networks.

We leverage an optimization proposed in [25,29] that adds a new *peak picking* stage to FDD. To illustrate how this optimization works, we note that most of the computations in the FDD routine do not contribute to the final results. As described in Section 3, the CSD step normally requires the cluster head to pool D data points from each of its cluster members. This data is processed into D CSD matrices, which the SVD routine further processes into D outputs and discards all but the P corresponding to the structure's natural frequencies (note that $P \ll D$). A key observation about this procedure is that the ith CSD matrix is only constructed using the ith power spectrum data point from each cluster member. Moreover, only the P CSD matrices corresponding to the structural's natural frequencies contribute to the FDD stage's final output.

The peak picking routine allows each node to independently identify these P natural frequencies solely from local data. Hence, only those P relevant data points are passed onto the CSD stage, which in turn passes only the relevant P matrices onto the SVD stage. In this way, both the computational and communication cost of identifying modal parameters are reduced considerably. We emphasize that the data which the nodes withheld would not have contributed to the final flexibility computation. Hence, even though significantly fewer data are transmitted and processed, there is no loss in damage identification or localization performance.

5. IMPLEMENTATION

We have built a proof-of-concept implementation of our system on top of the Imote2 [12] sensor platform using the TinyOS operating system [1]. Our implementation utilizes the ISHM services toolsuite [2] developed by the Illinois Structural Health Monitoring Project (ISHMP), which provides subsystems for sensor data acquisition, reliable data transmission, and time synchronization based on the FTSP protocol [18].

5.1 Hardware Platform

The Imote2 is an advanced wireless sensor node platform built around the low-power PXA271 XScale processor and 802.15.4-compliant radio hardware (Chipcon CC2420) with a built-in 2.4GHz antenna. While our proposed approach to SHM is not inherently tied to a particular platform, the Imote2 offers several salient improvements over previous generation WSN platforms that are particularly useful for our application.

First and foremost, the PXA271 CPU has 256 KB of embedded SRAM and can address 32 MB of on-board SDRAM, providing copious space for computations. In contrast, platforms such as the TelosB [11] and MICA [9,10] family have access to only 4 – 10 KB of RAM, which would not even be enough to store the entire raw sensor reading dataset. Accordingly, such platforms would either be restricted to purely streaming computations or would have to swap data in and out of onboard flash, a potentially expensive operation.

Second, the PXA271 CPU can be dynamically clocked from 13 – 416 MHz, allowing us to increase the CPU speed when needed (e.g., while collecting high-resolution sensor data) and decrease its speed at other times to save energy. Third, the Imote2 is a modular stackable platform which can be expanded with extension boards to customize the system to a specific application. The ITS400 sensor board provides an add-on accelerometer which we have confirmed to be sufficiently accurate for our SHM application. Fourth, the Imote2 is equipped with 32 MB of flash memory, which allows us to deploy the entire application on all nodes in the network. We take advantage of this capability to dynamically reconfigure the network without having to re-flash the nodes with new software, as discussed below.

5.2 Software Platform

As described above, our system is implemented in the nesC programming language on top of the TinyOS 1.1 operating system. Our software package uses several major components from UIUC's ISHM toolsuite. Namely, we use ISHM's `ReliableComm` components to perform reliable (ARQ) communication among sensor motes and the `Synchronization` components to start data collection simultaneously across all the participating motes. Moreover, we used the included `DistributedDataAcquireApp` as the basis for our data collection routines.

As discussed in Section 4, we have implemented a multi-level search technique that first activates a minimal number of sensors to identify and localize damage at a coarse-grained resolution. If damage is identified, then additional nodes are activated in the affected region in order to achieve a more fine-grained localization. In our current implementation, we employ a two-stage search; i.e., a few nodes are activated in the first stage, and all the nodes between the affected nodes are activated in the second stage. For larger structures, more stages may be added based on the relative sampling, communication, and computation costs.

Since different roles carry different sensing, computation, and communication costs, we do not want fix the roles of cluster member and cluster head to a single configuration. Hence, the base station node dynamically assigns the roles at the beginning of each damage identification round. In the interest of balancing energy consumption across the network, the base station currently assigns nodes in the first stage in a simple round-robin fashion. (If damage is identified, i.e., a second stage is needed, then the only change to the roles are that additional cluster members are added in the region of interest.)

