1-1-2009

Design And Evaluation of Flexibility-Based Structural Damage Localization Using Wireless Sensor Networks

Weijun Guo

WASHINGTON UNIVERSITY IN ST. LOUIS
School of Engineering and Applied Science
Department of Computer Science and Engineering

Thesis Examination Committee:
Chenyang Lu, Chair
Shirley Dyke
Christopher Gill
Viktor Gruev

DESIGN AND EVALUATION OF FLEXIBILITY-BASED STRUCTURAL DAMAGE LOCALIZATION USING WIRELESS SENSOR NETWORKS

by

Weijun Guo

A thesis presented to the School of Engineering
of Washington University in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

December 2009
Saint Louis, Missouri

ABSTRACT OF THE THESIS

DESIGN AND EVALUATION OF FLEXIBILITY-BASED STRUCTURAL DAMAGE LOCALIZATION USING WIRELESS SENSOR NETWORKS

by

Weijun Guo

Master of Science in Computer Science

Washington University in St. Louis, 2009

Research Advisor: Professor Chenyang Lu

The health of civil structures is very important and sometimes life-critical. While there are different ways to monitor their health, wireless sensor network (WSN) has the advantage of easy deployment and low cost, which make it feasible for most structures. We designed and implemented a system to localize damages on structures with a WSN by detecting the change in structure flexibility. This method has been validated to work well on bridges like a cantilever beam and a truss. It is also possible to be extended to other type of structures. Different from other systems, in network data processing was applied to lower the bandwidth requirement of large amount of raw sensing data. Only the intermediate computation results, that capture the flexibility related information, were transmitted back to the base station. We also divide the detection and localization into multiple levels. Lower level acts as the sentinel to detect the existence of damage; and higher levels, which consume more energy, are then triggered when necessary to get a higher resolution of localization. This design helps to further extend the lifetime of the system.

Acknowledgments

I would like to take this chance to thank my advisor Professor Chenyang Lu and Professor Shirley Dyke for giving me the opportunity to join this project and all the good advices and support. I also want to thank Professor Christopher Gill and Professor Viktor Gruev for joining the defense committee. Dr. Guirong Yan gave me much help from the civil engineering perspective, without her corporation I won't be able to make the system work on real structures. The WSN group in the CSE department at WUSTL offered many good advices on the implementation; I would like to thank them for their kindness, especially Greg Hackmann, Chien-Liang Fok and Octav Chipara, who are all experts in sensor network technologies.

I want to thank the department of Computer Science and Engineering for providing the scholarship for my program. It was a wonderful year for me to study here, learning from the distinguished faculties, listening to their lectures, attending colloquial talks and extending my views of varieties of interesting research areas.

I would like to also thank our Chinese students in the department, who are always willing to help each other. They made life here more easy and full of fun even when having tons of course work and duties. I will always cherish the good memories and friendship in my heart.

Weijun Guo

Washington University in St. Louis

December 2009

Dedicated to my parents and my wife Qing Kuang.

Mother and Father, I know you missed me a lot during my study here. Thanks for your support to my decision. My wife followed me here sacrificing her own career in China, I really appreciate that. I will always love you forever.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless sensors provide a natural connection between computational and physical elements, which enables people to be more aware of the environment and make quick responses to changes around them. Wireless sensor networks can continuously working in many harsh environments where humans will not or cannot stay all the time or at least not economic to do so. Such environments can be some wild habitats that might better be avoided of disturbances from human activities but attract much research interest from scientists. They also can be some industry facilities critical to the operation of business, such as computer servers in data centers, or civil structures that thousands of people rely on every day. It is really nice to have some "sentinels" out there watching for us, keep an eye on the temperature, humidity, light, vibration, sound, infrared signal, and geographical location measurements and acknowledge us when attention should be paid to. In reality, many of such WSN based systems have been designed and deployed to carry out such environment monitoring tasks, such as the Great Duck Island project by Berkley [1], the Redwood forest monitoring [2], the industrial infrastructure monitoring [3], the volcano monitoring [4], environmental monitoring [5] and the structure health monitoring [6].

Most of current civil structures are not monitored at this time due to the high cost of system deployment especially wired sensor systems. The collapse of the I-35W highway bridge over the Mississippi River in Minneapolis (Minnesota, US, August 2007) further underscores the need for reliable and robust structural health monitoring (SHM). With low installation and maintenance expenses, structural health monitoring and damage detection based on wireless sensor networks (WSNs) has attracted wide attention. Using a WSN, a dense deployment of measurement points on a structure is possible, which facilitates accurate and fault tolerant damage detection techniques [7]. There are several deployments for civil structure monitoring using WSNs. [8] used Damage Localization Assurance Criterion (DLAC) algorithm to analyze and locate damage; [9] used strain and acceleration readings to monitor Torre Aquila, a medieval tower in Trento (Italy); and in [10] a sensor network was deployed and tested on the Golden Gate Bridge to monitoring its health using ambient vibration data.

However, most current WSNs don't have much computation on the sensor nodes in the network based on the limitation of computational capability of sensor platforms. Table 1.1 gives the CPU and memory parameters of 3 commonly used sensor platforms. Many sensors were designed to use cheap microcontrollers so as to lower the cost of each sensor, and thus left little computation power with the nodes. The choosing of microcontroller and the memory limits the complexity of tasks can be carried out on sensor nodes. One platform, Imote2, is an exceptional among them in that it has a powerful XScale processor and larger memory space, which allows decent computation requirement.

Table 1.1 Three Sensor Platforms' Parameters

platform		MicaZ	Telosb/Moteiv	Imote2
CPU	**Microcontroller**	ATmega128L	TI MSP430	PXA271
	frequency	8MHz	8MHz	13-416MHz
Memory	**ROM**	128KB	48KB	256KB
	RAM	4KB	10KB	32MB
	Flash	512KB	1M	32MB

Many WSNs are based on a centralized design, where a base station collects raw data from all other sensors and acts as a gateway between WSN and PC or the Internet, since most of the time the objects or the places being monitored are far away from urban life and analysis are carried out on remote locations. In these cases, WSNs are mainly the data source of the environment and care more about data collection, transmission, and providing availability to the remote places. There are not many feedbacks in such systems.

While with the development of microcontroller techniques, the motes are getting more and more powerful and cheaper. Using the motes in the traditional way, doing nothing on data from the network is a waste of such computation capability on the devices. In network data processing (computation) can also be served to make WSNs more intelligent and save energy sometime for high data rate applications by filtering out trivial data and parameterize only the important features to the monitoring.

Another problem with the traditional centralized method without in network data processing is that it might require high bandwidth on the base station. For WSN, this could be a problem. In wired network, the bandwidth of a node can be improved by reserving high quality physical links, like optical fibers, or by providing some QoS with preserved resources, or using expensive hardware like 1G/s or 10G/s network interface cards. While in wireless network, especially in sensor network, this is not the case. The bandwidth of the base station cannot easily be improved with similar strategies. More commonly the base station uses the same kind of device as the sensor nodes. Furthermore, the bandwidth on the base station is also limited by the nodes that will communicate with it. Although the base station can send message to other nodes with higher power level and using antennas, the reverse direction is not improved this way. To make it worse, most data transmissions in traditional WSNs are from sensor nodes to the base station.

