Autonomous Structural Condition Monitoring based on Dynamic Code Migration and Cooperative Information Processing in Wireless Sensor Networks

K. SMARSLY, K. H. LAW and M. KÖNIG

ABSTRACT

Wireless sensor networks have emerged in recent years as a viable alternative substituting conventional, tethered systems for structural health monitoring. Because of the limited internal resources of wireless sensor nodes, new and innovative concepts are needed to implement resource-efficient operation for reliable monitoring. This study explores the use of dynamic code migration and automated, cooperative information processing as a means to achieve resource efficiency and system reliability. Autonomous software agents are installed on the sensor nodes continuously executing simple resource-efficient routines to locally process the data sets measured from the structure. Upon the detection of potential anomalies in the data sets, specialized agents are physically migrating to the relevant sensor nodes to perform more complex – and resource-consuming – algorithms in order to analyze the local sensor data. Furthermore, a central information pool, used by the software agents and continuously being updated, is introduced to assess the system behavior. This paper presents a prototype system which demonstrates the implementation of dynamic code migration and cooperative information processing for autonomous condition monitoring.

INTRODUCTION

Structural health monitoring systems aim to enhance the safety and reliability of engineering structures by detecting structural deteriorations before they reach critical states. Recent collapses of large-scale engineering structures as well as increased maintenance and repair costs underscore the need for cost-effective and reliable monitoring systems [1-3]. Integrating advanced embedded systems technologies and powerful wireless communications, wireless sensor networks have emerged as a cost-effective, decentralized approach for structural health monitoring. Compared to tethered monitoring systems that employ extensive wired infrastructure, wireless sensor networks, besides being low-cost, are easy to install and to maintain [4, 5].

Kay Smarsly, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA, USA
Kincho H. Law, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA, USA
Markus König, Department of Civil and Environmental Engineering, Ruhr-Univ. Bochum, Bochum, Germany

A wireless sensor network consists of a number of wireless sensor nodes installed in a structure that each node can automatically collect structural as well as environmental data. Embedded with engineering algorithms, each sensor node can also analyze and aggregate the collected data sets. The analysis results are then transferred to centralized computer systems for further processing. By first analyzing the data sets locally and then communicating the results to connected computer systems, transmissions of large records of raw sensor data can be avoided. As a result, energy consumption for wireless data transmission is substantially reduced. However, considerable computational power is needed for the local execution of complex engineering analyses. There have been active research efforts towards optimizing both sensor node hardware and software to reduce the energy consumption and to prolong the battery life of the sensor nodes. New software approaches, such as energy-efficient source coding and modulation as well as designing resource-efficient network protocols, and new concepts on hardware circuitry and transmitter modules for improving energy-consuming components have been proposed [6-8].

In addition to making the software and hardware more resource-efficient, a holistic (local/global) strategy is needed to assess local damages and global structural changes. Inherent to the embedded data interrogation executed on the sensor nodes is that the structural and environmental data, which is sensed and locally analyzed on a sensor node, represents an isolated, limited view on a small area of the total structure. Even though the data is usually collected at critical locations, individual sensor information does not provide a global picture of the structural condition. In addition to the detection of local damages and deteriorations (e.g. corrosion, cracks, etc.), changes in the global structural response and behavior (such as altered stiffnesses and structural stability) should also be considered. The local algorithms embedded *a priori* on the sensor node may not necessarily be the most appropriate routine to diagnose the structural conditions.

To achieve (i) a resource-efficient and (ii) a reliable wireless monitoring system holistically observing local as well as global structural phenomena, this research integrates dynamic code migration and cooperative information processing into a wireless sensor network. In this paper, the monitoring concept and the decentralized architecture of the wireless monitoring system are illustrated. The design and the implementation of the sensor nodes are briefly described. Validation tests of the prototype system are presented, followed by a discussion and a comparison of the new concept with approaches currently used in wireless monitoring systems.

