Structural health monitoring of wind turbines observed by autonomous software components – 2nd level monitoring

K. Smarsly & K. H. Law
Stanford University, CA, USA

D. Hartmann
Ruhr-University Bochum, Germany

Abstract

Wind energy, with an annual growth of about 30%, represents one of the fastest growing renewable energy sources. Continuous long-term monitoring of wind turbines can greatly reduce maintenance and repair costs, ensure structural safety, extend operational life, and, ultimately, improve the profitability of wind turbines. A decentralized wind turbine monitoring system has been developed and installed on a 500 kW wind turbine in Germany. During its operation, temporary malfunctions of the installed sensing units have been observed. If not detected in a timely manner, these malfunctions can lead to the loss of important monitoring data. To monitor the installed sensors, autonomous software based on multi-agent technology has been developed and modularly integrated into the monitoring system. Since the initial deployment of the autonomous software in 2009, various sensor malfunctions have autonomously been detected, and email notifications were automatically composed and immediately issued to the responsible personnel by the autonomous software. As a result, appropriate actions, such as replacing or remotely restarting malfunctioning sensing units, were undertaken in a timely fashion by the responsible engineers.

Keywords: Structural Health Monitoring, Fault Detection, Autonomous Software, Multi-Agent Technology, 2nd Level Monitoring, Wind Turbines

1 Introduction

Wind energy, considered to be safe, inexpensive and clean, is one of the fastest growing renewable energy resources (Bloomberg, 2011). According to the World Wind Energy Association, wind energy systems are currently used for power generation in 83 countries, 52 of which having increased their totally installed wind energy generation capacity in 2010 (WWEA, 2011). In total, the worldwide capacity has reached the potential of 196,630 MW in 2010, and is estimated to exceed 600,000 MW in 2015, and 1,500,000 MW in 2020.

When installing wind turbines, key considerations are their availability, reliability, and profitability. In this context, the necessity of deploying structural health monitoring (SHM) systems on wind turbines has been recognized (BMJ, 2009; MIWF, 2009). Main motivations for installing SHM systems are to ensure the long-term structural integrity and the operational functionality of wind turbines; both can be facilitated by systematically monitoring the condition of structural members and/or mechanical components with respect to potential damages and deteriorations.

In an ongoing research project funded by the German Research Foundation (DFG) at the Ruhr-University Bochum (Hartmann and Höffer, 2010), a decentralized real-time SHM system has been designed, implemented, and installed on a 500 kW wind turbine in Germany. The SHM system couples different information and communication technologies and provides automated data acquisition, data storage and data interrogation functionalities. In addition, it enables remote access to the monitoring data taken from the wind turbine including flexible data visualization, data analysis, and data export mechanisms.

During the operation of the system, it has been observed that temporary malfunctions occurred, affecting the performance and the reliability of the system. Malfunctions of a SHM system or breakdowns of sensor components can lead to undesirable interruptions of the data acquisition, degrading data quality considerably (Besnard, 2009; Hameed *et al.*, 2009; Nilsson and Bertling, 2007). The causes for such malfunctions are manifold, e.g. communication problems when using long-distance lines, temporary power outages that affect the computer systems or simply hardware problems of the data acquisition units due to extreme weather conditions. Evidently, if not detected timely the malfunctions might cause the loss of valuable monitoring data.

To overcome such drawbacks, autonomous software is implemented and integrated into the existing SHM system. Based on multi-agent technology, the autonomous software is designed to automatically detect malfunctions of the entire SHM system and its components and to immediately send out alerts for remediation ("2nd level monitoring" or "monitoring the monitoring"). This paper first presents the decentralized real-time SHM system installed on the wind turbine. The design and the implementation of the agent-based autonomous software and its integration into the existing SHM system are described. Finally, the operation of the autonomous software is presented to demonstrate the practicability and the reliability of the SHM system.

2 A decentralized wind turbine monitoring system

For the continuous monitoring of a 500 kW wind turbine, an ENERCON E-40 located in Dortmund (Germany), a decentralized real-time SHM system is designed and prototypically implemented. The SHM system consists of two subsystems, (i) an on-site hardware system and (ii) a decentralized remote software system (Figure 1). Installed in the observed wind turbine, the on-site hardware system includes sensors, data acquisition units (DAUs), and a local computer (on-site server). Remotely connected to the hardware system, the software system is composed of software modules that are designed to continuously execute relevant monitoring tasks, such as storing and converting the monitoring data collected from the wind turbine. Both subsystems are briefly described in the following subsections.

2.1 On-site hardware system

The on-site hardware system is implemented to collect structural and environmental data for assessing the performance and the condition of the wind turbine tower. For that purpose, the wind turbine is instrumented with sensors that are installed both inside and outside the tower as well as on the foundation of the wind turbine. The sensors are controlled by the data acquisition units which are connected to the on-site server located in the maintenance room of the wind turbine.