So a hierarchical system design is a better choice in handling these problems. The network can be divided into several clusters, each cluster with a cluster head or manager to act as the central controller for that cluster, and communicate with higher level nodes like the base station. When having a large network, more sensor nodes can be supported by having more clusters and more intermediate levels. By separating the WSN spatially into clusters and controlling the scheduling within the cluster temporally, the whole network can afford more data transmission and concurrency.

In our work, we proposed a flexibility based damage detection and localization method for civil structure health monitoring. The approach is based on the observation that for most metal structures, damages will change the flexibility of the structure. And the change in flexibility can be captured with frequency domain analysis. To simplify, when damage happened, the natural frequencies of the structure usually shift a little and this shift can be used to calculate the change in flexibility.

We then designed and implemented this method on a system to monitor the health of civil structures, mainly but not limited to bridges. In the system, sampled vibration responses are collected, frequency domain analysis was carried out, and damage localization algorithm was used to decide where the damages come from. All these computations are distributed among the sensor nodes in the network. The system is able to detect, locate and evaluate the degree of the damage in the structure so that necessary measurements can be taken to repair the damages and maintain the healthiness of the structure to avoid accidents caused by structural failure.

In order to extend the lifetime of the wireless sensor network, we took advantage of the computational power of the Imote2 platform and make motes do different computations on them. Then only the results related to the flexibility calculation is transmitted in the network. This strategy is a tradeoff between energy consumed by transmitting large amount of data and energy for computing.

We also make the system to work in multiple levels style. Since most of the time, the structure is normal and we just want to have a scan now and then, sampling and doing analysis in a long interval period. The assumption that damages only showed up infrequently drives us to use fewer motes on this lower level detection. On the other hand, whenever some suspected event does show up, we are able to trigger higher levels by activating more sensor nodes in the system to make a better damage detection and localization. This multi-level design helps to extend the lifetime of the whole network, since energy-saving lower level damage detection is used most of the time and the energy-consuming higher level damage detection is triggered only when necessary.

Chapter 2

Method of Damage Detection

There are two categories of damage detection algorithms, one focus on time domain, and another uses frequency domain. Our method used frequency domain analysis to detecting shifts in natural frequencies of the structure, and then calculated the change in flexibility to detect and locate damages. The major techniques used are: modal identification using FDD, flexibility-based damage detection, and distributed computation strategy.

2.1 Modal Identification Using FDD

When monitoring in service civil engineering structures, one effective method for output-only modal identification is the Frequency Domain Decomposition (FDD) method [11].

In the FDD method, the cross spectral density (CSD) matrix of responses at each discrete frequency is first estimated. To minimize the impact of measurement noise, the averaged CSD matrix is obtained by performing an averaging operation on the CSD matrices estimated from multiple frames of data. Then a singular value decomposition (SVD) is performed on the averaged CSD matrix at each discrete frequency. The maximum singular value in each singular value matrix is collected to form a vector. From the peaks of this vector, structural natural frequencies are identified. The first column of the left singular decomposition matrix corresponding to a particular natural frequency is an estimate of the corresponding mode shape.

2.2 Flexibility-Based Damage Detection

Techniques for damage detection based on structural flexibility have been gaining attention. A good estimate of the flexibility matrix can be obtained with easily identified low-frequency modes, making them attractive for civil engineering applications.

Based on the assumption that the presence of damage in structures reduces structural stiffness, and thus increases structural flexibility, the change in structural flexibility between the pre- and post-damaged states can be used to detect damage, which is the fundamental basis of the classical flexibility difference method [15]. Because the damage detection results using classical flexibilities are embodied as nodal or DOF's (degree of freedom) characterization, the classical flexibility difference method cannot directly localize damage to exact elements. Consequently, the ASH flexibility-based method [16] was proposed for localizing damage in beam-like structures. This method determines the change in Angles-between-String-and-Horizon (ASHs) of beam elements caused by damage, and thus it can localize damage to exact elements.

Thus, the components in the ASH flexibility are associated with beam-elements of the beam's finite element model rather than nodes.

The maximum absolute values of the components in each column or the diagonals in the difference of ASH flexibility matrices between the pre- and post-damaged structures are extracted as damage indicators. By observing a "step and jump" in the plot of damage indicators vs. element numbers, the damage locations are determined.

To perform damage localization at the member-level in truss or frame structures, the Axial Strain (AS) flexibility-based method was used. The basic idea is that if members in a structure are dominated by axial forces, as in truss structures, the axial strain will be a better index than deflection for damage detection.

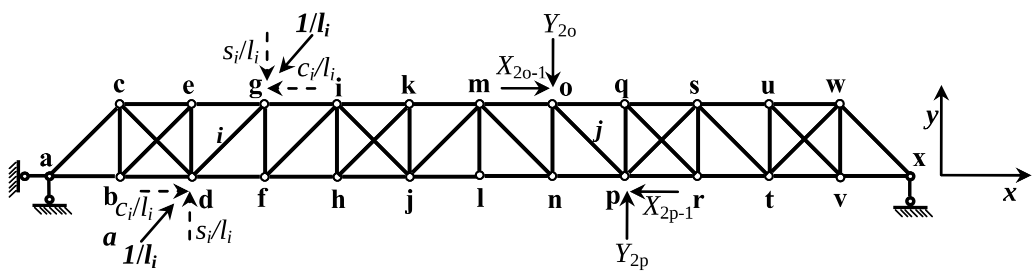


Figure 2.1 Truss structure

The percent change in diagonal elements of the AS flexibility matrices before and after damage is taken as the damage indicators for each element. The elements associated with large values of damage indicators are identified as damaged.

2.3 Distributed Computation Strategy

In this section, the FDD method is modified to reduce computational efforts, and the way in which the modified FDD method and flexibility-based damage detection methods are distributed throughout a WSN is designed to reduce the wireless communication amount to make effective use of energy in each sensor node.

