Resource Consumption Analysis for a Structural Health Monitoring Algorithm Using Wireless Sensor Networks

Igor Leão dos Santos[1], Luci Pirmez[1], Érico Lemos[1], Luiz Vaz[1], Flávia C. Delicato[2], Neuman Souza[3]

[1]NCE – Universidade Federal do Rio de Janeiro (UFRJ) – Rio de Janeiro – RJ – Brasil

[2]Dimap – Universidade Federal do Rio Grande do Norte (UFRN) – Natal – RN – Brasil.

[3]Universidade Federal do Ceará (UFC) – Ceará – CE – Brasil

{igorlsantos,luci.pirmez,ericotl,fdelicato,neuman.souza}@gmail.com, vaz@peno.coppe.ufrj.br

Abstract: *This work presents a partial evaluation of a fully decentralized approach to detect, localize and determine the extent of damage on civil structures in environments that range from offshore oil and gas industry to wind farms. The evaluation presented in this work is a first step towards the conception of a fully decentralized algorithm for Structural Health Monitoring, whose key idea is to fully distribute the procedure of monitoring a structure among the nodes in a Wireless Sensor Network. Experimental results showed that the algorithm, still under development, performed well in terms of network lifetime, and proved that all the operations required by the algorithm can be efficiently provided by a Wireless Sensor Network.*

1. Introduction

The recent advances in wireless technologies and MEMS (Micro Electro Mechanical Systems) enabled the emergence of Wireless Sensor Networks (WSNs), whose devices have limited sensing, storage and processing capabilities. Since sensor nodes are battery-powered, energy consumption is a very important issue. So it is crucial that the protocols and algorithms for such devices are designed in an energy efficient way, thus reducing power consumption and maximizing the system lifetime.

Recently, there has been much interest in the use of WSNs [Hatler and Chi 2005], [Yick et al. 2008] in sectors of exploration and distribution of the oil and gas industry, as well as in the renewable energy sector, particularly in wind farms, with the purpose of Structural Health Monitoring (SHM) [Doebling et al. 1996], [Sohn et al. 2004]. The monitoring of physical structures enables damage prediction (fractures) and, therefore, to anticipate repairs, avoiding accidents. In applications built for that purpose, the sensing devices are used to perform measurements of the structure and the external events that affect the monitored structure, delivering such measures to a data collection station, the sink node. In this context, WSNs are used to remotely monitor structures and determine their physical integrity. The sensing devices typically used in WSNs for SHM applications, are, among others, strain gauges and accelerometers. The sensing devices, in general, collect the analog measurements from the environment and convert them into digital measurements. The structural information collected during the operation of the structure by sensor nodes can be handled inside these sensor nodes before being transmitted to the sinks through the wireless network.

The main motivations for using WSNs in a remote structural monitoring system are fivefold: (i) reduction of the need for experts (engineers) to perform tasks of *in sito* verification about the structural state; (ii) the possibility of dynamically changing the layout of the WSN; (iii) the ease and flexibility of installation of wireless sensor nodes, including areas where access is difficult or expensive; (iv) the easy and fast WSN reconfiguration; and (v) the lower cost of the system. Obviously, the use of wireless networks as the communication infrastructure to the structural monitoring system leads to a series of new challenges. For instance, a key challenge is how to adapt signal processing techniques, already available in SHM, to perform as much data compression within the network as possible, reducing the need for transmissions, extending the WSN lifetime.

This work represents a first step towards the conception of a fully decentralized approach to detect, locate and determine the extent of damage on structures from environments like offshore oil and gas industry and wind farms, making use of WSNs. The key idea of our approach is to distribute the procedure associated with the task of monitoring a structure among the sensor nodes in a WSN, so that only through collaboration among the cluster-heads it is possible to detect, locate and determine the extent of damage. Unlike other approaches [Kim et al. 2007], [Chintalapudi et al. 2006], all the SHM processing of our proposal runs inside the network without any help from the sink node. The network topology has two layers, the lower layer, containing ordinary sensor nodes in charge of sensing tasks, and the higher layer, containing cluster-heads (CHs). Our objective, at this current stage of development, is to investigate the actual performance of such fully decentralized approach in terms of communication and energy.

Among the contributions of our proposed approach stands out, in first place, the discussion and clarification of the main requisites of a fully decentralized monitoring system that makes use of a real WSN. In future works, this contribution may evolve to the presentation of an innovative monitoring system. There is still not much knowledge of fully decentralized and autonomous proposals in the SHM literature. Our proposal does not make use of analytical values of the structure's natural frequencies, taken from a finite element model and commonly used in most existing solutions in the literature. Rather, natural frequencies taken at the beginning of the structure's operation are used for comparison between the healthy and actual state of the structure. This choice is didactic, since our focus is on the networking issues. Explaining the structural engineering domain, which encompasses Finite Element Modeling (FEM) techniques and dynamic analysis, is not the focus of this work. Although our proposal may receive the first values of natural frequencies directly from the first sensing as well as from a finite element model, the didactic choice for the values from the beginning of the structure's operation is enough for understanding the overall idea. In this way we avoid the need to make an in-depth explanation of the structural engineering domain, but we also assume the possibility of a damage occurrence before the beginning of the structure's operation to be unnoticed. Finally, it is also important to mention that depending on the intensity of the damage site or its extent, real time actuators can be triggered in order to shut down equipments or fire alarms and so on.

The remainder of this paper is divided as following. Section 2 depicts related works. Section 3 presents the proposed algorithm, SENSOR-SHM. Section 4 details the experiments performed to evaluate SENSOR-SHM algorithm and the obtained results. Finally, section 5 concludes this work and draws future directions.

