A Reconfigurable Wireless Sensor Network for Structural Health Monitoring

M. Bocca[1], E. I. Cosar[1], J. Salminen[1], and L.M. Eriksson[1]

[1] Helsinki University of Technology (TKK), Department of Automation and Systems Technology, Helsinki, Finland

ABSTRACT: The paper presents a reconfigurable wireless sensor network for structural health monitoring. Real-time and periodic structural health monitoring can reduce the probability of collapse and the consequences of potential life-threatening conditions. The monitoring system is composed of two parts: the wireless network in which the sensor nodes are equipped with a 3-axis accelerometer and a temperature and humidity sensor, and a Matlab program that processes and visualizes the data in real-time. The first records acceleration signals in different locations of the monitored infrastructure and conveys them at the sink node. The latter plots the collected signals and computes their frequency spectrum in real-time. The wireless sensor node and the implemented network architecture were tested on a bridge structure built to scale. The results show that the sensor nodes provide data that, if processed in the frequency domain, allow the correct identification of the vibrations frequencies of the structure.

1 INTRODUCTION

Civil infrastructures such as bridges, dams, buildings, pipelines, road tunnels, etc. are subjected to harsh environmental conditions (strong winds, heavy rains, high humidity, huge temperature variations) and to catastrophic events (earthquakes, hurricanes, floods) that can severely affect their health and induce potential life-threatening conditions. The above mentioned factors and the deterioration caused by corrosion, age, and fatigue, can result in partial or total collapses. The I-35W Mississippi River bridge located in Minneapolis, Minnesota, USA, is one recent example: it failed on August 1, 2007, collapsing to the river and riverbanks beneath, killing 13 people and injuring 145.

Structural health monitoring (SHM) can prevent these tragic incidents. For civil engineers, wireless sensor networks (WSNs) are an attractive technology: compared to traditional wired systems, they consistently reduce the installation time and costs, as described by Celebi (2002), and are not subjected to wires wear and tear or breakage caused by harsh weather conditions or other extreme events. However, the wireless sensor nodes forming the network do not guarantee the same reliability and speed of the data link: radio packets can go lost during the transmissions, and the employed transceivers usually have limited power and bandwidth. The resources of the nodes are also limited in terms of computational power of the microcontroller unit (MCU), memory space, and power source. All these facts make the implementation of a WSN a challenging task.

Because of the low installation costs, WSNs allow the deployment of a big number (potentially hundreds) of wireless sensor nodes on the structures, increasing the screening capability of the civil engineers. Moreover, the nodes can be placed on particularly critical components of the structure difficult to be reached by wires. However, the data collected by the sensors installed on the structure need to be processed by some damage detection method able to detect signs of structural distress and fatigue. Over the last decade, many methods have been developed, as described in Sohn & Farrar (2001) and Doebling et al. (2004). Other methods have been specifically designed for the resource constrained wireless sensor nodes, which can locally process the collected data and reduce the number of radio transmissions at the sink node of the network, as in Lynch et al. (2004) and Mizuno et al. (2008).

In recent years, wireless sensor nodes and WSNs for SHM have been an object of intensive research. Prototypes have been proposed in Paek et al. (2005), Lynch and Loh (2006), Kim et al. (2007), Chen and Tomizuka (2008), and Ceriotti et al. (2009). This paper presents the created wireless sensor node for SHM and describes the architecture of the implemented WSN, which introduces the possibility to reconfigure the parameters of the monitoring application depending on the needs of the end-user operating at the sink node. To optimize the radio transmission of the data and to avoid interferences, the sink node establishes a reserved communication link with each of the sensor nodes. The paper also presents a Matlab program that visualizes and processes the data arriving at the sink node in real-time. The systems tests are performed on a bridge structure built to scale. The measurements collected by the wireless sensor nodes are compared to those collected by high quality wired accelerometers. The results show that the nodes provide accurate data that, processed in the frequency domain, allow the correct identification of the vibration frequencies of the structure.