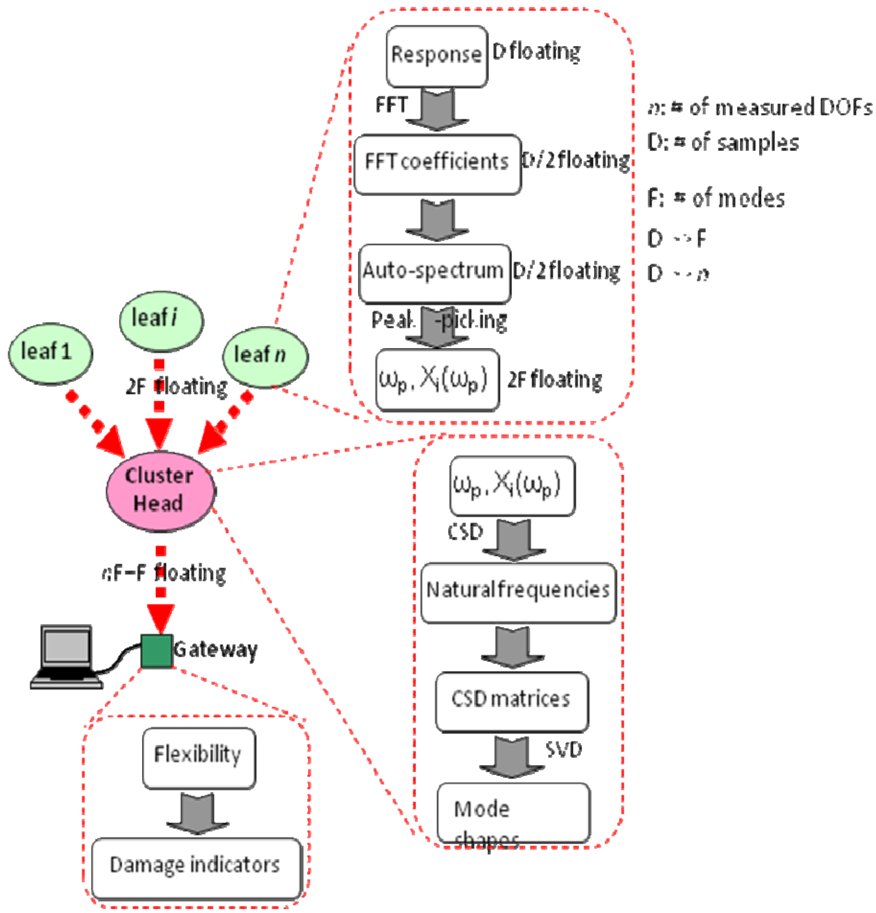


Figure 2.2 Distributions of system identification and damage detection across the WSN

A variation on the traditional FDD method is proposed here. Rather than performing a SVD on each of the CSD matrices at all discrete frequencies, a method with minimal computational efforts, peak-picking, is used first to identify the natural frequencies. Then, noting that only the left singular decomposition matrices associated with the identified natural frequencies are used for obtaining mode shapes, we perform a SVD on each of the CSD matrices associated with those natural frequencies. And accordingly, we will just construct the CSD matrices associated with natural frequencies.

In this way, the computational cost of identifying modal parameters is reduced considerably.

Once natural frequencies and mode shapes are obtained at the cluster head, they are transmitted to the gateway mote. First, the identified natural frequencies and mode shapes are applied to construct a flexibility matrix. Then, damage indicators are extracted from the difference between the flexibility matrix in the current state and the flexibility matrix constructed from the baseline data stored on the gateway mote.

The distribution of the modified FDD method and damage detection methods across the WSN and the data flow between stages are shown in the flowchart in Figure 2.2. Herein, it is assumed that the number of modes to be identified is F, and each data frame has D sampling points, and the number of points in the FFT is D. The amount of data transmitted from each leaf node to the cluster head is $2F$ floating, and the amount of data transmitted from the cluster head to the gateway mote is $(n+1)F$ floating. Both are much smaller than D.

In summary, in this modified FDD method, the SVD is performed on only a few matrices (the number is equal to the number of the identified natural frequencies), therefore the computing efforts at the cluster head are reduced significantly as compared with the original FDD method without sacrificing accuracy in the identified mode shapes. In addition, using the distributed computation strategy, only small amount of data is transmitted wirelessly, which subsequently alleviates the problem of limited power supply of WSNs.

Chapter 3

Design and Implementation

The system was implemented on the Imote2 (IPR2400) sensor devices. The ISHM services toolsuite developed by Illinois Structural Health Monitoring Project (ISHMP) at the University of Illinois at Urbana-Champaign provides many useful components for structural health monitoring projects based on the Imote2 platform. In our implementation, we utilized this toolsuite in our implementation of system, which accelerated the process of the developing cycle of the system and promised a better reliability based on the high quality of the ISHM toolsuite.

3.1 Hardware Devices

Imote2 is an advanced wireless sensor node platform. It is built around the low-power PXA271 XScale processor and integrates an 802.15.4 radio (CC2420) with a built-in 2.4GHz antenna. The Imote2 is a modular stackable platform and can be expanded with extension boards to customize the system to a specific application. Through the extension board connectors sensor boards can provide specific analog or digital interfaces. A battery-board is provided to supply system power, or it can be powered via the integrated USB interface.

a) Imote2

b) ITS400 sensor board

Figure 3.1 Hardware Devices Used

3.2 Software Package

Illinois Structural Health Monitoring Project (ISHMP) is developing hardware and software systems for the continuous and reliable monitoring of civil infrastructure using a dense network of smart sensors. The project has released an open source toolsuite containing a library of services for, and examples of, SHM applications. This toolsuite has been validated on laboratory-scale bridge structures; full-scale validation is currently underway.

ISHMP is collaboration between the Smart Structures Technology Laboratory, directed by Prof. Bill F. Spencer, Jr., from the Civil and Environmental Engineering Department and the Open Systems Laboratory, directed by Prof. Gul Agha, from the Computer Science Department at the University of Illinois at Urbana-Champaign.

Several major components we used in this software toolsuite were the ReliableComm, for reliable communication between sensor motes, the Synchronization components for time synchronize between sensors, the DistributedDataAcquireApp for the base frame work of driving sensors and collecting data in the network.

Based on the toolsuite, we extended the DistributedDataAcquireApp to calculate the FFT and peak frequencies picking on all the leaf motes, and on the managers CSD, PSD and mode shape were computed and then sent to the base station, which will then calculate the flexibility and damage indicators to identify and locate the damages.

Except for the basic function of the network, we implemented a node management scheme to carry out the multilevel damage detection process. Nodes of different roles in the network end their round at different time based on the hierarchical design. The leaf nodes first finish their computation of FDD, and after data was delivered to managers, they are put into deep sleep mode to save energy. Similarly, after the managers finish the computation of mode shape of the leaf nodes and send the data back to the base station, the managers are put into deep sleep mode. Then the base station calculates and localizes the damage and based on the decision to start another round of damage detection. If in the first level, damage was detected, the second level of damage detection will immediately be triggered. Otherwise, the system will wait until the next period come to start another level 1 routine damage detection.

The civil structure may extend in a large geology span. To save energy, we divide the sensors into clusters, each cluster can work as a unit and has a cluster head to coordinate nodes within the group and communicate with nodes in other groups and the base station.

Many algorithms for structure damage detection require the sampled data to be synchronized. The synchronization of the sensor motes was achieved with the SynC component in the ISHM toolsuite. Our system also takes advantage of the mote synchronization into node management and scheduling.

3.3 In Network Data Processing

Although collecting and storing raw sampling data may be the major tasks for some wireless sensor network applications, such as [1, 2, 3, 4]), for them firsthand data from those environments are very precious for scientific research, there are also other applications that are more targeted and well studied and modeled, they care more about the related things in the environment to help make decisions, interested in capturing some event or change in the environment monitored by a WSN. Take the structure monitoring for example, most of the time we don't care about the vibration data when the structure is in good condition, since those data are most often stable or predictable for a long time. It doesn't make too much sense to keep record of a large amount of data with repeating pattern and model. On the contrary, people care more about those events that might reflect the changed in health condition of the structure, like when some damage showed up on somewhere, so as to analyze the reason and dynamic of the environment and take actions. Therefore, it is beneficial to let the sensor nodes preprocess the raw data to 1) decrease the data size so as to save energy, and also as a filter for the event, and drop those data that we are not interested in. Considering this we designed our system to processing data in the network, instead of sending the raw data back, we capture the major features of the condition and send them back for analyzing.