PROTOTYPE DESIGN AND IMPLEMENTATION OF A COOPERATIVE MOBILE AGENT-BASED WIRELESS MONITORING SYSTEM

The concept of dynamic code migration and cooperative information processing is introduced here for the implementation of a resource-efficient and reliable wireless structural monitoring system. As depicted in Fig. 1, the monitoring system consists of a base system (or base node) and clusters of sensor network systems, each cluster being composed of one head node, that manages the cluster, and a number of sensor nodes. The base node is connected to a local computer system which includes a database and an information pool providing, e.g., information on modal properties of the structural system observed, information on sensor nodes installed, and a catalog of data analysis algorithms. For the prototype implementation of the database and the information pool, the open-source database management system MySQL is employed.

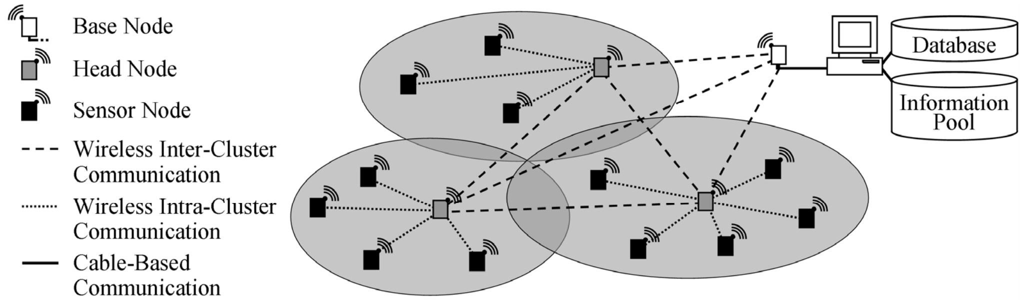

Fig. 1. Hierarchical architecture of the monitoring system.

Embedded Mobile Agents

Two basic types of mobile software agents are designed, namely the "on-board agents" permanently residing at the head and sensor nodes and the "migrating agents" located at the head nodes to be sent to the sensor nodes upon request. The on-board agents installed on the sensor nodes are self-contained, interacting software entities capable of making their own decisions and acting in the wireless sensor network with a high degree of autonomy. The on-board agents are designed to continuously take measurements from the observed structure, to perform simple routines for detecting suspected abnormalities and to aggregate the measurement data (for example, to extract representative features and values from sets of measurements). The aggregated data is then transmitted to the database installed on the connected local computer for persistent storage.

As opposed to on-board agents that are permanently residing at the nodes, the migrating agents are capable of physically migrating from one node to another in real-time. While the on-board agents at the sensor nodes are continuously executing relatively simple yet resource-efficient routines, the migrating agents are designed to carry out more comprehensive data analysis algorithms directly on a sensor node. Acting upon a request by an on-board agent in case of detected or suspected abnormal changes of the monitored structure ("anomalies"), a migrating agent is dynamically composed with the most appropriate algorithm selected from the information pool for analyzing the detected anomaly.

Implementation of Sensor and Head Nodes

For rapid prototype purpose, Oracle SunSPOT devices are deployed for the implementation of the sensor and head nodes. The devices, built upon the IEEE 802.15.4 standard for wireless networks, comprise a fully capable, CLDC 1.1-compatible Java Platform, Micro Edition (Java ME). The computational core is a 32-bit Atmel AT91RM9200 executing at 180 MHz maximum internal clock speed. The devices include a 3-axis linear accelerometer that measures a bandwidth of 4.0 kHz over a scale of ± 6 g as well as an integrated temperature sensor, a number of analog inputs, momentary switches, general purpose I/O pins and high current output pins [9, 10]. On the software side, a Squawk Java virtual machine, running without an underlying operating system, ensures a light-weight execution of multiple embedded applications on the nodes [11, 12]. For carrying out the monitoring tasks in a fully autonomous fashion, the agents are embedded into each node, following a mobile multi-agent approach as proposed by Smarsly *et al.* [13].

On the *sensor nodes*, two categories of on-board agent are prototypically implemented, administrator agents and temperature analysis agents. An administrator agent is designed to manage a sensor node. For example, it is responsible for battery level monitoring and for memory management of the sensor node. The temperature analysis agent continuously collects and aggregates the temperature data, and detects significant temperature changes in the structure that may signify abnormality. Incorporating the basic features of multi-agent technology, such as autonomy, social ability (interaction), reactivity and pro-activeness [14], the temperature analysis agent periodically communicates the aggregated data and the analysis results to the head node, which in turn forwards the data through the base node to the local computer for persistent storage and further processing.