2. Related Work

Several researches suggesting a partial decentralization of the procedure for monitoring structures based on already established techniques in SHM can be found in the literature and are presented in this section. In such works, the monitoring procedure is performed partially by the network sensor nodes and partially by the sink. In this section, we also intend to clarify the reason for relying on decentralization in our work. The main reasons are, in short: reduction of the communication overhead and, as one of the consequences, energy saving. The Fast Fourier Transform (FFT) is a powerful technique for reducing the need for transmissions, and its calculation by the sensor nodes has proved to be possible in many works. For instance, in [Hackmann et al. 2008] the need for transmissions is reduced from 2048 floats to 5 floats, and in [Chintalapudi et al. 2006], the reduction is around 99%.

In [Caffrey et al. 2004] a sensing technique is defined, which uses electrodynamic shakers to produce vibrations on the structure and accelerometers in the sensor nodes to collect data for a few seconds in order to measure these vibrations. In order to perform a structural analysis, a FFT is applied to the acceleration data collected in each sensor node, transforming the signal in time domain to a signal in frequency domain. Then, the power spectrum is analyzed and the frequencies of the structure modes of vibration whose values correspond to the energy peaks of the spectrum are extracted. Then, the sensor nodes elect a sensor node that is responsible for obtaining a more accurate result by aggregating all measures of modal frequencies and their associated energies extracted from all network nodes. Finally, this aggregated result is sent to the sink and then the frequency variation analysis can be done. Our proposal differs from this one since the frequency variation analysis is performed within the network, with the collaboration of cluster-heads.

In [Chintalapudi et al. 2006], many methods to detect damage on the structure are discussed. Most studies in the literature are inspired in these last two methods, including our proposal. One of the methods discussed in [Chintalapudi et al. 2006] consists in the methodology for damage detection in basic structures that makes use of the structure's signature variation, and it is widely accepted. This variation is seen by comparing signatures obtained when the structure was sound and when the structure is damaged. The results of applying this methodology are found in several works, among which we cite [Cawley and Adams 1979], [Messina et al. 1996] and [Contursi et al. 1998], where an evolution of the correlation based techniques for locating damage can be observed. The essence of these methods is to correlate sets of theoretical ideal frequencies with a set of frequencies experimentally obtained. In [Cawley and Adams 1979], one of the first studies based on the concept of linear correlation to detect damage in structures is developed, in which metrics to locate damage based on the variation of the structure's natural frequencies are presented. An application of the principles of the technique developed in [Cawley and Adams 1979] is found in [Messina et al. 1996], in which a similar method called "Damage Location Assurance Criterion" (DLAC) is presented. This method measures the degree of correlation between an experimental vector with frequencies' variation rates, and several vectors with analytical frequencies' variation rates. The vectors, both experimental and those obtained through analytical models, should contain information on the frequencies of the first modes of vibration of the structure. The number of vibration modes depends on the depth of the performed analysis. These analytical vectors come from a finite element model of the structure, each of them relative to damage present in different positions. The most perfect correlation between the experimental vector and an analytic vector reveals the location of the damage in the structure. This method allows the location of only one site of damage on the structure, being the damage represented by a mass loss, through a cross sectional area reduction to an element that composes the basic structure used in the experiment. To locate multiple damage sites in the structure, a method called "Multiple Damage Location Assurance Criterion (MDLAC) is proposed in [Contursi et al. 1998]. Unlike DLAC algorithm [Messina et al. 1996] [Contursi et al. 1998], in which analytical values of the structure natural frequencies taken from a finite element model are used, the system proposed in our work uses $\omega_{i,0}$ values, whose meanings are described ahead, in Section 3. Another difference is that our algorithm runs on the sensors and the DLAC algorithm runs on the sink.

The proposal described in [Hackmann et al. 2008] makes use of a WSN to monitor structural conditions. The performed study resulted in the emergence of new challenges for WSNs, since these networks are used in the proposal not only as an infrastructure to carry information from the SHM application, but also the sensor nodes incorporate part of the application core. The cited work proposes a partially decentralized algorithm to be used along with the DLAC method. In the algorithm, the data is collected and partially processed by the sensor nodes, inside the network (the so called *in-network* processing). In the sink node, the frequency values that compose the signature of the structure are extracted by solving a mathematical equation expressing a curve that fits the resultant curve of a FFT. After, the DLAC algorithm runs on the sink, taking the sensed data relative to the structural frequency response as input, and data relative the same responses of an analytical model, developed through a finite element modeling to detect and locate damage. The partial decentralization of the procedure for damage detection, allowing the sensor nodes to act on the collected data, significantly reduces energy consumption since it minimizes the number of needed transmissions. However, damage detection and localization through the use of the DLAC method, was still centrally held, in the sink node. The algorithm for damage detection proposed in our work is mainly inspired by the work presented in [Hackmann et al. 2008]. One of the differences between our proposal and the work in [Hackmann et al. 2008] is that in our solution the whole procedure of extracting the frequency values from the power spectrum (in [Hackmann et al. 2008] conducted by the Curve Fitting stage) is performed on the sensors. Unlike other algorithms proposed in the literature, all the SHM processing of our algorithm is performed on the sensor and cluster-head nodes, without the help of the sink node. And through collaboration among cluster-head nodes, it is possible detect, localize, and determine the extent of damage.

3. SENSOR-SHM Description

The description of the decentralized methodology used in the SHM system proposed in this article is divided into two procedures. Firstly, a setup procedure is performed, which consists in setting the algorithm initial parameters before making the application deployment, ie, before installing the program on the sensors and allocating the sensors on their fixed positions in the structure. The second procedure consists in running the algorithm common operation cycle, and encompasses several stages of data collection.