Another motivation for in network data processing is that it saves energy for the WSN and extends the lifetime of the whole system, which is always a concern for WSNs since they always have to rely on just several batteries to work for more than one year. Sometimes, it is very expensive for the deployment and maintenance of the WSN, some of them are deployed to very harsh natural environments [5, 6]. It takes a large amount of energy to transmit these raw data to the base station.

Moreover, for systems with high data rate and large amount of data, transmitting the raw data might just not be possible due to the bandwidth limitations that wireless sensors can achieve as we discussed about centralized WSNs. Although it is promising to use energy procure techniques, like solar power and environmental vibration power, these kinds of technique are still not matured at this time and it also increase the total cost of the system, and moreover not suitable for all the environments, for example solar power has requirements for sunshine.

More detailed of the distribution of the in network data processing is described in Figure 2.2, where the computations on the different network nodes was listed according to their roles in the system.

3.3.1 Roles in the System

There are three different roles in our wireless sensor network: base station, cluster head, and cluster member. The roles are divided by what data they are handling and in what level of the hierarchy the nodes are working. Cluster members are the end sensors that collecting the raw samples of responses like vibration. These data was then not directly transmitted to nodes of higher hierarchy, cluster heads. Instead a FFT is first carry out to transform the response into frequency domain, and a peak picking algorithm is run to extract the FFT value near those peak frequencies, which usually shift a little comparing to the natural frequency of the structure. Then the peak frequencies and the corresponding FFT values of the cluster members are aggregated to the cluster head, where a SVD is carried out to extract the mode shape vectors of the current state. Finally, all the mode shapes from the different clusters are aggregated by the base station to calculate the flexibility of the structure and to decide with our ASH approach whether some damage appeared and where the damage are.

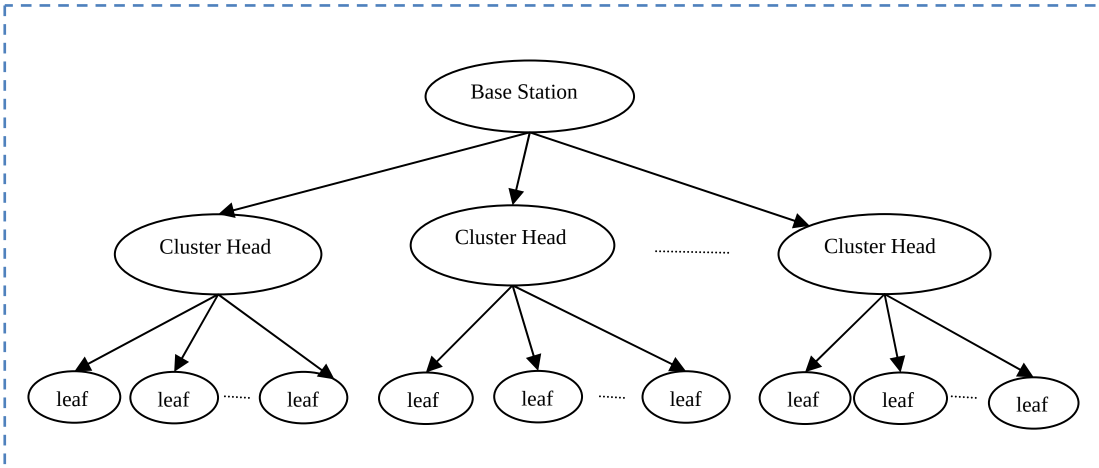


Figure 3.2 Sensor Roles in the System

3.3.2 Dynamic Configuration of the Network

Since the nodes in the network have different roles and we do not want to fix the roles of the nodes to one scheme. Instead we want to be able to schedule the roles according to their power levels, so as to balance the whole energy consumption on all the nodes. Notice that different roles consume different amount of energy in one round. Cluster head consumes the most since it has to carry out the most expensive SVD to get the mode shape, and work as an intermediate node for leaf nodes and the base station.

The idea of dynamic configuration of the network is that every node has all the codes for the different roles, in each round the base station will send a message to him to configure this topology of the network and the hierarchies.

We can even have two or more base stations although at one time, maybe only one base station is selected, so that we can get more concurrency from the network. For example, according to the finite state machine, the cluster members finish all its task the earliest in one damage detection cycle, and the cluster heads finish a little later, after it collected all the FFT data, it have to do mode shape and sending those data to base station and then done, and the base station is the last one to finish after it get the mode shape from the cluster heads, it has to compute flexibility and carry out a damage localization algorithm to identify the damages. During this time period, we actually can activate another base station to carry out another round of run, of act only as a backup when one base station breaks.

Figure 3.3 show an example of the initializing and configuration of the network. At first the nodes in the network don't know what roles they will be played in the system. Then a configuration parameter was disseminated from the gateway node to all the other nodes in the network, the parameters include information on the cluster division and the leaf nodes in the cluster. Each node gets the configuration information and set up their function accordingly. Thus the whole network is initialized.

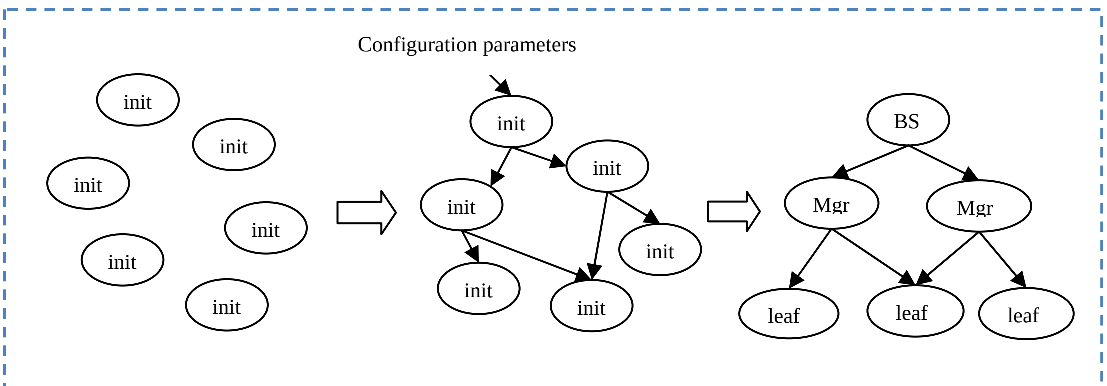


Figure 3.3 Network Configuration Process

3.3.3 Finite State Machine

The coordination of the different sensor motes with different roles is maintained by finite state machine. The base station, cluster heads and cluster members changing message to move the network forward. The three major phases of the network are network initialization, synchronization and sampling, FFT, mode shape, and calculate flexibility and damage localization. It can be looked at that these functions are carried out sequentially, but maybe on different nodes in the wireless sensor network. And the nodes acknowledge each other and exchange state information through a finite state machine mechanism. During each state, the node will have some model in handling incoming message and processing, after it finished with the state and received the reference signal from its controllers it will move on to the next state and acting in a different pattern.