2 SYSTEM DESIGN

2.1 Wireless sensor node

The created wireless sensor node is based on Sensinode U100 Micro.2420 sensor networking platform. Its core is a TI MSP430 MCU having 10 KB RAM and 256 KB FLASH memory. The platform is equipped with an external 500 KB serial data FLASH memory. The MCU provides one 12 bits ADC (up to 8 channels available) and two 12 bits DACs. The radio module is a ZigBee, IEEE 802.15.4 compatible, Chipcon CC2420 transceiver, operating in the 2.4 GHz band, having 250 Kbps bandwidth. The platform runs the FreeRTOS real-time kernel. The node dimensions are 40 x 50 mm. The clock of the MCU and the bus runs at 8 MHz.

The node is equipped with a dedicated sensor board, in which a 3-axis digital accelerometer (LIS3LV02DQ by STMicroelectronics, 7 x 7 x 1.5 mm) is soldered. The accelerometer transfers the data to the MCU through a SPI serial interface. The sensor has a selectable full scale of ±2g or ±6g. The accelerometer board and the Micro.2420 platform are connected in a stack through a 50 pins connector. The default measurement resolution is 16 bits.

The Micro.2420 platform provides another 12 pins external connector to which a low power temperature and humidity sensor (SHT71 by Sensirion) is connected. The default measurement resolution is 14 bits for temperature, and 12 bits for humidity. Changes in environmental conditions affect structural vibration properties such as natural frequencies, mode shapes and damping, as reported in Xia et al. (2006). Hence the temperature and humidity measurements may be used to compensate the effects of environmental conditions on the monitored variables.

The three components (Micro.2420, accelerometer board, temperature and humidity sensor) and two AAA batteries are contained in a 88 x 64 x 35 mm plastic box (Figure 1). To increase the accuracy of the accelerometer measurements, the sensor board is screwed very tightly to the bottom of the box. In order to minimize the damping, the sensor node is attached to the targeted

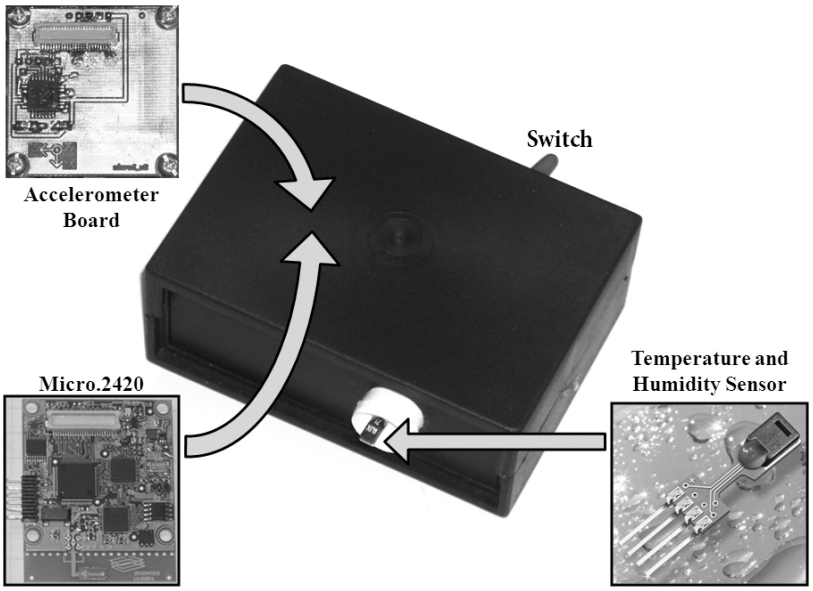

Figure 1. The wireless sensor node.

structure through two very strong magnets glued on the surface of the plastic box. The node can be easily turned on and off through an external switch. Its total weight is 133 grams.