3.1. Setup procedure

The setup procedure encompasses several activities of setting parameters of the algorithm that are needed in order to start monitoring the structure. The structural monitoring is performed in a periodic basis, and each monitoring cycle is based on a collected signature sample. Since the signature is extracted from the acceleration signals, the data collection stage is also an acceleration data processing stage, and since this is the operation that takes the longest time to complete, the common operation cycle of the algorithm that ends with detecting, localizing and determining the extent of possible damage after processing the collected signature, is called a **data collection stage**. In the setup procedure all the parameters needed for purposes of identification and configuration during the data collection stage are set.

A **data collection stage** is identified by an integer t, which is incremented by one for each performed data collection stage. These stages start at a given time, defined by the sink node. The **collection period**, that represents the duration of each data collection stage, is defined as being long enough to collect 512 acceleration samples at a sampling rate of 1.0 kHz, which means the collection period will last for about 500 milliseconds. The **number of collected samples** is determined by the following criteria: (i) it must be enough to ensure a good resolution in the power spectrum that will be returned, which implies in better precision in the modal frequencies determination, (ii) it must be a power of 2, since this is a requirement for the entry of data in the FFT algorithm, and (iii) it shall not exceed the sensors' storage capacity (Flash memory). It is important to mention that the amount of data collected by the sensing module is a determinant of energy consumption. The **sampling rate** is set according to the following criteria: (i) it must be greater than the value of the first modal frequencies of interest so that these are shown in the power spectrum (commonly, values below 200Hz for the first five modal frequencies of structures are expected), (ii) it must be high enough to ensure accuracy, (iii) it should be twice the highest modal frequency of interest, to meet the Nyquist criterion. In [Kim et al. 2007] a sampling rate of 1.0 kHz is used, with hardware similar to the used in our work, showing that it is possible to achieve our proposed sampling rate.

Other constant parameters must be set, such as the arrays of cluster-head neighbors that inform each cluster-head who are its neighbors. Neighbors of a cluster-head are nodes allowed to communicate in order to accomplish the tasks related to damage localization and extent determination. Also, the arrays of sensors that are subordinated to each cluster-head must be manually set. At this time, the sensors are aware of who are their respective cluster-heads, and each cluster-head knows the sensors pertaining to its clusters as well as the other cluster-heads in their immediate neighborhood, thus all the necessary communications can be easily established and a static network is characterized. The sets of existing sensors and cluster-heads are properly defined as a collection J of Z cluster-heads where each cluster-head is identified by j = {1, 2, ..., Z}, and has a subset of subordinated sensors. All the subsets of sensors subordinated to each cluster-head are part of another collection defined by I, that includes all the N sensors in the network, where each sensor is identified by i = {1, 2, ..., N}. These definitions are made according to the roles of each node in the network. Constants like T_i, L_j and A_i must also be set before the deployment, although they can be changed during the network operation. The use of these constants is better explained in the following subsections. These constants are stored in the cluster-heads.

After all these settings, the network can be physically deployed over the sensing area (the chosen structure), and the nodes can be installed in their fixed positions. First, cluster-head nodes are physically installed and turned on, in any order. Then, sensor nodes are physically installed and turned on, and as part of their initialization process, they collect the initial signature of the structure, $\omega_{i,0}$. It is supposed that at this time the structure is at the beginning of its operation. Thus, each sensor generates its $\omega_{i,0}$ vector and transmits it to its cluster-head (CH). These initial values will be used as a reference for the undamaged (healthy) structure. Then, sensors enter in sleep mode and wait for the next data collection stage, that will be identified by a value of t = 1. Data collection stages are described in the following subsections, and basically consist of sensing data, processing and extracting the measured frequencies, forwarding them to the cluster-heads, further process the data inside the CH nodes, first individually by each CH and after through collaboration, among the

CHs nodes, and finally starting actuators or sounding alarms, keeping the sink node aware of the presence of damage (whenever it occurs) in the structure, sending reports.

3.2. Data collection stages from the sensors viewpoint

The data collection stage starts when requested by the sink node. A message is sent from the sink to the CHs, and those are responsible for sending messages to schedule the next sensing task on their subordinated sensors. Then, when the specified time comes, the sensors start collecting data. In order to save memory and energy the sensors do not keep any data about the current data collection stage. That means the sensors are not aware of the value of t, and this information is stored at the CHs.

So, at every data collection stage t, each sensor node is responsible for sensing the structure, collecting the acceleration in the time domain at its relative position. After each sensor collects the acceleration signals, it performs a FFT on the respective collected signals. Then a simple method is applied to extract the modal frequencies in the power spectrum generated in the previous step. This method has the same goal of the Polynomial Curve Fitting seen in [Hackmann et al. 2008], but is simpler and less accurate, leading to more errors. This lack of accuracy can be compensated by increasing the number of sensors in each cluster, generating more redundant data. However, this method is much less expensive in terms of energy, and is able to be fully implemented within the sensors. These frequency values extracted from the power spectrum generated by the FFT, assuming that the noise is very low, are related to the modal frequencies of the structure. The frequency values obtained by each sensor, considering the used sampling rate, are the first peaks of the power spectrum returned by the FFT, and will compose the signature of the structure for that sensor.

Formalizing, "$\omega_{i,t}$" is a signature of a given structure, acquired at a data collection stage "t", and represented by a vector of sensed frequencies in a sensor identified by "i". Therefore, different sensors get different values of signatures for the structure, depending on their location and instant of time in which the data collection stage started. Thus, vectors "$\omega_{i,t}$" will be obtained for each sensor i, each of them with M frequency values in each data collection stage t. For the first 5 modal frequencies (M = 5), the vector has the form as described in Equation 1 (see Section 3.3.1).