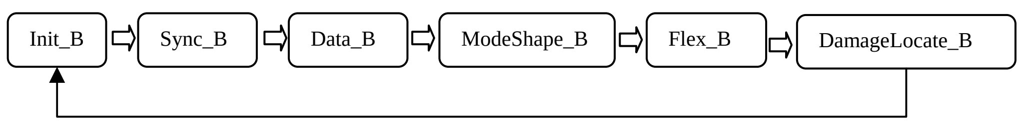


Figure 3.4 Base Station Major States

The base station has the several major states as described in Figure 3.4. First it will get the proper configuration for the network and use that configuration to initialize other nodes in the network. After the configuration of the network is finished, it starts a synchronization process to synchronize the time on the sensor motes. Then data sampling and retrieval is triggered. The leaf nodes sampled the vibration responses and do a FDD to get the FFT value at the peak frequencies and send the results to the managers of its cluster. The managers then can calculate the mode shape within its cluster and send the results to the base station. Receiving the mode shape data from the managers, the base station transfer to the Flex_B state to calculate the flexibility values, and then in DamageLocate_B state a damage localization algorithm was executed to determine whether there is damage or not. If there is damage, it will also evaluate the coarse grained possible damaged area and trigger level-2 damage localization by activating more sensor nodes. If no damage was detected, the system will wait until the next period to start another round of level-1 damage detection.

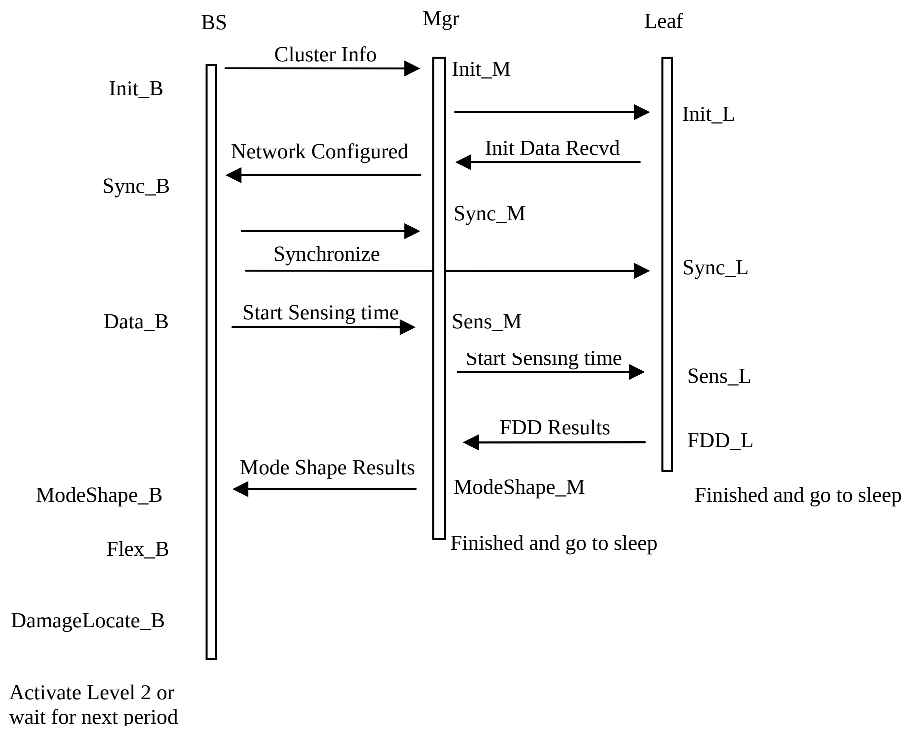


Figure 3.5 Distributed Finite State Machine

The coordination between the sensor nodes is very important for the system. This coordination was maintained by the exchange of messages and the propagation of the state information. In principle, the base station is the control center, it send commands to the managers and get results back. When received a command from the base station, the manger nodes will send corresponding task commands to its leaf nodes and retrieve the results. So it is a network based on hierarchical control flow. Figure 3.5 shows the major states of the different nodes and the message exchanged between them.

3.4 Multi-level Damage Localization

For civil structures, most of the time their statuses won't change dramatically unless some accidental event like earthquake happened. So we can use fewer sensors during normal period and only activate more sensor nodes when some event happened that indicates that there might be some damage in the structure. Since we are using sparse sensors for low levels, it can only get a coarse evaluation of the suspected areas. Then the system can activate more sensor nodes in that area to get much higher resolution detection. This concept was implemented in our system by a multi-level damage detection and localization design. Take two levels as an example, during level-1, we only use 10 percent of all the sensors in the network as sentinels. When in level-1, some abnormal event was detected, we will go on to activate more sensors to take level-2 damage localization, in this level most of the sensors around the suspected area is activated to get a finer grained damage localization report. These model can be extends to more levels with more percentage of sensors used for each higher level. So when we find some false positive detection report from the lower level, we can stop as earlier as possible before it activates too many sensors for that. On the contrary, when some damage really happened, we'll accelerate the process by skipping some intermediate levels in a binary search way.

Corresponding to multi-level, we can also design the system to be heterogeneous by having different sensor platforms in different levels. For example, we can have Telosb, Mica motes which consume less energy than Imote2 for data collecting to work in lower level detections and only activate more energy consuming devices such as Imote2 sensors, to handle complex computations when necessary, such as some event was detected.

3.5 Energy Saving Techniques

As we have mentioned, in network data processing helped to decrease the wireless communication cost, and the multi-level damage localization strategy further extends the system lifetime. There are other techniques that are also designed to better tune the energy consumption. In level-1, we also try to balance the energy consumption of all the nodes in the network in a round robin approach. Since only some percent of the sensor nodes should be activated, we chose different nodes in the cluster as the manager of that cluster and choose to activate different sentinels in different rounds. This strategy in selecting sensors also helps to cover damage showed up in different places. It is like that we deployed several patrol guards; each round different sections of the structure is examined.

Another technique for saving more energy is to coordinate the transmission of data within clusters. By let the higher level nodes control the lower level nodes, and schedule their data transmission so that no collision was formed in a TDMA way, we save more energy for node to handle transmission failure and do retransmission. Geographically dividing the whole structure area into clusters also help to lower the communication cost within the cluster. Since now they are geographically close to each other, we can lower the power level in sending message within the cluster.

3.6 Mapping of Structure Location

For the localization to workout, the knowledge of the structure and the deployment is very important for finding the relationship between data collected and the location it corresponds to. In our system, the structure model is maintained by data structures that store the deployment information and information related to the structure such as elements, length, the available motes, the active motes etc. so that the base station can use these information to localize the physical place the damage indicators show.

For beam, the data structure is simply an array which stores the node IDs in order and the distance information. When ASH was used, each damage indicator was corresponding to the beam range between two nodes on the beam. So maximal or minimal damage indicator reflects the corresponding area might have some damage if it exceeds some threshold based on the knowledge and model simulation to the beam.

For truss structure, the data structure is a little more complex. Since now we have a 3-Dimension structure, and the AS method use elements as damage localization units. In order to separate the sensor node deployment to the damage detection algorithm, we established a mapping from sensor nodes to its physical position on the structure, so that damage detection algorithm only care about the data related to the physical position and it doesn't matter which specifically sensor node was deployed there. This also simplified the deployment of the system, since we only have to input the mapping relation into the right place. We also build the structural model so as to retrieve the element information and which two nodes the element connects. This information was used by the AS damage localization algorithm. Furthermore, for our multi-level damage localization method, we implemented two data structures, the active nodes, the available nodes. So in different levels rounds, the AS algorithm will find the exact elements that are active, both its end nodes are activated, for damage detection.