2.2 Network architecture and implementation

In the simulation test bed, the implemented WSN is organized according to a star topology. The nodes, distinguished by an unique identification number, are deployed on the bridge and communicate with the sink node. Although the star topology represents a rather simple scenario, the implemented procedures of sampling, data transmission, and recovery of the lost packets can be ported to more complex multi-hop topologies. WSNs literature reports several routing protocols, as described in Akkaya and Younis (2005), that could be used in a more complicated scenario. Another important issue concerns the synchronization of the clocks of the nodes in the network. An extensive survey is in Sundararaman et al. (2005). The addition of these protocols to the implemented WSN will be part of the future research work.

As the first step, the end-user operating at the sink node sets the parameters of the application. He defines how many sensor nodes, and which ones, will collect data. To specify which nodes, he uses their identification number. Then, he sets the order of reception of the data at the sink node: since the processing and visualization of the data is done in real-time by Matlab, the user may be interested in receiving at first the measurements collected in critical parts of the structure, and in a second time the data collected in other locations.

The end-user proceeds defining the parameters of the sampling that will be performed by the sensor nodes. First, the accelerometer sensitivity, ±2g or ±6g: these two values make the device suitable for ambient vibrations, which are in the order of few mg, or earthquake vibrations, which can go beyond 2g. Second, the sampling frequency: although for very large structures such as bridges, buildings, dams, etc. the natural frequencies of vibration are in the order of few tens of Hz, thus reducing the required sampling frequency to approximately 100 Hz, the sampling frequency of the sensor nodes can be pushed up to 1 kHz. This capability provides to the used data processing method a sufficient amount of data even for vibrations or events lasting

a very limited interval of time (e.g. few ms). Other parameters the end-user can specify are the number of axis, and which ones, the sensor nodes have to consider during the sampling. In this way the data to be transmitted to the sink node can be limited to the directions along which the end-user has already observed or is expecting to observe interesting phenomena. The last parameter is the length of the sampling period: this value is upper-bounded by the limited storage space provided by the external 500 KB serial data FLASH memory contained in the Micro.2420 platform. Specifically, the amount *D* (in bytes) of data collected by one sensor node during the sampling is as follows:

$$D = 2 \cdot T_s \cdot f_s \cdot N_{axis} \quad (1)$$

in which T_s is the time length of the sampling period (in s), f_s is the sampling frequency (in Hz), and N_{axis} is the number of considered axis. The multiplying factor 2 comes from the fact that the accelerometer resolution is 16 bits, i.e. 2 bytes.

After the initial configuration phase, the sink node broadcasts a message containing the values defined for the parameters of the application to the sensor nodes located on the bridge. At the reception of this message, each sensor node checks that its identification number is present in the list of the selected nodes. The ones that are not included in the list do not start sampling and wait for a future request. The selected nodes start the sampling phase by measuring temperature and humidity.

The accelerometer measurements are stored in the external serial data FLASH memory. This memory is divided into 8 sectors, each containing 256 pages. Each page can contain in turn 256 bytes. The memory can be written only in a page-wise manner: during the sampling, a batch of samples is temporarily stored in the integrated RAM memory of the MCU. When the number of acquired samples reaches a pre-defined number, the MCU interrupts the acquisition of the data and stores these samples in a page. The writing operation takes approximately 3.45 ms. Despite the fact that this mechanism would make the sampling asynchronous, the nodes manage to keep the synchronism: at the completion of a writing operation, the MCU waits before restarting the sampling till a multiple of the sampling interval has elapsed from the last sample acquisition.