Finally, all vectors generated by each sensor at each data collection stage t are sent to their respective cluster-heads. It is important to mention that the initial signature of the structure $\omega_{i,0}$ is obtained for each sensor i. The initial signature is obtained at the beginning of the structure's operation. Thus, each sensor generates its $\omega_{i,0}$ vector and transmits it to its cluster-head. These initial values will be used as a reference for the undamaged state of the structure. Refer to Section 1 for understanding the use of the values of $\omega_{i,0}$ as the values taken at the beginning of the structure's operation. Also, the reasons for the sensors to transmit data to the CHs are: (i) reduction of energy consumption and (ii) reduction of false positive occurrence, since the CHs can evaluate and compare the results of their neighbors.

3.3. Data collection stages from the cluster-heads viewpoint

This section details the procedures of data collection stages from the CH viewpoint.

3.3.1. Damage detection

Once the CH has the $\omega_{i,0}$ vectors from all its subordinated sensor nodes, it is responsible for comparing these and the subsequent $\omega_{i,t}$ vectors, generated in a similar way as $\omega_{i,0}$ in the subsequent data collection stages. If any of the sensor nodes send to their CH a value of $\omega_{i,t}$

that differs (considering certain T_i tolerance) from the value of $\omega_{i,0}$ (original signature), the structure may have changed temporarily due to some external event, or may be damaged. If this discrepancy persists in further data collection stages, the structural change is considered permanent. It is important to mention that each sensor is put in sleep mode, after completing its sensing task and it is normally woken up at the next data collection stage, or whenever requested by the sink.

The verification of change in the modal frequencies of a structure is performed by comparing $\omega_{i,0}$ and $\omega_{i,t}$ vectors. The comparison is done by the absolute value of the difference between $\omega_{i,0}$ and $\omega_{i,t}$, and its result is stored in the $\Delta\omega_{i,t}$ vector, as seen in Equation 2.

$$\omega_{i,t} = \begin{bmatrix} \omega_{i,t}^1 \\ \omega_{i,t}^2 \\ \omega_{i,t}^3 \\ \omega_{i,t}^4 \\ \omega_{i,t}^5 \end{bmatrix} \quad [1] \qquad \Delta\omega_{i,t} = \left|\omega_{i,0} - \omega_{i,t}\right| = \begin{bmatrix} \left|\omega_{i,0}^1 - \omega_{i,t}^1\right| \\ \left|\omega_{i,0}^2 - \omega_{i,t}^2\right| \\ \left|\omega_{i,0}^3 - \omega_{i,t}^3\right| \\ \left|\omega_{i,0}^4 - \omega_{i,t}^4\right| \\ \left|\omega_{i,0}^5 - \omega_{i,t}^5\right| \end{bmatrix} = \begin{bmatrix} \Delta\omega_{i,t}^1 \\ \Delta\omega_{i,t}^2 \\ \Delta\omega_{i,t}^3 \\ \Delta\omega_{i,t}^4 \\ \Delta\omega_{i,t}^5 \end{bmatrix} \quad [2]$$

The CH, through the analysis of the $\Delta\omega_{i,t}$ vector from each sensor i, can detect whether there has been any variation in the signature structure for that sensor. The ideal situation is identified when the $\Delta\omega_{i,t}$ vector has values close to a null vector, which corresponds to the situation where there is no variation in the structure's signature. For $\Delta\omega_{i,t}$ values that are different of the null vector, the CH can assume, considering a given T_i comparison tolerance, that there has been a significant change in the structure, which may mean the presence of damage on it. If a value from one of the $\Delta\omega_{i,t}$ vector positions exceed its tolerance for a sensor i, the CH proceeds to the next step in the monitoring process, which refers to the damage location, since it has detected an abnormal condition in the structure. It is important to mention that the T_i vectors are determined for each sensor based on knowledge and analysis of the localities in which each sensor will be installed. Also, the T_i vector can be statistically determined after making a series of experimental samples. The purpose of adopting a tolerance value is to prevent small random disturbances, which do not imply the occurrence of abnormal conditions, from being considered by the monitoring procedure as such abnormal conditions. So, if no values of the $\Delta\omega_{i,t}$ vector exceeding the T_i tolerance are detected, the monitoring procedure does not perform the next steps, instead it is interrupted right at this moment, waiting for the next data collection stage. This procedure reduces the energy consumption since it avoids the waste of energy from the following unperformed steps. In short, for each sensor i, the CH performs the comparison between the $\Delta\omega_{i,t}$ vector and the T_i vector, for each of the first five positions, relative to the modes of vibration.

At this damage detection step it is possible to send a report to the sink, if the goal is only to detect the presence of abnormal structural states. This report should contain the $\Delta\omega_{i,t}$ vector from the sensors that are out of the tolerances, and also the values of the $\Delta\omega_{i,t}$ vectors from all the sensors that belong to the cluster in question, enabling the scenery reconstitution and further analysis in the sink.

3.3.2. Damage localization and extent determination

The presence of damage in a structure can affect both the higher and the lower frequencies in a given sensor, depending on, respectively, if the sensor is positioned close to the damage or not, as assumed in [Kim et al. 2007]. By knowing that changes in the higher modal frequencies mean changes in local vibration modes, each CH analyzes the $\Delta\omega_{i,t}$ vectors of all sensors located in its cluster in search of these kinds of change. In each CH and for each data collection stage t, this analysis is aided by using the $D_{i,t}$ coefficient, which is calculated for each sensor i that has exceeded the given tolerance, in the given cluster, as described in Equation 3.