Chapter 4

Experimental Validation

To experimentally validate the proposed strategy, we implement and deploy it on the Intel Imote2 wireless sensor network platform and associated sensor boards. Procedures executed in the proposed system at the leaf nodes, cluster heads and the gateway mote of the base station are listed in the respective blocks in Figure 4.1.

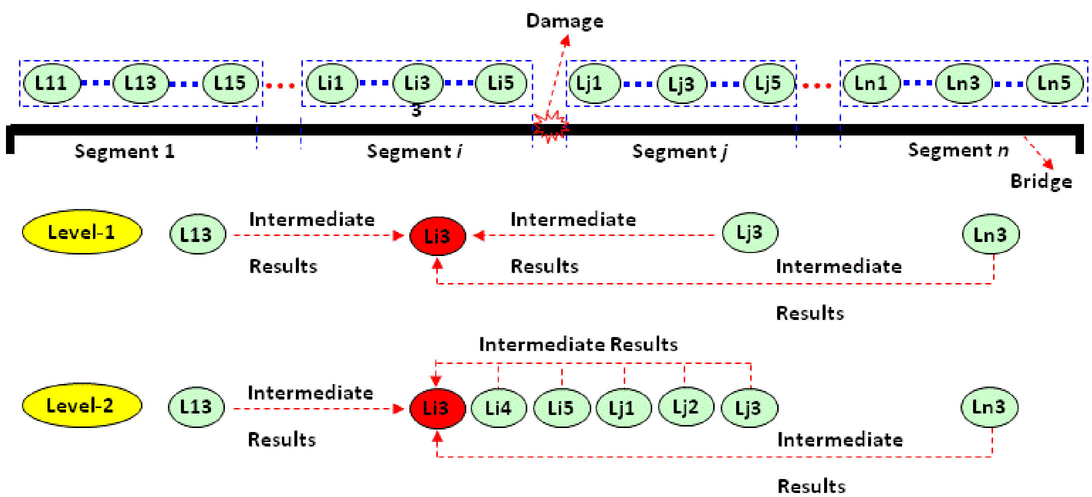
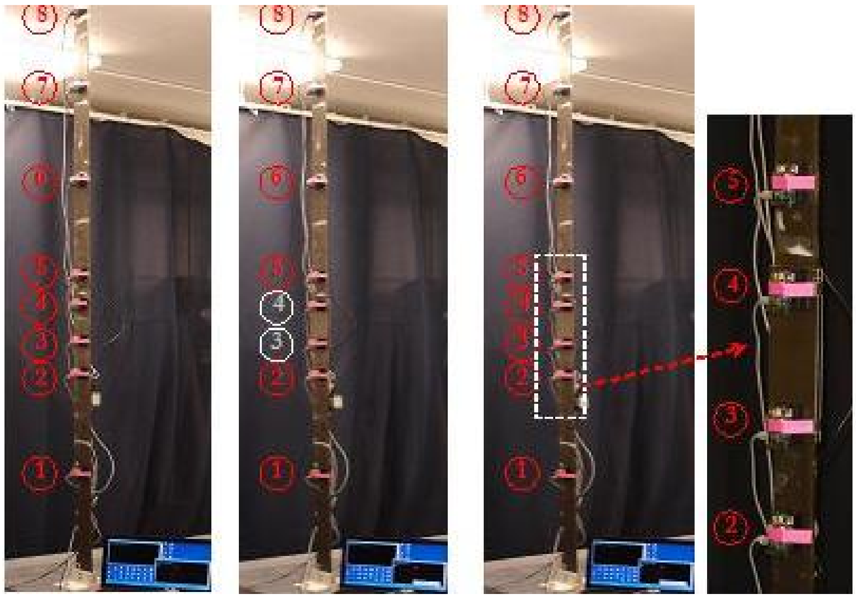


Figure 4.1 Network architecture for multi-level damage localization strategy with distributive computation

Experimental validation tests are conducted using a steel cantilever beam at the Structural Control and Earthquake Engineering Lab at Washington University. The beam is 108 inch long, 3 inch wide and 0.25 inch thick, as shown in Figure 4.2. The sensor numbers are shown in the circles in each figure. The beam is fixed to the shake table. Damage in the beam is simulated by adding a pair of thin, symmetric steel plates in element 4. These plates are 9 inch long, 3.625 inch wide and 0.0625 inch thick.



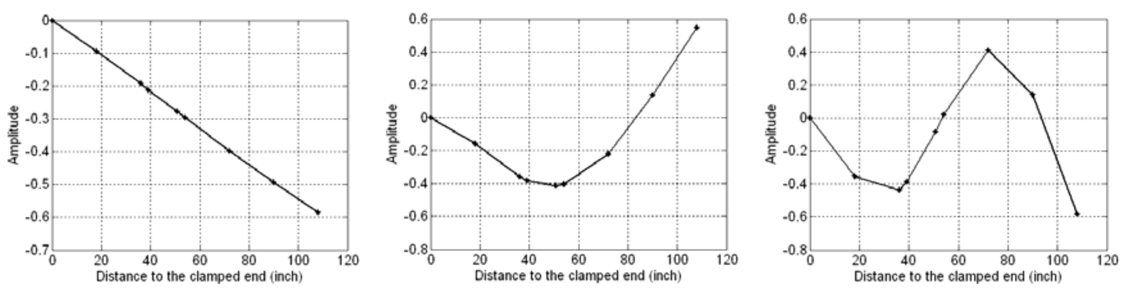
a) baseline test b) level-1 detection c) level 2 localization

Figure 4.2 The cantilever beam and sensor placements in the experiment tests

In these tests, the SHM system includes a PC base station, eight Intel Imote2 motes (IPR2400) with sensor boards (ITS400C), and a "gateway" Imote2 tethered to the base station with a PC interface board (IIB2400). Sensors are deployed along the beam, as shown in Figure 4.2. In this experiment, all sensors are within a single hop from the base station. All modal identification and damage detection procedures are automated on the sensors. The damage indicators are extracted at the gateway mote connected to the base station.

The beam is excited along the weak axis of bending using an impact. The acceleration response in this direction is collected at each node. For data collection, the sampling frequency of acceleration response is 280 Hz, the length of record is 7168 points, and the number of points in the FFT is 2048.

First, we run the developed WSN-based SHM system on the intact beam to obtain baseline modal parameters. These values are saved on the gateway mote connected to the base station. For purposes of code validation, we write a file containing the obtained baseline data, the identified natural frequencies (Table 4.1) and mode shapes (Figure 4.3).