The sink node waits for the sensor nodes to complete the sampling. Subsequently, it sends a message to the first of the nodes present in the activation list to start the data transmission. The communication link between the sink node and the transmitting sensor node can be established in three different ways: by setting a different UDP port number for each link, by exploiting the capability of the ZigBee transceivers to operate on one of the 16 different radio channels found in the 2.4 GHz band, or by using the 8 bytes MAC addresses of the nodes. By doing this, the sink node creates a reserved communication link with each of the sensor nodes deployed on the structure. The different UDP port numbers and radio channels are calculated as follows:

$$UDP_{Node} = ID_{Node} \cdot UDP_{Network} \quad (2)$$

$$CH_{Node} = CH_{Network} - ID_{Node} \quad (3)$$

in which UDP_{Node} and CH_{Node}, respectively, are the UDP port number and the radio channel the node identified by ID_{Node} uses to transfer data to the sink node, while $UDP_{Network}$ and $CH_{Network}$, respectively, are the UDP port number and the radio channel all the nodes forming the network use to receive and transmit broadcasted messages.

At the reception of the authorization message from the sink node, the targeted sensor node starts transmitting packets containing the data previously stored in the FLASH memory. Because of the limited size of the packets that can be transmitted by the IEEE 802.15.4 compliant ZigBee transceiver, which exploits the IPv6 protocol for networking, the maximum available payload size is 80 bytes. The data stored in one page of the FLASH memory are temporarily saved in the integrated RAM memory: for each page, 3 data packets are transmitted to the sink node. The

first byte of each packet indicates the type of the content (accelerometer data). The next 2 bytes contain a sequential number indicating the position of the transmitted data in the acceleration signal. This number is increased at the transmission of each data packet. In the end, the sensor node sends a message indicating the completion of the data transmission phase. This last packet contains also the information concerning the total number of data packets that were transmitted, and the temperature and humidity measurements.

During the transmission phase, the sink node exploits the sequential number contained in the received packets to control if some data went lost. This can be due to the low power of the radio signal transmitted by the CC2420 transceiver. Another reason can be the fact that UDP provides an unreliable service in which packets can arrive out of order, duplicated, or can go lost without notice. The sink node saves all the lost packets sequential numbers in a list that is used later on in the retransmission phase. Otherwise, when a packet is correctly received, the sink node writes the accelerometer measurements in a .txt file saved on a PC connected to it. This file is read by the Matlab program for processing and visualizing the data in real-time.

The data are transferred to the PC through the serial communication device (UART) of the sink node. This operation is slow and limits the transmission rate of the data packets: if the sensor node transmits a packet while the sink node is still transferring previously received data in the .txt file, the packet is lost. Thus, the serial communication represents the main bottleneck for the bandwidth of the application, limiting it to approximately 6 Kbps. A solution to this problem has been recently presented by Cosar et al. (2009).

In real-world deployments it can happen that, because of sudden breaks or the end of the power source, the node located on the monitored structure is not able to complete the transmission of all the acquired data. The sink node must control if this is the case: at the expiration of a fixed time-out, equal to the time required for the sensor node to transmit 15 packets, in which no data are received, the sink node closes the communication link and sends an authorization message to the next node to start the data transmission. This guarantees the continuity of the operation in case of failure of the nodes.

Otherwise, if some packets went lost, the retransmission phase begins. In it, the sink node sends a retransmission request message to the sensor node indicating the sequential number of the first lost packet. At the reception of this request the node searches the corresponding data in the FLASH memory and transmits them back to the sink node. The sink node transfers again the received values into the previously created .txt file. Similarly, this happens for all the other lost packets, until all the lost data are correctly received. If the number of lost data packets does not decrease over three consecutive attempts of recovery, the sink node interrupts the retransmission phase and proceeds in the flow of operation.

2.3 Data processing and visualization

The Matlab program for data processing and visualization runs on the external PC connected to the sink node. It parses in real-time the content of the .txt file which receives the values through the UART of the sink node. For each node deployed on the structure, and for each selected axis of vibration, the program plots the acceleration signal and its frequency spectrum. In this way the end-user is able to observe in real-time changes in the natural frequencies of the structure. In addition, by analyzing the amplitudes of the signals, it is possible to detect other anomalies.