The head nodes contain both on-board agents and migrating agents. The on-board agents at a *head node* play a very different role than those at the sensor nodes within the same cluster. Unlike on-board agents situated on sensor nodes, the on-board agents operating on head nodes are primarily administrative agents, which serve as local coordinators for their cluster without being responsible for sensing or analyzing measurement data; furthermore, they are responsible for performing cluster-internal management and administration tasks, including communication with other clusters and the base node. Migrating agents implement the different analysis algorithms to be used for diagnostic purpose. Technically, the bytecode of the migrating agents and algorithms is stored on the head nodes. Depending on the symptoms received from the sensor nodes, different types of migrating agents, comprising different analysis algorithms, are assembled. The corresponding agent objects, together with all relevant agent capabilities and algorithms, are composed and instantiated during runtime on the head node, and transmitted to the responsible sensor nodes.

For implementing the wireless agent migration, the characteristics of the Squawk Java virtual machine, as described earlier, are advantageously utilized. Squawk employs an application isolation mechanism which represents each application as an object, being completely isolated from other objects. Objects running on a node, such as migrating agents, can thus be paused, serialized and, together with agent behaviors, agent states and required algorithms, physically transferred to Squawk instances running on other nodes.

VALIDATION EXPERIMENTS

For the proof of concept of the prototype system, validation tests are conducted in the laboratory. Main objectives of the validation tests are (i) to verify the condition monitoring capabilities of the wireless monitoring system and (ii) to obtain system performance data for evaluating the resource efficiency achieved by the newly proposed concepts. As shown in Fig. 2 and 3, the prototype monitoring system is first mounted on an aluminum plate that serves as a test structure. Heat is introduced to the test structure simulating an anomaly to be detected and handled by the wireless monitoring system.

The test structure, a 900 mm × 540 mm aluminum plate (t = 0.635 mm) with one edge being clamped, is instrumented with an array of 9 precision temperature sensors, type LM335A manufactured by National Semiconductor. For this experimental test, the wireless monitoring system prototypically installed is composed of one cluster – comprising one head node and three sensor nodes – as well as one base node that connects the wireless monitoring system to the local computer. As shown in Fig. 2, the test structure is subdivided into three monitoring sections A, B, C. Each of the three sensor nodes (S_A, S_B, S_C) is responsible for monitoring one section. Each sensor node is connected to three temperature sensors that are located within its monitoring section. In addition, each sensor node includes an integrated STMicroelectronics LIS3L02AQ accelerometer. Fig. 3 shows the fully instrumented test structure.

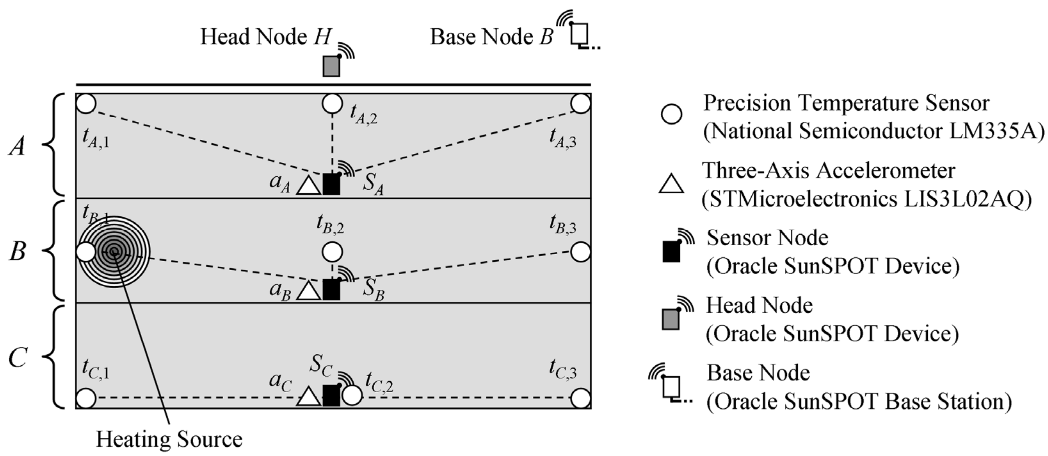

Fig. 2. Overview of the prototype wireless monitoring system.