In detail, six inductive displacement transducers, type HBM-W5K, are mounted at two different levels inside the tower. The displacement transducers are complemented by Pt100 resistance temperature detectors to capture temperature influences on the displacement measurements. Additional temperature sensors are placed at two other levels inside and outside the tower to measure temperature gradients. For the temperature data acquisition, three 4-channel Picotech RTD input modules PT-104 are installed and, through RS232 to USB interface converters, connected to the on-site server. In addition, six three-dimensional PCB-3713D1FD3G piezoelectric accelerometers, manufactured by PCB Piezotronics, are placed at five levels in the tower. At the foundation of the wind turbine, three single-axis PCB-393B12 piezoelectric seismic accelerometers are installed. For acquiring wind information, a Metek USA-1 ultrasonic anemometer is mounted on a telescopic mast next to the wind turbine.

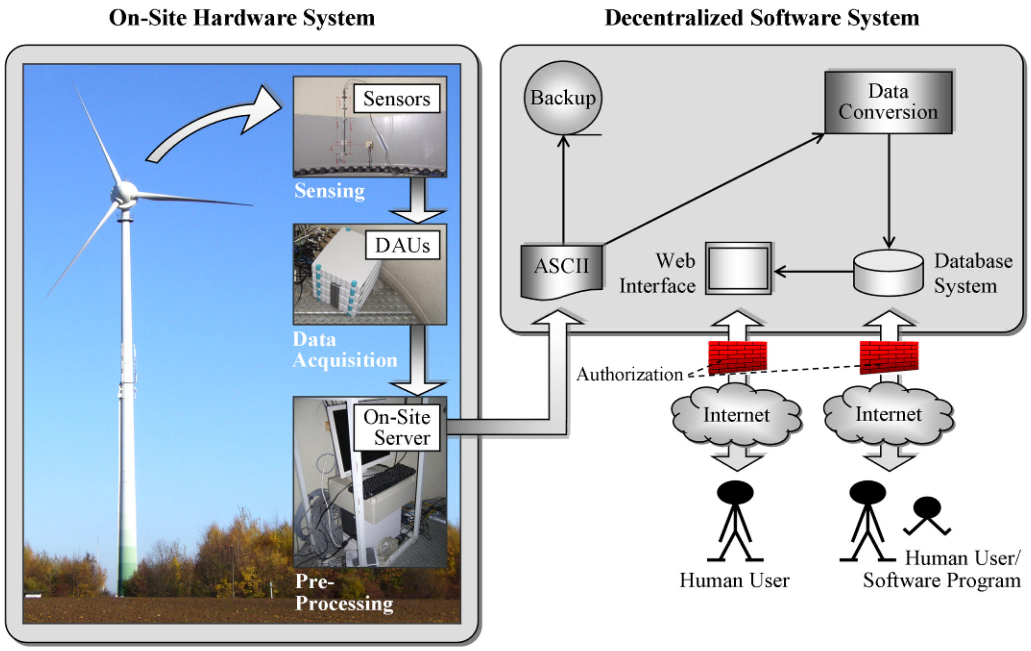
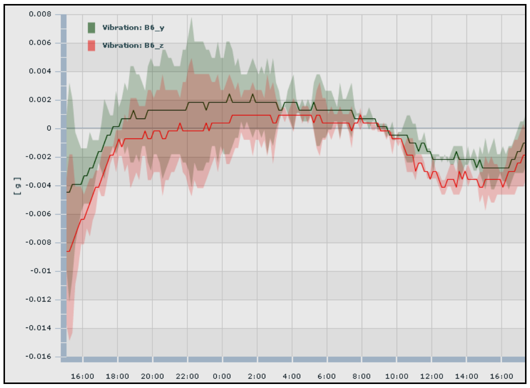


Figure 1. Architecture of the SHM system.

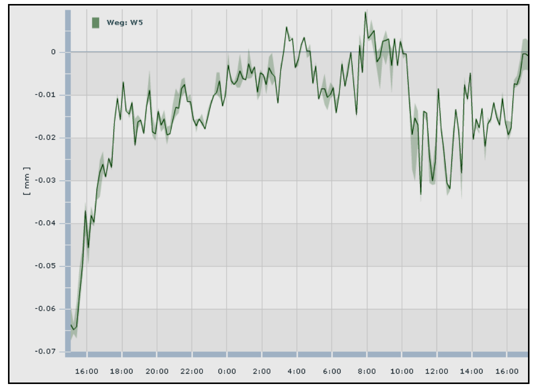
2.2 *Decentralized remote software system*

The data collected and pre-processed by the on-site hardware system is forwarded to the decentralized remote software system using a permanently installed DSL connection. The purpose of the software system is to automatically perform data archival and data processing tasks such as condensing, converting and storing the acquired data sets. The software system allows authorized users to remotely access the SHM system independent of their place of work. It comprises (a) server systems for on-line data synchronization, data conversion and data transmission, (b) RAID-based storage systems for periodic backups, and (c) a MySQL database for persistent data storage.