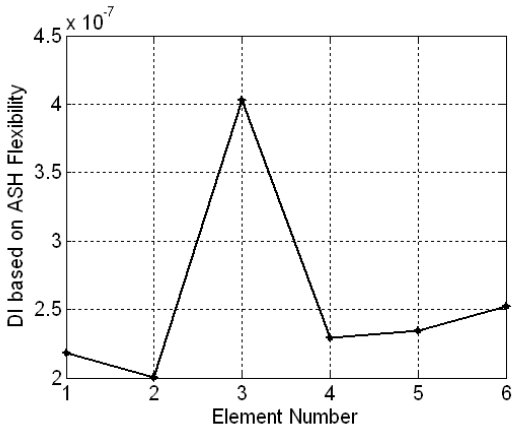
(a) 1st mode of intact beam (b) 2nd mode of intact beam (a) 3rd mode of intact beam

Figure 4.3 Identified mode shapes of the intact beam

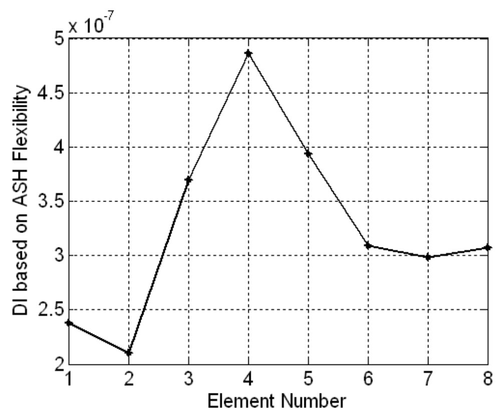
Table 4.1 Identified natural frequencies of the cantilever beam before and after damage

Order	Intact (Hz)	Damaged (Hz)	Percentage Change (%)
1	0.5469	0.5469	0
2	3.9648	3.9648	0
3	11.1454	11.2109	0.59

Then, we deploy the WSN-based SHM system on the damaged beam. The gateway mote extracts the damage indicators automatically, and identifies if the beam is damaged and when to initiate the level-2 damage localization. For level-1 damage localization, only six sensor nodes (nodes 1, 2, 5 through 8) are activated. The extracted damage indicators at the gateway mote are plotted in Figure 4.4(a). The damage indicator associated with element 3 exhibits a peak, which means the damage is localized to element 3 (corresponding to the current network architecture). Then, the system automatically activates two more sensors within element 3 and performs level-2 detection. The damage indicators extracted by the system are plotted in Figure 4.4(b). From the peak among the damage indicators, we can localize damage to a smaller region (element 4 in the new network architecture) which is consistent with the position of the two steel plates.



a) damage indicators
when six sensors are activated



b) damage indicators
when eight sensors are activated

Figure 4.4 Damage localization results using the developed WSN-based SHM system

Chapter 5

Discussion and Future Work

After the validation of our method and implementation on two different categories of bridges: steel cantilever beam and truss, we have more confidence in the effectiveness of our method. The experiments have shown the ability of the approach in identifying multi-damages, although at this time, we only implemented the reporting of the major damage, since it is the most critical.

One problem with multiple damages identification is how to evaluate the influence of concurrent damages. It is possible when multi-point damages coexist, they interference with each other and make it hard to detect the exact location of the damages. In this case, we have to improve our approach to take the influence of multiple damages, and try to eliminate the damages we have already detected before so as to detect new damages.

Another possible direct improvement to our system is to evaluate the degree of the damage. At this time, we use a threshold based on experimental data to differentiate between damage or non-damage using the damage indicators. While in reality, this threshold should be based on the specific structure and some theoretical analysis.

Giving that analysis, we can further use the damage indicator to not only locate the damage but also to evaluate the degree of the damage.

Better energy conservative schemes can be designed to save even more energy and configure the network based on the power level of the nodes in the network, and how the computation should be distributed among the network. Moreover, for large structures, multihop routing protocols can be designed to better decide the network configuration; we should dynamically activate, and take advantage of the hierarchical system design when picking routings.

Chapter 6

Conclusion

We proposed a frequency domain damage detection and localization method, which use the shifts in natural frequencies and the change in flexibility to identify damages. We designed and implemented the system in a hierarchical way so that damages were detected in a multi-level approach, which helped to extend the lifetime of the whole WSN. Many techniques helped to make the system to better balance the power consumption on all the nodes. The distribution of computation and the dynamic configuration of the network made nodes to take turn to act as different levels of roles in the system instead of rely on fixed nodes. The experiments on a cantilever beam and truss structure validated our damage detection method and the system design for saving energy.

Appendix A

Computation of FDD and Flexibility

FDD is combined with peak-picking to identify the modal parameters. First, on the microprocessor of each leaf node, a fast Fourier transform (FFT) is performed on the data collected by this sensor node as

$$X_i(\omega) = \mathrm{F}\left[x_i(t)\right] \tag{5}$$

where $\mathrm{F}[\cdot]$ represents the FFT operation. $X_i(\omega)$ is the FFT coefficient of the response $x_i(t)$ at the *i*th node. Second, the auto-spectrum of each response is calculated as

$$P_i(\omega_j) = X_i(\omega_j) X_i^*(\omega_j) \tag{6}$$

where $P_i(\omega_j)$ denotes the power spectral density (PSD) function at the *j*th discrete frequency of $x_i(t)$. The peaks of the PSD of $x_i(t)$ are identified for determining the natural frequencies using that the assumption that the external excitations considered here are broadband ambient vibrations. Here we use ω_p to represent the discrete frequencies associated with the identified *p*th peak. This step is also performed independently at each leaf node. However, not all peaks are necessarily related to natural frequencies of the system. A discussion of some practical issues associated with this step is provided in the sequel.

From each leaf node, only the discrete frequencies ω_p and the FFT coefficients $X_i(\omega_p)$ corresponding to the peaks are transmitted to a cluster head. Obviously, this significantly reduces the amount of data to be transmitted compared with transmitting the entire time history.

The remaining steps involved in modal identification are performed at the cluster head. After the cluster head receives a set of intermediate results obtained from one frame of data from each leaf node, the CSD between each response and a reference response (the response at the cluster head is taken as the reference response here) is calculated to determine if each discrete frequency ω_p is a structural frequency. To judge this, the phase of the CSD is examined. For a discrete frequency ω_p, if the phase of the corresponding CSD at $\omega = \omega_p$ is close to 0 or π, the discrete frequency ω_p is a natural frequency of the structure (designated ω_n). Using this criterion, the natural frequencies can be identified with the intermediate results.

Then, for this frame of data, the CSD matrix corresponding to each natural frequency is estimated from the FFT coefficients associated with the identified natural frequencies

ω_n (instead of ω_p). The estimated CSD matrix corresponding to the *k*th natural frequency ω_n^k is expressed as

$$\mathbf{G}(\omega_n^k) = \begin{bmatrix} X_1(\omega_n^k)X_1^*(\omega_n^k) & \cdots & X_1(\omega_n^k)X_i^*(\omega_n^k) & \cdots & X_1(\omega_n^k)X_n^*(\omega_n^k) \\ \vdots & & \vdots & & \vdots \\ X_i(\omega_n^k)X_1^*(\omega_n^k) & \cdots & X_i(\omega_n^k)X_i^*(\omega_n^k) & \cdots & X_i(\omega_n^k)X_n^*(\omega_n^k) \\ \vdots & & \vdots & & \vdots \\ X_n(\omega_n^k)X_1^*(\omega_n^k) & \cdots & X_n(\omega_n^k)X_i^*(\omega_n^k) & \cdots & X_n(\omega_n^k)X_n^*(\omega_n^k) \end{bmatrix} \tag{7}$$

After the intermediate results obtained from various frames of data are transmitted to the cluster head, the average value of identified natural frequencies from all leaf nodes and from all various frames of data is calculated to obtain the final identified natural frequency for each mode. The averaged CSD matrix associated with each natural frequency (designated $\bar{\mathbf{G}}(\omega_n^k)$) is obtained by performing an average on $\mathbf{G}(\omega_n^k)$ estimated from various frames of data. Next, a SVD is performed on each of the averaged CSD matrices corresponding to each natural frequency to identify the associated mode shapes

$$\mathbf{U\Sigma}V^{\mathbf{T}} = SVD\left(\bar{\mathbf{G}}(\omega_m^k)\right) \tag{8}$$

where $\mathbf{\Sigma}$, $\mathbf{U}$ and V denote the singular value matrix, the left singular decomposition matrix and the right singular decomposition matrix.