Figure 5 illustrates the network configuration process. At the start of the procedure, the base station nodes disseminates a configuration packet to all other nodes in the network. This packet includes information about the cluster division and the assignment of roles within each cluster. Because the Imote2 platform is equipped with copious flash

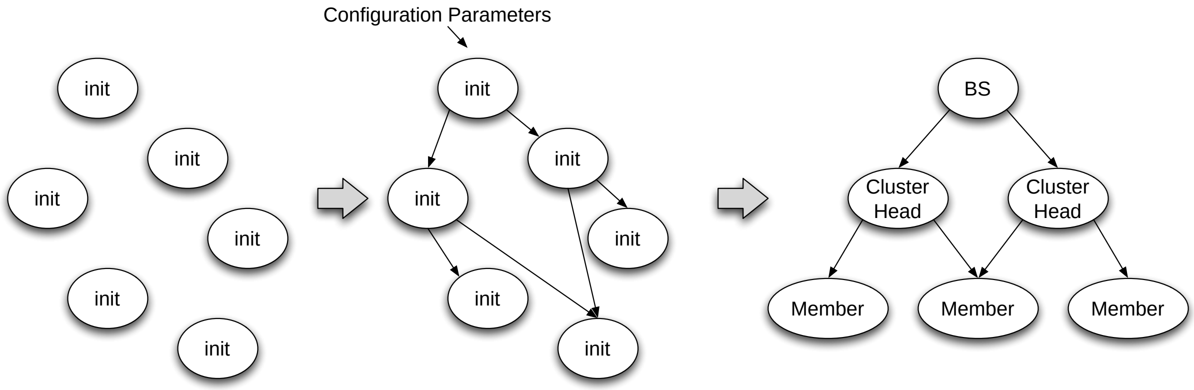


Figure 5: Network Configuration Process

memory, all nodes are programmed with the code for all roles. Thus, nodes can respond to a reconfiguration message by simply changing their configuration parameters in RAM.

After cluster members complete their computations and deliver their data to the cluster heads, they are put into deep sleep mode to save energy. Similarly, after the cluster heads finish their mode shape computations and transmit the results to the base station, they are put into deep sleep mode. If the base station identifies damage, then additional damage localization stages will be triggered as described above. Otherwise, the entire network will remain asleep until the next scheduled damage identification round.

We use the ISHM toolsuite's `SyncC` component to time-synchronize the network at the start of each round. In addition to ensuring that the collected samples are time synchronized, we take advantage of the synchronization to hierarchically assign a TDMA transmission schedule. Thus, we save energy by reducing transmission failures due to packet collisions.

6. EVALUATION

To validate our system, we implemented and deployed our multi-level damage localization system on two representative structures. We will first briefly discuss our deployment of the ASHFM-based approach on a steel cantilever beam, the relatively simpler of the two structures. We will then describe a deployment of our system on a simulated steel truss structure, which presents a more challenging scenario for damage localization due to its structural complexity. Experimental results demonstrate that our system is able to accurately localize damage at the member-level to both structures. Moreover, latency and energy consumption data collected during the truss structure experiment illustrate the efficiency of our decentralized approach.

6.1 Cantilever Beam

Our first set of experiments were performed on a steel cantilever beam at Washington University's Structural Control and Earthquake Engineering Lab. The beam is 2.75 m long, 7.6 cm wide, and 0.6 cm thick and fixed to the ground to approximate a cantilever support.

For these experiments, we deployed eight Imote2 motes with sensorboards directly on the beam, as well as a gateway mote tethered to a base station PC via a USB interface board. The sensors were distributed along the beam as shown in Figure 6. Because only these nine motes were available at the time of the experiment, we deployed them non-uniformly with increased sensor density around the area of damage. Off-line analysis showed that, had additional sensors been deployed to achieve a uniform distribution, they would have remained asleep as part of the multiresolution search procedure described in Section 4.1.