The $D_{i,t}$ coefficient is set so that its value should indicate how close the sensor i is from the damaged site, and A_i is a vector of weights, assigned to each modal frequency. In order to identify the sensors that are closest to the damage site, higher values to the weights associated with the higher modal frequencies can be assigned. The weights will also be distributed according to the characteristics of the structure and specific locations of the deployed sensors, also considering relevant to the analysis the variation of modal frequencies that are not the highest. Such weights should be as much as possible the same for each sensor i, so that the $D_{i,t}$ coefficient can be used with greater relevance to future comparisons among other coefficients extracted in the same way in other clusters from the network.

$$D_{i,t} = A_i^T \Delta\omega_{i,t} = \begin{bmatrix} A_i^1 & A_i^2 & A_i^3 & A_i^4 & A_i^5 \end{bmatrix} \begin{bmatrix} \Delta\omega_{i,t}^1 \\ \Delta\omega_{i,t}^2 \\ \Delta\omega_{i,t}^3 \\ \Delta\omega_{i,t}^4 \\ \Delta\omega_{i,t}^5 \end{bmatrix} \quad [3] \qquad C_{j,t} = \sum_{i=1}^{k} D_{i,t} \quad [4]$$

To sum up, the weights should be assigned so that the sensors belonging to a given cluster located near the damage site obtain the highest $D_{i,t}$ values of the whole network. In other clusters, the $D_{i,t}$ values are smaller, but are still different from zero, since the lower frequency values will be changed and these changes are identified by many other sensors. In the following step, $D_{i,t}$ coefficients are aggregated in each cluster j. The aggregation is made by summing all $D_{i,t}$ coefficients for all k sensors in the cluster j, resulting in a $C_{j,t}$ coefficient, as in Equation 4. By its mathematical definition, the $C_{j,t}$ coefficient is an indicator of how close to the damage the cluster as a whole is. The algorithm makes use of this indicator to locate and determine the damage extent.

In our algorithm of damage localization and extent determination, each CH node compares its $C_{j,t}$ value with a L_j tolerance. When the $C_{j,t}$ value exceeds the tolerance, the CH node should send a message informing its $C_{j,t}$ value to its neighbor CH nodes. The L_j tolerance is defined for each CH, in a similar way to the determination of the T_i tolerances. Nodes and clusters in places where the physical characteristics lead to inaccuracies in determining the $D_{i,t}$ coefficients, and therefore in determining the $C_{j,t}$ coefficients, should consider this into its tolerance choices, which is a way to avoid false positives in the data collection stages. The tolerance values depend on the structural characteristics, and therefore should be determined by an expert in the structure, and through statistical analysis.

After the CH j transmits its $C_{j,t}$ value to its immediate CH neighbors, it is expected that some of these neighbors also have exceeded their tolerances, and thus have sent their respective $C_{j,t}$ values to their neighbors. All CHs who receive $C_{j,t}$ values from other CHs will compare these values with theirs. In a given neighborhood, the CH who has the greatest $C_{j,t}$ value assumes the role of a "collector". The collector node is responsible for two tasks: (i) aggregate the information about the $\omega_{i,t}$ values from all its neighboring CHs and build a report to be sent to the sink, issuing a warning and (ii) act on the environment around it by triggering a relay, aiming to prevent the progression damage and avoiding further problems in the locality.

To build the report that will be sent to the sink, the values of $\omega_{i,t}$ vectors at that data collection stage were chosen, because from these values it is possible to deduce all the other relevant information. Since the sink node has knowledge of $\omega_{i,0}$ vectors, and all the weight and tolerance values, it is possible to calculate all the other related values that were shown before in this explanation, and still have a global view of the events that happened into the network during the data collection stage t.

It is assumed that the damage location and extent are determined by the positions of the sensors that are CHs and whose $C_{j,t}$ value exceeded the L_j tolerance at that moment, that is, those who sent their $C_{j,t}$ values and also received $C_{j,t}$ values from its neighbors. The CH action is immediate, as soon as the damage is detected in its vicinity, and may prevent further problems. In case of multiple damage sites, or large damage sites that cover a large area on the structure, the trend is that there will be many emerging collectors, and multiple reports from different locations will arrive at the sink node.

4. Experiments with SENSOR-SHM

This section describes the experiments conducted to evaluate the performance of our algorithm. Such performance can be evaluated in two main ways: (i) the algorithm capability to precisely and accurately detect, localize and determine the extent of damage sites, and (ii) the algorithm efficiency when running within a resource constrained WSN, in terms of energy and processing consumption. In this work, we present the algorithm evaluation in terms of resources consumption (ii), while item (i) is not evaluated, since it requires a specific hardware that is not currently available for us.

To assess the actual performance of our SENSOR-SHM algorithm on real sensor hardware, we prototyped the sensor network scenario using the MICAz motes from Crossbow Technology. The motes are programmed in nesC under the TinyOS 2.1 development environment [Hill et al. 2000]. NesC language embodies the structuring concepts and execution model of TinyOS, an event-driven operating system designed for sensor nodes. It adopts a component-based architecture which enables rapid development while minimizing code size.

To implement our SENSOR-SHM algorithm, two programs were built. One for running in the motes assigned as CHs and other for running in the motes acting as ordinary sensor nodes. It is important to note that no additional protocol, for instance, cluster formation or routing protocols, was used in this application design, since it may influence in the energy and communication performance of the network. Our experiments aim at evaluating the performance of the Sensor-SHM algorithm by itself.