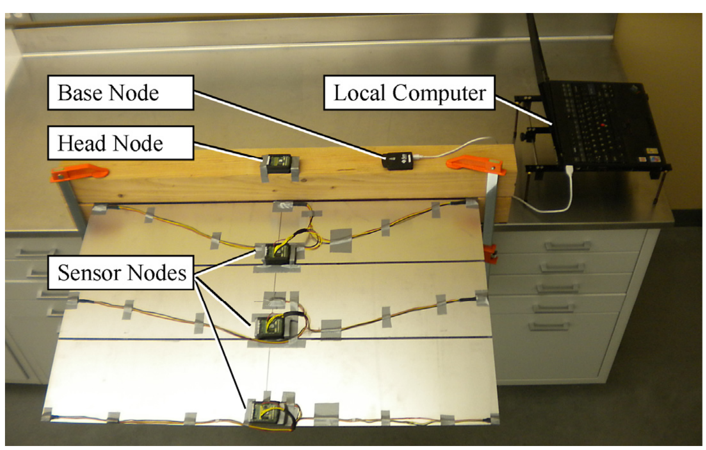

Fig. 3. Laboratory setup: Wireless monitoring system installed on an aluminum test structure.

In the laboratory tests, the on-board agents operating on the sensor nodes are continuously sensing temperature measurements using the temperature sensors attached. The collected temperature measurements are locally analyzed based on simple threshold computations, then forwarded to the local computer and stored in the database. To simulate structural changes of the test structure, heat is introduced underneath the aluminum plate near the monitoring section B below temperature sensor $t_{B,1}$. A critical plate temperature T_{crit} = 60 °C is pre-defined as a threshold value indicating that an anomaly may occur and may require attention by the monitoring system. Fig. 4 illustrates the temperature distribution calculated from the temperature measurements collected by the on-board agents at the time $t = t(T_{crit})$.

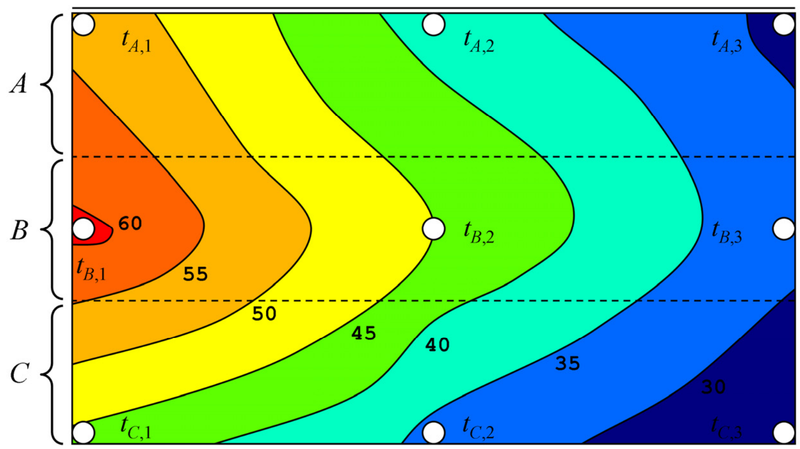

Fig. 4. Temperature distribution (°C) on the upper side of the test structure.

Upon detecting a potential anomaly, the monitoring procedure follows the steps depicted in the sequence diagram shown in Fig. 5. As can be seen from Fig. 4, T_{crit} is first exceeded in monitoring section *B*, for which the on-board agents of sensor node S_B are responsible. The on-board agents of sensor node S_B notify the head node about the observed situation. As a result, on the head node a migrating agent is immediately composed and instantiated in order to analyze the current condition of the test structure in more detail. First, the information pool installed on the local computer is queried for appropriate actions to be undertaken. In this example, the Cooley-Tukey FFT algorithm is selected to analyze the structural condition with respect to determining the modal parameters of the test structure. Furthermore, based on the information provided by the information pool, details on the migration are specified. In this case, sensor node S_C, instead of sensor node S_B where the anomaly has first been detected, is defined as the target node for the agent migration. The reason is that sensor node S_C along with its internal accelerometer is installed at the free end of the aluminum test structure and can most likely generate more sensitive results than S_B when acquiring acceleration measurements for analyzing the modal properties of the structure. Together with the selected algorithm and migrating path, modal parameters for the undamaged test structure is also passed onto the head node for composing the migrating agent.