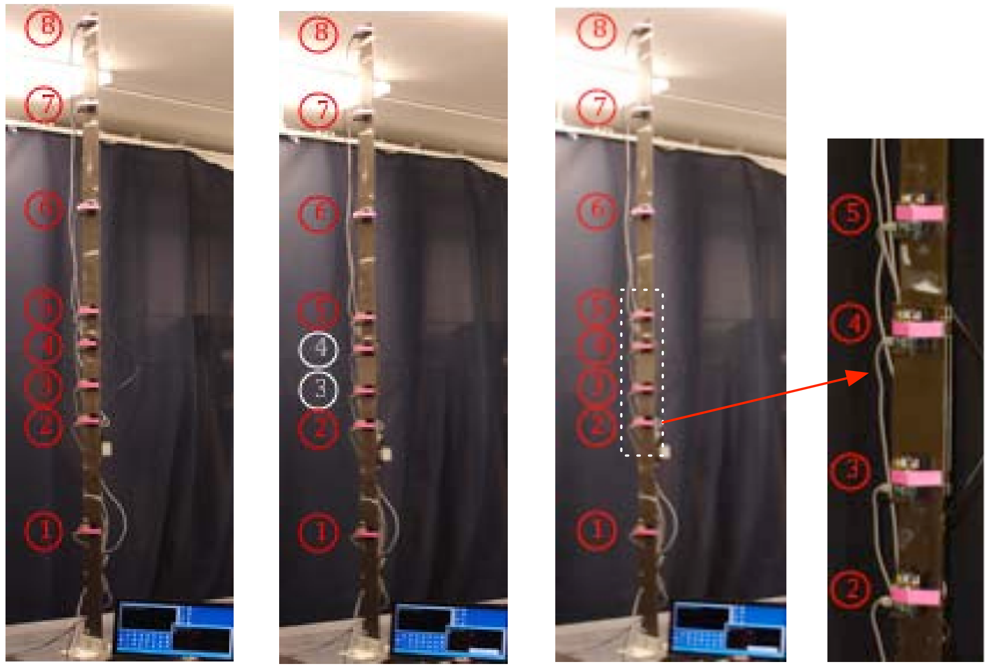
To obtain the structure's baseline modal parameters, we excited the beam along the weak axis of bending using an impact. All of the eight motes attached to the structure recorded vibration data using a sampling frequency of 280 Hz, a record length of 7168 points, and an FFT size of 2048 points. The motes calculated the structure's modal parameters (using the distributed algorithm discussed in Section 4) which were collected at the base station to compute the structure's flexibility. For the purposes of validation and offline analysis, we also collected the raw vibration data and intermediate results at the base station, although as noted above only the modal parameters are required to monitor the structure's health.

We then repeated the procedure, simulating damage by attaching a pair of thin, symmetric steel plates to element 4. For the first level of our multi-level search, six sensors were activated uniformly across the truss structure, as shown in Figure 6(b). (Again, we note that even had more sensors been deployed on the structure, only these six would have been activated.) The base station collected the new modal parameters from the motes, producing the damage indicators shown in Figure 7(a). These results indicate that the structure is damaged, with a coarse-grained location somewhere between sensors 2 and 5.

Accordingly, the base station automatically invoked a second stage of damage localization, activating two additional sensors in the region of damage as shown in Figure 6(c). The finer-grained damage indicators shown in Figure 7(b) indicate damage specifically in element 4, which is consistent with the position of the steel plates.

6.2 Truss

Our second set of experiments involve simulated sensor data from a 5.6 m steel truss structure [8] at the Smart Structure Technology Laboratory (SSTL) at the University of Illinois at Urbana-Champaign. In order to accommodate the truss's increased structural complexity, we increased the



(a) baseline test

(b) level-1 identification

(c) level-2 damage localization

Figure 6: The cantilever beam and sensor placements for each stage of the experiment

sampling frequency to 560 Hz, the record length to 18,432 data points, and the FFT size to 4096 points. Unfortunately, a deadlocking bug in the ITS400 sensorboard subsystem prevented us from collecting sufficient vibration data on the real truss to perform our experiments. Instead, we produced two sets of simulated data traces using a finite element model of the truss in MATLAB, with additional measurement noise added to simulate noisy sensor readings. The first set represents the truss in its intact case, providing a baseline flexibility measurement. The second set was generated with simulated damage to three members of the left side of the truss and four members to the right side of the truss.