The first column of $\mathbf{U}$ is an estimate of the *k*th mode shape and is designated $\mathbf{U}_1$. By dividing all of the components of $\mathbf{U}_1$ by the component of $\mathbf{U}_1$ chosen as a reference, the normalized mode shape is obtained with one component having a value of one. Its components are, in general, complex values. The phase associated with each complex value represents the phase difference between that response location and the reference sensor location in the *k*th mode. To obtain the real-valued components of the mode shape, which are typically used for damage detection, the magnitude of each component of the normalized $\mathbf{U}_1$ is calculated. The corresponding sign for each component is determined by its respective phase. The phases of the components in the normalized mode shape are ideally equal to 0 or π for proportionally damped systems with no measurement error. In practice, due to measurement and numerical errors, the phases are not exactly 0 or π. The signs of the components are determined as follows (as in the original FDD method): if the phase is in the range of $\left[-\frac{\pi}{2} \quad \frac{\pi}{2}\right]$, the corresponding sign is positive; otherwise, if the phase is in the range of $\left[\frac{\pi}{2} \quad \frac{3\pi}{2}\right]$, the corresponding sign is negative.

The ASH flexibility matrix can be constructed as

$$\mathbf{F}_\theta = \sum_{r=1}^{n} \frac{1}{\omega_r^2} \mathbf{R}_r \mathbf{R}_r^T \tag{1}$$

where $\mathbf{R}_r$ is called the *r*th ASH mode shape, which can be expressed in terms of the *r*th translational mode shape as

$$\mathbf{R}_r = \left[\frac{1}{l_1}\varphi_{1,r} \quad \frac{1}{l_2}(\varphi_{2,r} - \varphi_{1,r}) \quad \cdots \quad \frac{1}{l_i}(\varphi_{i,r} - \varphi_{i-1,r}) \quad \cdots \quad \frac{1}{l_n}(\varphi_{n,r} - \varphi_{n-1,r}) \right]^{\mathrm{T}} \tag{2}$$

where $\varphi_{i,r}$ denotes the *i*th component of the *r*th mode shape, and l_i denotes the length of the *i*th beam element. The components in the *r*th column of this flexibility matrix represent the ASHs of all beam elements of the structure resulting from a unit moment applied at two nodes of element *r*, with no force or moment on the other elements.

References

[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. "Wireless sensor networks for habitat monitoring." In *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*, Sept. 2002.

[2] Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., and Hong, W. 2005. "A macroscope in the redwoods." In *Proceedings of the 3rd international Conference on Embedded Networked Sensor Systems* (San Diego, California, USA, November 02 - 04, 2005). SenSys '05. ACM Press, New York, NY, 51-63.

[3] Krishnamurthy, L., Adler, R., Buonadonna, P., Chhabra, J., Flanigan, M., Kushalnagar, N., Nachman, L., and Yarvis, M. 2005. "Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea." In *Proceedings of the 3rd international Conference on Embedded Networked Sensor Systems* (San Diego, California, USA, November 02 - 04, 2005). SenSys '05. ACM Press, New York, NY, 64-75.

[4] W.-Z. Song, R. Huang, and M. Xu., "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, Kraków, Poland, 2009.

[5] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter. "LUSTER: Wireless sensor network for environmental research." In *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys), Nov. 2007*.

[6] Rice, J. A. and Spencer Jr., B. F., "Flexible Smart Sensor Framework for Autonomous Full-scale Structural Health Monitoring," *NSEL Report No. 18*, University of Illinois at Urbana-Champaign. 2009.

[7] Nagayama, T., Spencer, Jr B.F. (2007). "Structural Health Monitoring Using Smart Sensors" *Newmark Structural Engineering Laboratory (NSEL) Report Series, No. 1*, University of Illinois, Urbana, IL.

[8] G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dyke, "A Holistic Approach to Decentralized Structural Damage Localization Using Wireless Sensor Networks," *IEEE Real-Time Systems Symposium (RTSS'08)*, December 2008.

[9] Matteo Ceriotti, Luca Mottola, Gian Pietro Picco, Amy L. Murphy, Stefan Guna, Michele Corr´a, Matteo Pozzi, Daniele Zonta, and Paolo Zanon.

"Monitoring Heritage Buildings with Wireless Sensor Networks: The Torre Aquila Deployment." In *Proceedings of the 8th ACM/IEEE Int. Conference on Information Processing in Wireless Sensor Networks - SPOTS Track (IPSN/SPOTS),* San Francisco (CA, US), April 2009.

[10] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon, "Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks," In *the Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN '07)*, Cambridge, MA, April 2007, ACM Press, pp. 254-263.

[11] Brinker, R., Zhang, L.M., Andersen, P. (2000)„ "Modal identification from ambient responses using frequency domain decomposition", *Proceedings of the 18th International Modal Analysis Conference*, San Antonio, TX, pp. 625-630.

[12] Gao, Y., Spencer, B.F. (2002), "Flexibility-Based Damage Location Employing Ambient Vibration", *Proceedings of the 15th ASCE Engineering Mechanics Conference*, New York.

[13] ITS400 Imote2 Basic Sensor Board, Crossbow Technology, Inc..

[14] Maróti, M., Kusy, B., Simon, G., Lédeczi, A. (2004), "The Flooding Time Synchronization Protocol", *Proceedings of the 2nd International Conference on Embedded Network Systems*, 39-49.

[15] Pandey, A.K., Biswas, M. (1995), "Experimental Verification of Flexibility Difference Method for Locating Damage in Structures", *Journal of Sound and Vibration*, 184, 311-328.

[16] Yan, Guirong, Duan, Zhongdong and Ou, Jinping (2009), "Damage Detection for Truss or Frame Structures Using an Axial Strain Flexibility", *Smart Structures and Systems, an Int. Journal,* 5(3).

Vita

Weijun Guo

Date of Birth	February 04, 1980
Place of Birth	Pingxiang, Jiangxi, PRC
Degrees	B.S. Computer Science, May 2002 M.E. Computer Engineering, May 2005 M.S. Computer Science, December 2009
Publications	Guo Weijun, Xu Yinlong, Wu Junmin. "A Distributed Framework for Integrated Network Management Applications Based on CORBA", Journal of Computer Applications(Chinese), 2005 Vol.25 No.z1 Xu Yinlong, LI Lin, CHEN Guoliang, WAN Yingyu, Guo Weijun, "Multicasting and Broadcasting in Undirected WDM Networks and QoS Extensions of Multicasting", International Journal of Foundation of Computer Science, 2004, 15(1):187-203.

Dec 2009