Consequently, the migrating agent composed and initialized on the head node is equipped with a Cooley-Tukey FFT algorithm and global system information such as first modal frequency of the undamaged test structure. After having migrated to sensor node S_C, the migrating agent accesses the sensor node's internal accelerometer, senses acceleration measurements and calculates the current first modal frequency of the test structure. Fig. 6 shows the frequency response function computed by the migrating agent from the vertical accelerations at S_C. The current first modal frequency is calculated to 1.6 Hz, which does not significantly differ from the first modal frequency of the undamaged system. These results are sent by the migrating agent to the local computer, where they are stored in the form of a safety report accessible by any responsible individuals.

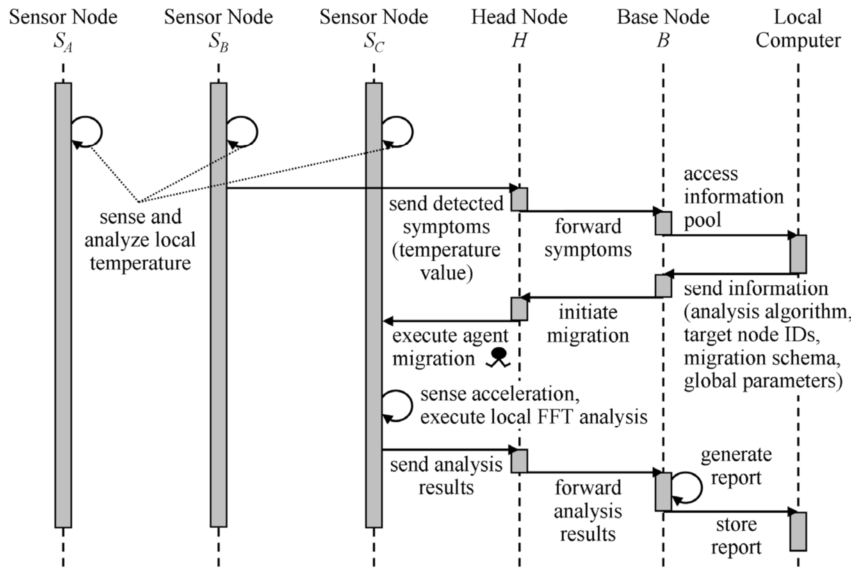

Fig. 5. Monitoring procedure automatically executed in consequence of the detected anomaly.

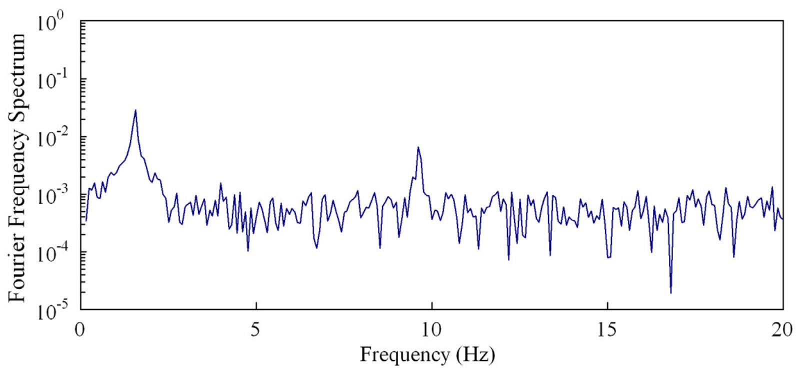

Fig. 6. Frequency response function calculated by the migrating agent.

SUMMARY AND DISCUSSION

In this paper, the prototype development of an autonomous real-time wireless monitoring strategy has been presented. Specifically, the deployment of dynamic code migration and cooperative information processing for structural health monitoring has been demonstrated. As illustrated, the prototype system implemented operates (i) resource-efficiently and (ii) reliably, considering both the local and global structural phenomena. Coupling local information, directly processed on the sensor nodes, with global information, provided by a central information pool, the prototype system is capable of detecting and handling structural changes in real-time.