For the truss experiments, we wished to evaluate our system's damage localization performance as well as its energy consumption. Thus, we made two augmentations to our nesC code for this set of experiments. First, we added a "fake" sensor driver which replayed sensor data traces from the motes' flash memory, allowing us to inject our simulated traces into live experiments. Second, we collected timestamping data at key points in our code in order to measure the latency and energy consumption of each major component of our system.

6.2.1 Damage Localization

To evaluate our system's damage localization performance, we performed three different experiments with nine Imote2 motes. In our first configuration, we injected simulated sensor data collected at uniform points along the truss's length. This configuration represents the "level 1" damage identification phase. The damage indicators computed during this experiment are plotted in Figure 8(a). Based on the step-and-jump surrounding bays 4 and 10 (shown as the two peaks in Figure 8(a)), our system correctly identified damage on both halves of the truss.

In our second and third configurations, we used simulated sensor data collected at a greater density on the truss's left and right halves, respectively. This configuration represents the more fine-grained "level 2" damage localization phase. As shown in Figures 8(b) and 8(c), our system indeed correctly localized the three damaged members on the left side of the structure and the four damaged members on the right side.

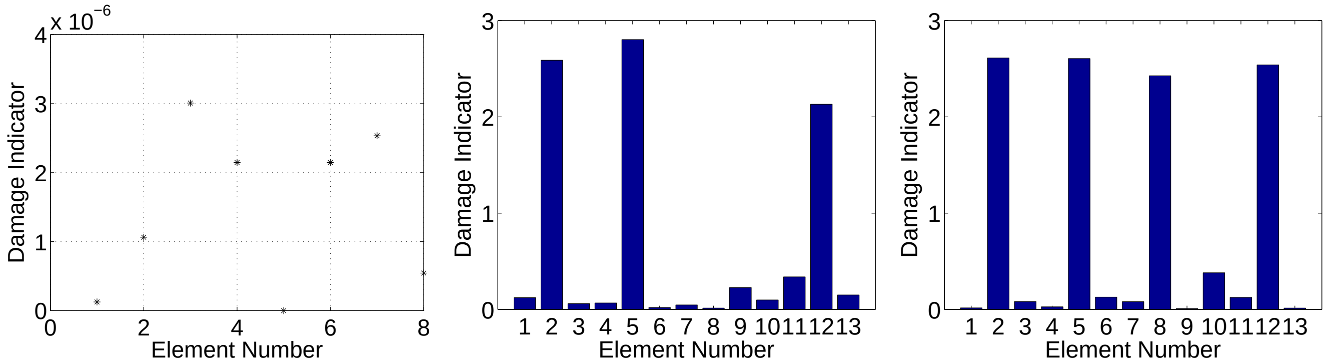
6.2.2 Energy Consumption

During the experiments described above, we collected timestamp data from the motes in order to directly measure the latency of each major stage in the experiment. We also performed a separate set of experiments to measure the latency of time-synchronizing the motes and collecting 18,432 data samples (since the previous truss experiments used replayed data traces). Tables 1 and 2 present the average latencies for the cluster member and cluster head nodes, respectively. Offline, we measured the power draw of each stage using an oscilloscope, which we used to estimate the total energy consumption of each stage in the experiment.

State	Latency (s)	Energy (J)
Synchronization	30.00	12.06
Sensing	53.80	22.96
Compute FDD	21.47	9.28
Transmit FDD	0.21	0.08

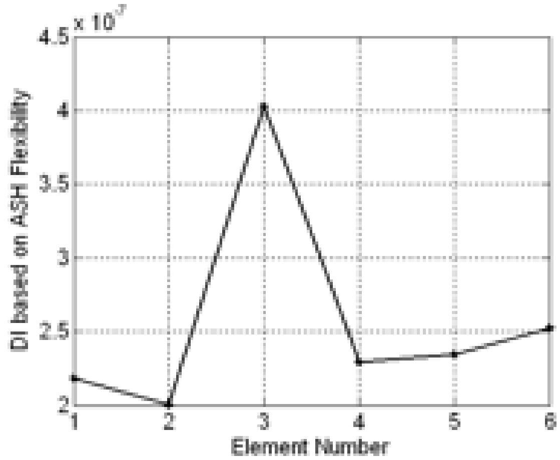
Table 1: Mean latency and energy cost at cluster member