4.1. Methodology

The methodology used in the experiments is now briefly described. A set of experiments in real hardware was performed to evaluate the Sensor-SHM algorithm in terms of communication, and the same set of experiments was repeated in a virtual simulation, to evaluate Sensor-SHM in terms of energy and also communication. The virtual environment was chosen to evaluate energy consumption since it is the most useful way to achieve such measurements, once currently there is not a clear way of doing so in TinyOS 2.1 and measuring energy directly on each mote through multimeters would demand a large amount of work and tools in topologies with many sensors. Using a uniform energy model in the simulation, it is possible to estimate energy parameters for all nodes at the same time. The communication was mainly evaluated through real tests, but the simulation results could also present an output of communication parameters, so it was used to predict what would be expected in the real experiments.

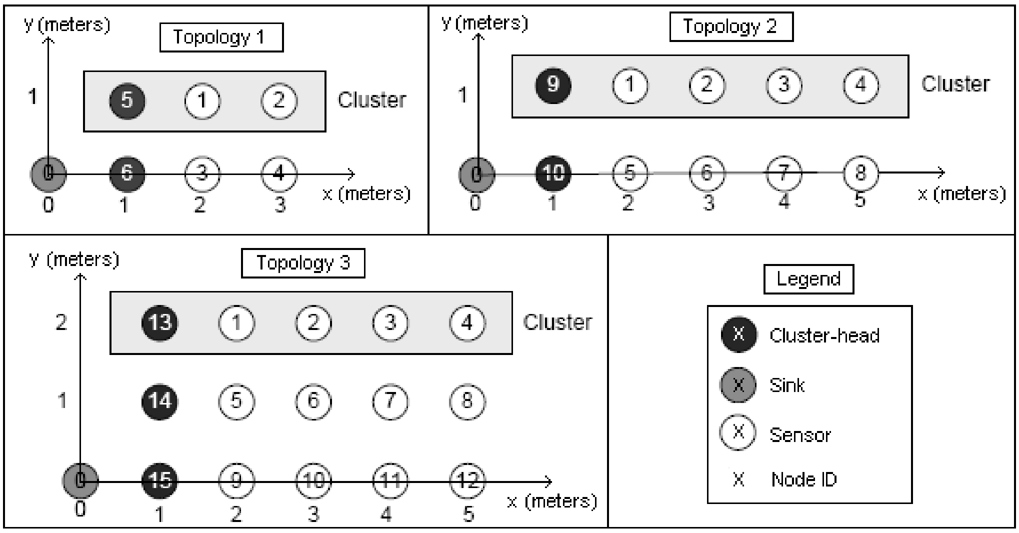

Figure 1. The three simulated topologies

The set of experiments consists of three different topologies, described in Figure 1. The virtual simulations were performed using the open-source wireless network simulator Avrora [Titzer et al. 2005], with an extension named AvroraZ, for modeling energy and communication with the MICAz platform [Alberola and Pesch 2008]. The energy model adopted in our experiments is described in these works. Many works, as for instance [Jin and Gupta 2008], make use of this simulator. The real tests were performed with MICAz motes without any sensor board, only the motes themselves, once the data sampled was simulated in both the real and virtual experiments.

The acceleration values collected by the sensors were simulated, and were the same at every data collection stages. Five summed sinusoids with known frequencies made every sensor return modal frequencies of 20Hz, 40Hz, 60Hz, 80Hz and 100Hz. During the data collection stages, each sensor generates a random error of around ±2Hz in the determination of the modal frequencies, only relative to calculations and truncations, necessary to transmit the data over the radio. All the mentioned limits in the algorithm were set to zero, so all the data sensing stages were considered as having found damage, what is the scenario considered as the most resource consumptive.

In both virtual and real scenarios, nodes are placed in a grid topology, commonly used in SHM applications since the node placement in these applications is made through a fixed deterministic scheme. In this grid, nodes are spaced 1m from each other, starting from the position (0,0) of the sink, as in the Cartesian plan in Figure 1, which allows a better understanding of the experimental scenarios. The height of each node from the ground is considered as zero.

In the first topology, two clusters with two sensors in each are used. In the second topology, the number of sensors per cluster is increased by two, keeping the same number of CHs. In the third topology, the number of clusters is increased by one, keeping the same number of sensors per cluster. In each one of the three topologies, all the CHs are considered to be neighbors among themselves.

For the virtual simulation, specifically, the procedure is to start a data collection stage, send a message to each CH and wait for the data collection stage completion. Once it is completed, another data collection stage is started, and so on. During the simulations, it was found that each data collection stage takes around 10 seconds to complete, and the number of sensors in the network is not very significant to influence this time. Then, the chosen time between each data collection stage was 15 seconds (the time which the sink takes between each sent message, to start a data collection stage), to assure that all data collection stages are finished and, at the same time, to minimize the idle time between each data collection stage. So, at every 15 seconds the sink sends a message for each CH to start a data collection stage. Concerning to the battery, it was assumed the use of a battery of 40 Joules for each node. A common energy source of the real MICAz motes comes from two AA batteries, which are considered to be able to provide 16000 Joules, as estimated from the results provided by [Krämer and Geraldy 2006]. But, to keep the simulations short, the value of 40 Joules was chosen.

For the real experiments, the time between each data collection stage was 15 seconds and 15 data collection stages were performed. This number is the smallest number of data collection stages that one of the virtually simulated topologies survived. So, it was assured that the energy in the two AA batteries of each sensor was enough to perform the real tests. Moreover, for both the virtual and real experiments, the maximum message size is 28 bytes, the radio data rate is 19.2 kbps. No sleep mode was implemented, so the duty cycle is 100% since the experiments will be short and the amount of energy spent during the idle time is very low.