The Matlab program places the data contained in the retransmitted packets in the correct order inside the time series. The acceleration signals are finally stored in a database which can be used for further processing or for comparisons with previously stored samplings.

3 TEST RESULTS

The proposed platform for SHM was tested on a wooden bridge built to scale, 420 cm long, 65 cm wide, and 33.5 cm high, which has iron joints. Six wireless sensor nodes were attached through their magnets to the joints of the structure as depicted in Figure 2. The bridge is connected to an electric motor that can be programmed to generate sinusoidal vibrations having pre-defined amplitudes and frequencies. Wired high sensitivity digital 1-axis accelerometers (Kistler 8712A5M1) are located in proximity of the joints. These sensors are used as a reference to evaluate the accuracy of the data collected by the wireless sensor nodes.

After checking that the amplitudes of the acceleration signals collected by the nodes and by the wired accelerometers are in the same range, we concentrate our attention on the comparison of their frequency spectrums. As it can be observed in Figure 3, the spectrum of the wired sensors signals does not present components due to measurement noise. This does not happen with the wireless sensor nodes: the spectrum of their signals shows the presence of a small measurement noise component. During the tests it was noticed that the noise can be consistently reduced by screwing very tightly the accelerometer board to the plastic box and by securing all the loose components of the sensor node. As for the accuracy of the data, the peaks in the spectrum of the acceleration signals collected by the wireless nodes (dashed line) slightly move away (few Hz difference) from the reference ones collected by the wired accelerometers (solid line).

Nevertheless, this inaccuracy can be consistently reduced: at the reception of the message specifying the parameters of the sampling, the sensor node computes the number of samples to be collected by multiplying the length of the sampling period by the sampling frequency. In fact, because of the time consumed by the MCU to write the acquired data in the FLASH memory, the actual length of the sampling period, $RealT_s$, is as follows:

$$RealT_s = T_s + T_{page} \cdot N_{pages} \tag{4}$$

in which T_{page} is the time interval, equal to 3.45 ms, in which the MCU suspends the sampling to write a page of the FLASH memory periodically during sampling, and N_{pages} is the number of

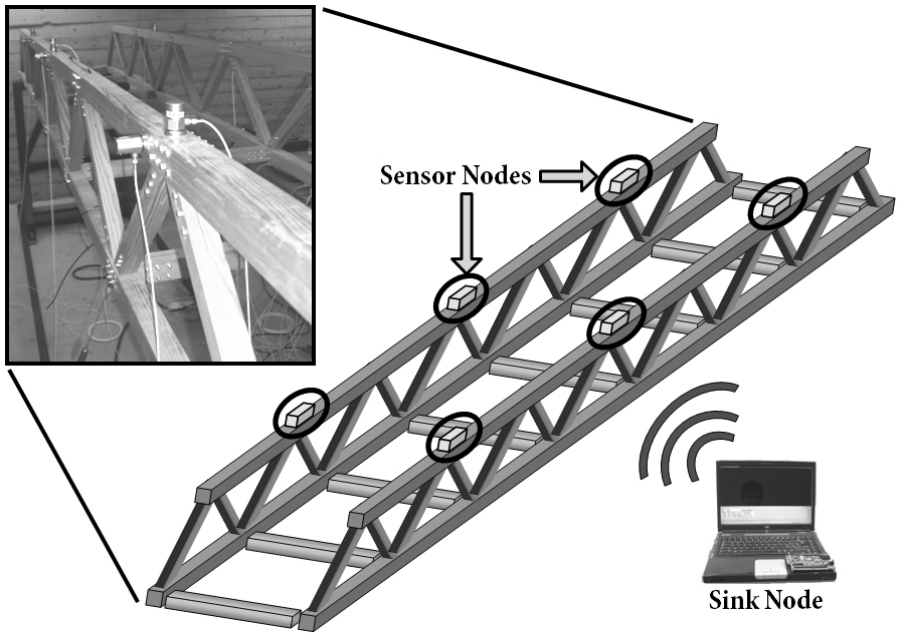

Figure 2. The bridge structure used in the tests. In black circles, the sensor nodes locations.