As corroborated by the validation tests conducted, the resource consumption of the prototype system, compared to conventional systems, could significantly be reduced. Owing to the migration-based approach, a reduction of more than 90% of wirelessly communicated data could be achieved as compared with traditional approaches that communicate all collected raw sensor data to a central server. Also, the memory consumption, as measured in the laboratory experiments, could be reduced by approximately 70 kB per sensor node in comparison with those conventional approaches that perform embedded analyses directly on the sensor nodes without applying dynamic code migration.

To further expand the capabilities of the wireless monitoring system, future research will focus on incorporating existing engineering applications and software tools into the system, to be utilized by the on-board agents situated on the sensor nodes. Last but not least, the concept of dynamic code migration and cooperative information processing represents a flexible and modular software strategy that can be applied to a wide variety of wireless and distributed monitoring systems.

ACKNOWLEDGMENTS

This research is partially supported by the German Research Foundation (DFG) under grant SM 281/1-1, awarded to Dr. Kay Smarsly. The research is also partially funded by the US National Science Foundation (Grant No. CMMI-0824977), awarded to Professor Kincho H. Law.

REFERENCES

1. Minnesota Department of Transportation. 2007. “Interstate 35W Mississippi River Bridge Fact Sheet,” St. Paul, MN, USA: Minnesota Department of Transportation.
2. United States Department of Transportation – Federal Highway Administration. 2008. “2008 Status of the Nation’s Highways, Bridges, and Transit: Conditions and Performance,” [online] Washington, DC, USA: FHA. Available at: <http://www.fhwa.dot.gov> [Acc. March 9, 2011].
3. Wardhana, K., and Hadipriono, F. C. 2003. “Analysis of Recent Bridge Failures in the United States,” *Journal of Performance of Constructed Facilities*, 17(3), pp. 144-150.
4. Lynch, J. P., and Loh, K. J. 2006. “A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring,” *Shock and Vibration Digest*, 38(2), pp. 91-128.
5. Lynch, J. P., Sundararajan, A., Law, K. H., Kiremidjian, A., and Carryer, E. 2004. “Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency,” *Smart Materials and Structures*, 13 (4), pp. 800-810.
6. Tang, Q., Yang, L., Giannakis, G. B., and Qin, T. 2007. “Battery Power Efficiency of PPM and FSK in Wireless Sensor Networks,” *IEEE Trans. on Wireless Comm.*, 6(4), pp. 1308-1318.
7. Prakash, Y., and Gupta, S. K. S. 2003. “Energy Efficient Source Coding and Modulation for Wireless Applications,” *Proceedings of the Wireless Communications and Networking Conference*, New Orleans, LA, USA.
8. Laneman, J. N., Tse, D. N. C, and Wornell, G. W. 2004. “Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior,” *IEEE Transactions on Information Theory*, 50(12), 3062-3080.
9. Oracle Corporation, 2010. “*SunSPOT Programmer's Manual – Release v6.0*,” Redwood Shores, CA, USA: Oracle Corporation.
10. Oracle Corporation, 2010. “*Main Board Technical Datasheet Rev 8.0*,” Redwood Shores, CA, USA: Oracle Corporation.
11. Smith, R. B., Cifuentes, C., and Simon, D. 2005. “Enabling Java TM for small wireless devices with Squawk and SpotWorld,” *Proceedings of the Second Workshop on Building Software for Pervasive Computing*, San Diego, CA, USA.
12. Simon, D., Cifuentes, C., Cleal, D., Daniels, J., and White, D. 2006. “Java on the Bare Metal of Wireless Sensor Devices: The Squawk Java Virtual Machine,” *Proceedings of the Second International Conference on Virtual Execution Environments*, Ottawa, Canada.
13. Smarsly, K., Law, K. H., and König, M. 2011. “Resource-Efficient Wireless Monitoring based on Mobile Agent Migration,” *Proc. of the SPIE Smart Structures/NDE Conf.*, San Diego, CA, USA.
14. Wooldridg, M., and Jennings, N. R. 1995. “Intelligent Agents: Theory and Practice,” *The Knowledge Engineering Review*, 10(2), pp. 115-152.