The WSN performance can be evaluated according to different metrics. In our analysis we chose (i) the percentage of data messages and control messages among all the messages exchanged in the network, and (ii) the network lifetime, measured in number of data collection stages. There are two kinds of ***Data messages*** in our implementation. One kind carries information on the values of natural frequencies between the CHs and the sensors, and a different kind of data message carries information on the values of C_{jt} exchanged among CHs. Each data message used to transport the values of natural frequencies requires four bytes per frequency in its payload. As we use 5 natural frequencies, this kind of data message has 20bytes of payload. And data messages used to transport the value of C_{jt} require 4bytes in its payload. ***Control messages*** are used: (i) to establish communication among the nodes avoiding communication problems like packet colision and packet loss and (ii) to start the data collection stages. The control messages have empty payloads. Both data and control messages require 8bytes for their headers. The energy consumption has a direct consequence over the ***network lifetime***, thus in the WSN performance. This is also a crucial issue for several application domains, other than SHM, that have the network lifetime as main requirement, as for instance environmental monitoring applications. All results are based on what happened during the network lifetime, which is considered to be the time interval from the deployment until the first node dies. In the three performed simulations, the first node to reach the zero value in its battery was always the CH, as expected, since it is the type of node in the network which makes the most extensive use of its radio.

4.2. Results and Analysis

In the first topology, the network lifetime was around 12,2 minutes, in the second 7,6 minutes, and in the third was 5,2 minutes. The number of data collection stages for a battery of 16000 Joules is estimated as being respectively 19520, 12160 and 8320 for topologies 1, 2 and 3. Our results show that if each data collection stage were 1 hour spaced in time from each other, and implementing a sleep mode during this time, the network lifetime could be estimated as being around 800 days in topology 1, 500 days in topology 2 and 350 days in topology 3. These conclusions are based on the simulation results, and also on the AvroraZ energy model. The battery used in the real tests was enough to perform more than 15 data collection stages in each topology.

In [Hackmann et al. 2008], where a decentralized approach based on DLAC method was presented, the achieved projection for the network lifetime was 213 days on an Imote2 platform with 3 AAA batteries (2400mAh) and 7 nodes in the network performing sensing tasks, scheduling samples once an hour and assuming a sleep time between each sampling. In Sensor-SHM experiments, the most similar scenario to the one described in [Hackmann et al. 2008] would be topology 2. In this topology, a network lifetime of 500 days was projected, performing the data collection stages once an hour, using the MICAz platform with 2 AA batteries (1202mAh) (a more resource constrained platform than the Imote2 used in [Hackmann et al. 2008]) and a total of 8 sensors and 2 CHs in the network. Also, the main determinant of the network lifetime was the battery of the CHs, since these nodes were the first ones to have their energy source depleted in all the three topologies. It means that the CH lifetime is 500 days long, and the other sensors are expected to last for more time (3 times more, as graphically estimated from the results in Figure 2, since the radio is the most energy consumptive resource [Alberola and Pesch 2008], and considering the number of bytes in each message). Then, using the localized approach of the Sensor-SHM algorithm instead of the approach proposed in [Hackmann et al. 2008] can be more advantageous, in terms of resource consumption, once the number of sensors per cluster is wisely chosen. A large number of clusters encompassing few sensors is a better choice when using Sensor-SHM, in terms of energy saving, since in topologies 2 and 3 the CHs are sending and receiving almost the same number of messages, in contrast with topology 1, with less sensors. This can be seen in Figure 2.

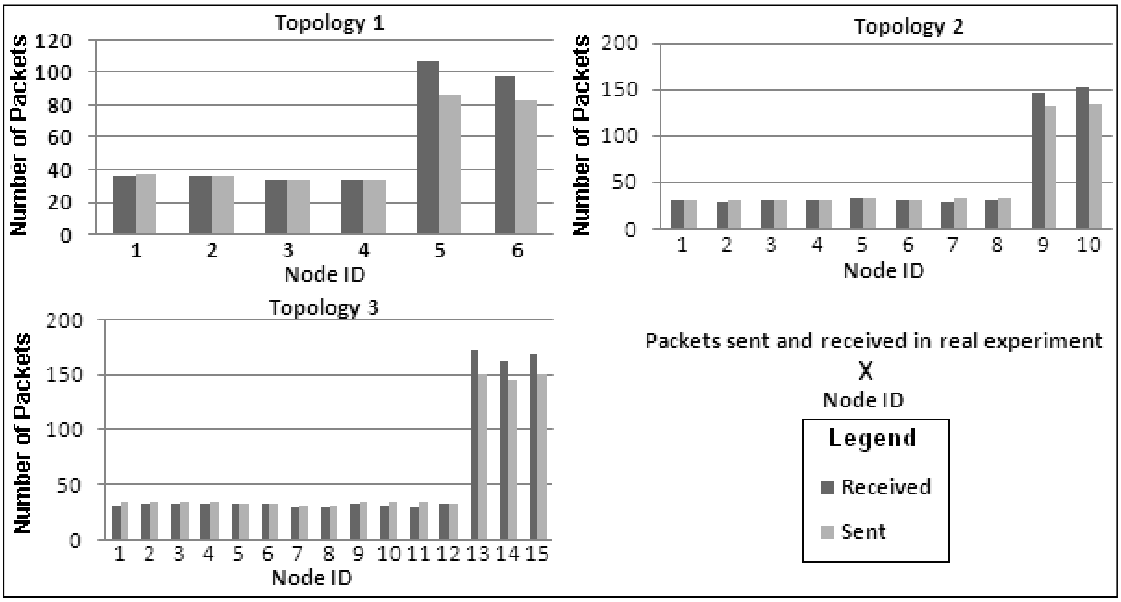

Figure 2. Number of sent and received packets per Node ID and topology in real experiment

Results presented in Figure 2 are related to the efficience of the algorithm in terms of communication. This efficience was analysed in terms of the minimal amount of data and control messages needed to use the Sensor-SHM algorithm in a real network. The Node IDs 5 and 6 in topology 1, the Node IDs 9 and 10 in topology 2 and the Node IDs 12, 13 and 14 in topology 3 represent CHs. It is clear that these nodes are the ones which make the most extensive use of the radio. All the sensors were expected to have the same number of exchanged packets among themselves, as also all the CHs. The small disturbance is explained by some eventual packets that were not perceived by the sink, which was listenning to all the network traffic, with all the nodes in its range.