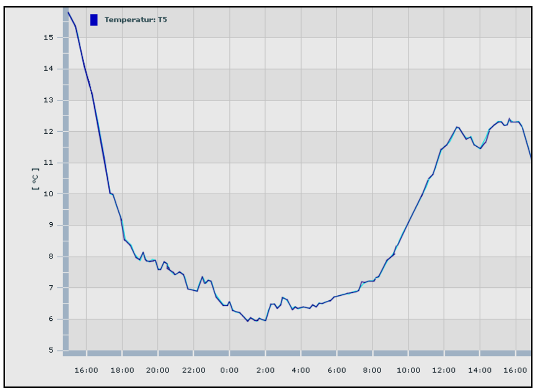
Remote access to the data is provided in two ways: through a *web interface* and through a *direct database connection*. The web interface offers GUIs for remotely visualizing, exporting and analyzing the monitoring data, while the database connection allows accessing the data directly, for example by software programs and other tools. To illustrate the remote access to the SHM system using the web interface, Figure 2 exemplarily shows parts of the GUIs visualizing monitoring data collected during 24 hours on October 22, 2010. In Figure 2a, a typical acceleration time history, recorded from the wind turbine tower at the 21 m level, is displayed. Figure 2b-d, respectively, show a displacement time history, temperature data, and wind speed measurements recorded during the same 24-hour period. In addition to visualizing the monitoring data, functionalities to remotely analyze the data sets are provided by the web interface. The web interface and the direct database connection provide the basis for further monitoring tasks such as damage detection, localization, and lifetime estimation of the wind turbine.



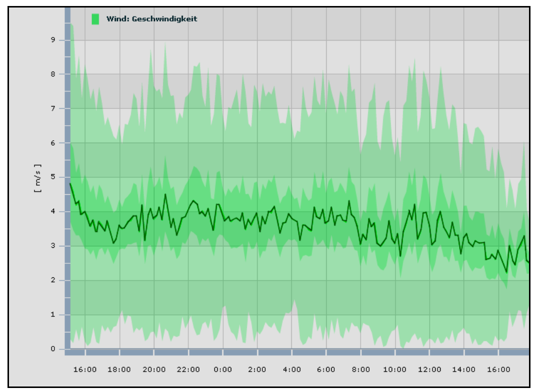
(a) Horizontal acceleration of the tower at 21 m.



(b) Displacement of the tower at 21 m.



(c) Temperature inside the tower at 21 m.

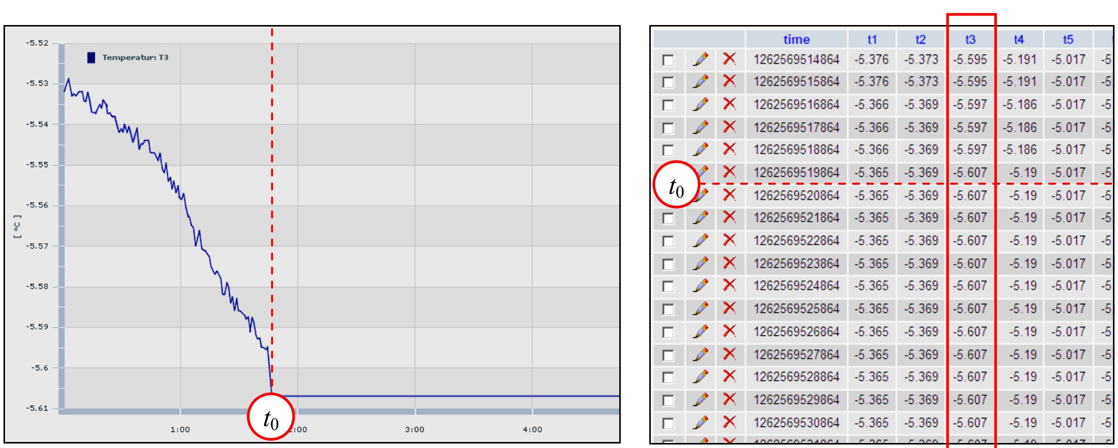


(d) Wind speed.

Figure 2. Data sets remotely accessed via the web interface provided by the SHM system (collected during 24 hours on October 22, 2010).