Several important observations can be made from this data. First, our decentralized architecture is indeed effec-

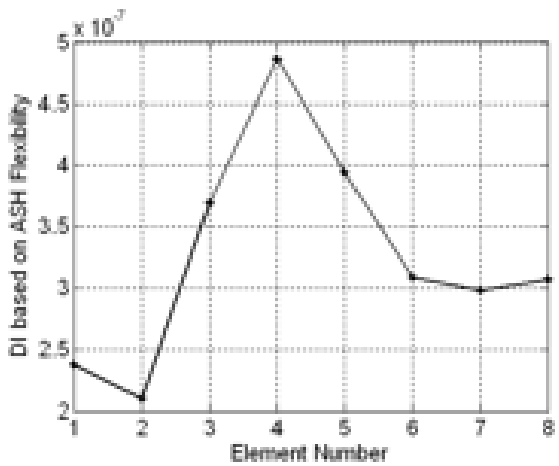


(a) level-1 identification (b) level-2 localization (left) (c) level-2 localization (right)

Figure 8: Damage localization results on the simulated truss



(a) damage indicators when six sensors are activated



(b) damage indicators when eight sensors are activated

Figure 7: Damage localization results on the cantilever beam

State	Latency (s)	Energy (J)
Synchronization	40.35	16.23
Sensing	49.68	21.20
Compute own FDD	18.19	7.86
Receive other FDD	0.56	0.23
Compute mode shapes	1.52	0.66
Transmit mode shapes	1.35	0.53

Table 2: Mean latency and energy cost at cluster head

tive at dramatically reducing the amount of bandwidth and energy consumed in exchanging data among nodes. Our decentralized architecture spends an average of 0.21 s per cluster member exchanging FDD results, plus an average of 1.35 s per cluster head transmitting the mode shape results to the base station. In contrast, based on our prior work [15], we estimate that it would have taken 87 s per sensor to reliably transmit the 18,432 raw sensor readings to the base station for centralized processing.

Second, our efficient architecture incurs relatively little overhead on the Imote2 hardware. On the cluster member nodes, as much as 79.4% of the latency and 78.9% of the energy consumption can be attributed to synchronizing the nodes and collecting data. Only 21.1% of the energy consumption represents reducible overhead. The cluster head nodes incur similarly low overheads, with only 20.4% of the latency and 19.1% of the energy consumption attributable to processing and data transmission.

Third, this low overhead leads to low total energy consumption in absolute terms. On average, the cluster member and cluster head nodes consume a total of 44.4 J and 46.7 J, respectively. A typical power supply of 3x 1.5V, 1250 mAh AAA batteries delivers a theoretical energy supply of 20,250 J. Thus, with proper duty cycling, we anticipate that each node could perform damage localization hundreds of times before depleting its energy supply.

7. CONCLUSION

Structural health monitoring of civil infrastructure represents an important application domain of cyber-physical systems. We propose a novel cyber-physical co-design approach to structural health monitoring based on wireless sensor networks. Our distributed structural health monitoring system integrates (1) flexibility-based structural engineering methods that can localize damages at different resolution and costs, and (2) an efficient, *multi-level* computing architecture that leverage on the multi-resolution feature of flexibility-based methods. A key feature of our approach is that it selectively activates nodes in the damaged region in order to achieve fine-grained localization damage localization while allowing many of the nodes to remain asleep. We have implemented our approach on the Intel Imote2 hardware platform and the TinyOS operating system. Experimental results show that our system is able to localize damage to the resolution of a single element on a representative physical beam and simulated truss structures, including multiple simultaneous damages on the latter. We also demonstrate the energy efficiency of this approach through latency and energy consumption measurements. Our results illustrate the promise of cyber-physical approach which consider both the architecture of the cyber (wireless sensor network) system and the characteristics of the physical (structural engineering) methods.

Acknowledgments

This work is supported by NSF NeTS-NOSS Grant CNS-0627126 and CRI Grant CNS-0708460. We would like to thank Prof. B.F. Spencer and Shin Ae Jang for their help with our experiments related to the truss.