5. Conclusion

The preliminary tests on the performance of the Sensor-SHM algorithm achieved good results, succeeding in proving that a distributed SHM application can reliably run under a WSN resource constrained environment. The resource limitation of such environments remains as one of the greatest drawbacks to the widely acceptance of the use of wireless sensor networks in the structural engineering field. The node platform used in this work was MICAz, which is considered as one of the most basic platforms commercially available, in terms of energy and computing resources. Some other platforms are considered as be better suited for this kind of application (SHM), as the Intel Imote2, which presents more computing and energy resources.

Performing the data collection stages once a day during one year leads to a better utilization of the monitored resources, like wind turbines in wind farms. Sensor-SHM can be used as a tool for predictive maintenance techniques, and its benefits are (i) improved energy offer by increasing the duty cycle of the monitored wind turbines, avoiding break down periods, (ii) reduction in maintenance costs, since the other available monitoring solutions, such as the ones based on wired networks and human intervention, tend to be more expensive, (iii) reduction in indemnity costs to repair damage to other properties around the wind farm, caused by a severe damage propagation at an aerogenerator (iv) increased safety also for the workers in the wind farms.

As future works we will test the Sensor-SHM algorithm in other platforms to compare their performance, as well as testing the Sensor-SHM algorithm in a real testbed aiming at evaluating the accuracy and precision of damage detection, localization and extension determination. Also, we intend to perform an optimization of the designed application prototype, including the use of auxiliary protocols.

Acknowledgements

This work is partly supported by the National Council for Scientific and Technological Development through processes 477226/2007-8 for Luci Pirmez and Flávia C. Delicato; 477229/2009-3 and 306938/2008-1 for Flávia C. Delicato; 481638/2007 for Neuman Souza, Luci Pirmez and Flávia C. Delicato; and 1979/09 for Luiz Vaz and Luci Pirmez.

References

Alberola, R. P., Pesch, D. (2008), “AvroraZ: extending Avrora with an IEEE 802.15.4 compliant radio chip model”, In Procs. of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks.

Caffrey, J., et al. (2004), “Networked Sensing for Structural Health Monitoring”, Center for Embedded Network Sensing. Papers. Paper 524.

Cawley, P., Adams, R. D. (1979), “The localization of defects in structures from measurements of natural frequencies”, Journal of Strain Analysis, Vol. 14, pp. 49-57.

Chintalapudi, K., et al. (2006), “Monitoring Civil Structures with a Wireless Sensor Network”, IEEE Internet Computing, 10, 2, p. 26-34.

Contursi, T., Messina, A., Williams, E. J. (1998), “A multiple-damage location assurance criterion based on natural frequency changes”, Journal of Vibration and Control, No. 4, pp 619-633.

Crossbow Technology Inc. (2009), “MICAz Specs.” Available at: <http://www.xbow.com/>, last access: 13 Dec. 2009.

Doebling, S. W., et al. (1996), “Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in their Vibration Characteristics: A Literature Review”, Report No. LA-13070-MS, Los Alamos National Laboratory, Los Alamos, NM.

Hackmann, G., et al. (2008), “A Holistic Approach to Decentralized Structural Damage Localization Using Wireless Sensor Networks”, RTSS, pp.35-46, Real-Time Systems Symposium.

Hatler, M., Chi, C. (2005), “Wireless Sensor Networks: Growing Markets, Accelerating Demand”, On World Report, available at <http://www.onworld.com/html/wirelesssensorsrprt2.htm>, last access: 13 Dec. 2009.

Hill, J., et al. (2000), “System architecture directions for networked sensors”, In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems. Cambridge, MA. 93–104.

Jin, Z., Gupta, R. (2008), “Improved distributed simulation of sensor networks based on sensor node sleep time”, In: Nikoletseas, S. E., Chlebus, B. S., Johnson, D. B., Krishnamachari, B. (eds.). LNCS, v5067, pp. 204-218. Springer, Heidelberg.

Kim, S., et al. (2007), “Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks”, IPSN’07, April 25-27, Cambridge, Massachusetts, USA. ACM 978-1-59593-638-7/07/0004.

Krämer, M., Geraldy, A. (2006), “Energy Measurements for MicaZ Node”, available at: <http://vs.informatik.uni-kl.de/>, last access: 13 Dec. 2009.

Messina, A., Jones, I. A., Williams, E.J. (1996), “Damage detection and localization using natural frequency changes”, in Proceedings of Conference on Identification in Engineering Systems, Swansea, U.K., pp. 67-76.

Sohn, H., et al (2004), “A Review of Structural Health Monitoring Literature: 1996–2001”, Report Number LA-13976-MS, Los Alamos National Laboratory, Los Alamos, NM.

TinyOS: An open-source Operational System for the networked sensor regime, available at: <http://www.tinyos.net/>, last access: 13 Dec. 2009.

Titzer, B. L., Lee, D. K., Palsberg, J. (2005), “Avrora: scalable sensor network simulation with precise timing”, In: Procs of the 4th international symposium on Information processing in sensor networks (IPSN2005). Piscataway, NJ, USA.

Yick, J., Mukherjee, B., Ghosal, D. (2008), “Wireless sensor network survey”, Computer Networks, vol. 52, no. 12, pp. 2292-2330, August 2008..