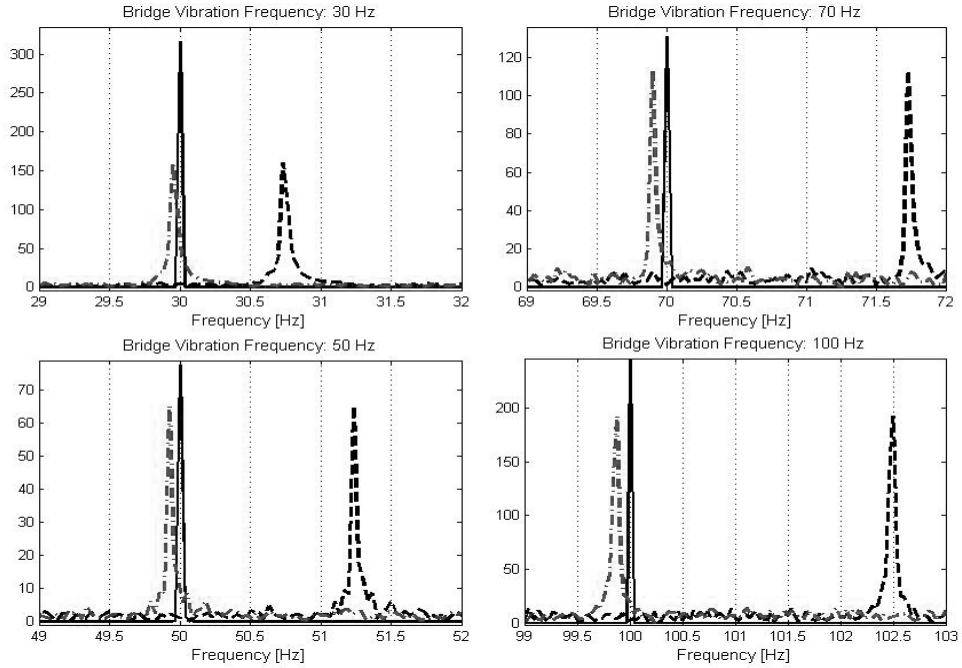

Figure 3. Comparison of the accuracy of the spectrums of the acceleration signals collected by the wired sensors (solid line), the wireless sensors (dashed line), and with the described adjustment of the length of the sampling period due to the writing in the FLASH memory (dash-dotted line).

pages written into the FLASH memory. Consequently, the actual sampling frequency, $Realf_s$, of the node becomes:

$$Realf_s = \frac{f_s \cdot T_s}{RealT_s} \tag{5}$$

Since $Realf_s$ is smaller than f_s, the peaks in the spectrums of the acceleration signals collected by the wireless nodes shift slightly to the left on the frequency axis (as in Figure 3), reducing the initial difference.

4 CONCLUSIONS

This paper describes a wireless sensor network for structural health monitoring. A wireless sensor node, equipped with a 3-axis digital accelerometer and a temperature and humidity sensor, is presented. The device can sample at frequencies up to 1 kHz, and exploits a 500 KB serial data FLASH memory for storing the acquired data before transmitting them to the sink node. The implemented wireless sensor network allows the end-user operating at the sink node to reconfigure key parameters of the application, such as which nodes perform the sampling, the sampling frequency, the length of the sampling period, the sensitivity of the accelerometer, and the vibration directions to be considered in the sampling. During the data transmission phase, the sink node establishes reserved communication links with the sensor nodes deployed on the structure. A retransmission procedure recovers those data packets originally lost during the transmission. A Matlab program running on an external PC connected to the sink node processes and visualizes in real-time the data received from the sensor nodes.