8. REFERENCES

[1] `http://www.tinyos.net/`.
[2] `http://shm.cs.uiuc.edu/`.
[3] American Society of Civil Engineering. 2009 report card for America's infrastructure. `http://www.asce.org/reportcard/2009/`.
[4] D. Bernal and B. Gunes. Flexibility based approach for damage characterization: Benchmark application. *Journal of Engineering Mechanics*, 130(1):61–70, January 2004.
[5] N. Castaneda, F. Sun, S. Dyke, C. Lu, A. Hope, and T. Nagayama. Implementation of a correlation-based decentralized damage detection method using wireless sensors. In *ASEM Conference*, 2008.
[6] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon. Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment. In *IPSN*, 2009.
[7] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar. BriMon: a sensor network system for railway bridge monitoring. In *MobiSys*, 2008.
[8] E. Clayton. Development of an experimental model for the study of infrastructure preservation. In *National Conference on Undergraduate Research*, 2002.
[9] Crossbow Technology. MICA2 wireless measurement system. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf`.
[10] Crossbow Technology. MICAz wireless measurement system. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf`.
[11] Crossbow Technology. TelosB mote platform. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf`.
[12] Crossbow Technology, Inc. *Imote2 Hardware Reference Manual*, 2007.
[13] Z. Duan, G. Yan, and J. Ou. Structural damage detection using the angle-between-string-and-horizon flexibility (under review). *Structural Engineering and Mechanics*.
[14] M. Gangone, M. Whelan, K. Janoyan, K. Cross, and R. Jha. Performance monitoring of a bridge superstructure using a dense wireless sensor network. In *International Workshop on Structural Health Monitoring, Stanford, California*, 2007.
[15] G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dyke. A holistic approach to decentralized structural damage localization using wireless sensor networks. In *RTSS*, 2008.
[16] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN*, 2007.
[17] J. Lynch and K. Loh. A summary review of wireless sensors and sensor networks for structural health monitoring. *Shock and Vibration Digest*, 38(2):91–128, 2006.
[18] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi. The flooding time synchronization protocol. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 39–49, New York, NY, USA, 2004. ACM.
[19] A. Messina, I. A. Jones, and E. J. Williams. Damage detection and localization using natural frequency changes. In *Conference on Identification in Engineering Systems, U.K.*, 1996.
[20] T. Nagayama. *Structural Health Monitoring Using Smart Sensors*. PhD thesis, University of Illinois at Urbana-Champaign, 2007.
[21] S. N. Pakzad, G. L. Fenves, S. Kim, and D. E. Culler. Design and implementation of scalable wireless sensor network for structural monitoring. *ASCE Journal of Infrastructure Engineering*, 14(1):89–101, March 2008.
[22] A. Pandey and M. Biswas. Experimental verification of flexibility difference method for locating damage in structures. *Journal of Sound and Vibration*, (184), 1995.
[23] W. Song, S. J. Dyke, G. J. Yun, and T. Harmon. Improved damage localization and quantification using subset selection (under review). *Journal of Engineering Mechanics*.
[24] B. Spencer and T. Nagayama. Smart sensor technology: a new paradigm for structural health monitoring. In *Asia-Pacific Workshop on Structural health Monitoring, Yokohama, Japan*, 2006.
[25] J.-H. Weng, C.-H. Loh, J. P. Lynch, K.-C. Lu, P.-Y. Lin, and Y. Wang. Output-only modal identification of a cable-stayed bridge using wireless monitoring systems. *Engineering Structures*, 30(7):1820–1830, 2008.
[26] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *SenSys*, 2004.
[27] G. Yan, Z. Duan, and J. Ou. An axial strain flexibility for damage detection of truss structure. *International*

Journal of Smart Structures and Systems.

[28] G. J. Yun, K. A. Ogorzalek, S. J. Dyke, and W. Song. A two-stage damage detection approach based on subset selection and genetic algorithms (under review). *Smart Structures and Systems*.

[29] A. T. Zimmerman, M. Shiraishi, R. A. Swartz, and J. P. Lynch. Automated modal parameter estimation by parallel processing within wireless monitoring systems. *Journal of Infrastructure Systems*, 14(1):102–113, March 2008.