In a small-scale system deployment, the acceleration signals collected by the wireless sensor nodes are compared to those collected by high sensitivity wired accelerometers. The analysis of

the frequency spectrums shows that the wireless nodes provide accurate data allowing the identification of the frequencies of the vibration of the structure. We propose a method to compensate the error induced by the writing of the samples in the FLASH memory of the nodes during sampling. This error occurs only in the case the sampling frequencies are higher than approximately 290 Hz, as the FLASH writing operation takes 3.45 ms. The compensation method remarkably reduces the inaccuracy initially observed in the signals frequency domain representations.

In the near future, the created wireless sensor node will be developed by adding a piezoelectric and solar energy harvester. This system, when properly tuned, converts the vibrations of the structure into energy for the batteries, hence extending the life time of the nodes.

5 REFERENCES

Akkaya, K, and Younis, M. 2005. A survey of routing protocols in wireless sensor networks, Elsevier Ad Hoc Network Journal, Vol. 3, No. 3, 325-349.

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

Ceriotti, M, Mottola, L, Picco, G.P, Murphy, A.L, Guna, S, Corrà, M, Pozzi, M, Zonta, D, and Zanon, P. 2009. Monitoring heritage buildings with wireless sensor networks: the Torre Aquila deployment. *Proceedings of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN09)*, San Francisco, CA, USA, April 13-16.

Chen, B, and Tomizuka, M. 2008. OpenSHM: open architecture design of structural health monitoring software in wireless sensor nodes. *Proceedings of the IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA 2008)*, Beijing, China, October 12-15.

Cosar, E.I, Bocca, M, and Eriksson, L.M. 2009. High speed portable wireless data acquisition system for high data rate applications. *Proceedings of the IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA 2009), San Diego, CA, USA, August 30-September 2.*

Doebling, S.W, Farrar, C, and Prime, M.B. 1998. A summary review of vibration-based damage identification methods. *Shock and Vibration Digest*, Vol. 30, No. 2, 91-105.

Kim, S, Pakzad, S, Culler, D, Demmel, J, Fenves, G, Glaser, S, and Turon, M. 2007. Health monitoring of civil infrastructures using wireless sensor networks. *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, Cambridge, Massachusetts, USA, April 25-27.

Lynch, J.P, and Loh, K.J. 2006. A summary review of wireless sensors and sensor networks for structural health monitoring. *The Shock and Vibration Digest*, Vol. 38, No. 2, 91-128.

Lynch, J.P, Sundararajan, A, Law, K.H, Kiremidjian, A.S, and Carryer, E. 2004. Embedding damage detection algorithms in a wireless sensing unit for operational power effriciency. *Smart Materials and Structures*, Vol. 13, No. 4, 800-810.

Mizuno, Y, Monroig, E, and Fujino, Y. 2008. Wavelet decomposition-based approach for fast damage detection of civil structures. *Journal of Infrastructure Systems*, ASCE, Vol. 14, No. 1, 27-32.

Paek, J, Chintalapudi, K, Govindan, R, Caffrey, J, and Masri, S. 2005. A wireless sensor network for structural health monitoring: performance and experience. *Proceedings of the Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, Sydney, Australia, May 30-31.

Sohn, H, and Farrar, C. 2001. Damage diagnosis using time-series analysis of vibrating signals. *Smart Materials and Structures*, Vol. 10, No. 3, 446-451.

Sundararaman, B, Buy, U, and Kshemkalyani, A.D. 2005. Clock synchronization for wireless sensor networks: a survey. Elsevier Ad Hoc Network Journal, Vol. 3, No. 3, 281-323.

Xia, Y, Hao, H, Zanardo, G, and Deeks, A. 2006. Long term vibration monitoring of an RC slab: temperature and humidity effect, *Engineering Structures*, Vol. 28, No. 3, 441-452.

ACKNOWLEDGEMENTS

This work was supported by the MIDE technology program of the Helsinki University of Technology. The authors would like to thank the colleagues in the “Intelligent Structural Health Monitoring System (ISMO)” project, who contributed to this study.