2.3 *Malfunctions recognized during operation*

Figure 3 elucidates a typical malfunction of one of the temperature DAUs observed during operation of the SHM system. As can be seen from Figure 3, the malfunction is implicitly indicated by anomalies within the data sets; the malfunction is characterized by a large number of equal measurements due to an internal system crash of the DAU at the time t_0. As a result, the DAU is unable to record and store the actual temperature measurements. Instead, the last normally collected value (sensed before the occurrence of the malfunction) is stored repeatedly by the DAU.



time	t1	t2	t3	t4	t5
1262569514864	-5.376	-5.373	-5.595	-5.191	-5.017
1262569515864	-5.376	-5.373	-5.595	-5.191	-5.017
1262569516864	-5.366	-5.369	-5.597	-5.186	-5.017
1262569517864	-5.366	-5.369	-5.597	-5.186	-5.017
1262569518864	-5.366	-5.369	-5.597	-5.186	-5.017
1262569519864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569520864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569521864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569522864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569523864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569524864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569525864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569526864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569527864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569528864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569529864	-5.365	-5.369	-5.607	-5.19	-5.017
1262569530864	-5.365	-5.369	-5.607	-5.19	-5.017

Figure 3. Visual and numerical representation of a DAU malfunction.

3 Autonomous software for malfunction detection – 2nd level monitoring

The purpose of the autonomous software, which is integrated into the SHM system in the form of a multi-agent system, is to automatically detect malfunctions in the DAUs. Furthermore, it informs the responsible individuals immediately about the defects so that the affected DAUs can be replaced or remotely restarted in a timely manner. The agent-based paradigm has been chosen as a technological basis for the implementation of the self-detection and self-management functions because software agents are capable of flexibly carrying out autonomous actions in cooperation with each other. According to Wooldridge and Jennings (1995), software agents are characterized by the following properties:

- Autonomy: software agents act without direct intervention of humans or other software systems, and have control over their internal states; unlike objects, software agents decide for themselves whether or not to perform an action upon request from another agent or human user
- Social ability: to achieve their goals, software agents are capable of interacting with other agents (and possibly with humans) via cooperation, coordination, and negotiation
- Reactivity: software agents perceive their environment (e.g. a SHM system) and respond timely to events and changes that occur in it
- Pro-activeness: software agents do not simply respond to external stimuli, but take the initiative and apply self-contained, goal-directed behavior

The multi-agent system, in detail, includes a number of collaborating software agents implemented to accomplish the tasks required for malfunction detection. (It should be noted that software agents can also be employed to autonomously execute other monitoring tasks such as damage detection of the wind turbine, which is, however, not addressed in this paper.) First, the multi-agent system hosts two special agents, namely the "Agent Management System (AMS) agent" and the "Directory Facilitator (DF) agent". Whereas the AMS agent is responsible for the overall management actions, the DF agent implements a sort of yellow page service, allowing an agent to advertise its services and, vice versa, to look up services provided by other agents needed to cooperatively accomplish a specific monitoring task. The AMS and DF agents together provide the overall configuration description of the multi-agent system. Furthermore, specific agents are included in the multi-agent system: an "interrogator agent" implemented for analyzing the monitoring data with respect to malfunctions, and a "mail agent" for sending notifications to the responsible individuals.

3.1 Data interrogator agent

The data interrogator agent analyzes the monitoring data collected by the DAUs to detect malfunctions. As described in the example in Figure 3, a typical malfunction of a temperature DAU is characterized by a large number of equal measurements. To detect such an anomaly, the interrogator agent at certain time intervals extracts and analyzes the temperature data sets stored in the centralized MySQL database (Smarsly *et al.*, 2010).

The connection to the database as part of the software system is established through a programming interface based on the Java Database Connectivity (JDBC). The JDBC is an industry standard for database-independent connectivity between Java applications, such as the software agents, and various database systems. The security of database requests and data transmissions is provided by the database system, i.e. the MySQL database, which requires password and username as well as secure drivers that the agent, like a human user, needs to specify.

3.2 Mail agent

Once a malfunction is detected, the engineers in charge are immediately notified. On behalf of the interrogator agent, notifications are issued by the mail agent via an agent-based email messaging service in a two-step process. In the first step, the mail agent collects all relevant data from the monitoring system that is necessary for composing the emails, i.e. the mail agent creates the emails using the information about the observed anomaly that is provided by the interrogator agent. In addition, metadata, such as addresses of the recipients and the email server to be used, is acquired for the automated emailing. This metadata is stored in configuration files located in the multi-agent system. In other words, the email content is assembled based on the anomaly information received from the interrogator agent and the email header is automatically created based on the metadata obtained by the mail agent from the configuration files resided in the multi-agent system. As soon as the emails are composed, they are sent by the mail agent to the email clients (the individuals responsible for the malfunctioning devices). Secure email messages are ensured by username- and password-based authentications to be specified by the mail agent when trying to access the mail server.

3.3 Extensibility and decentralization

The multi-agent system can easily be extended to accommodate additional functions. For example, self-detections of other types of malfunctions (that are not characterized by a large number of identical values) or interrogations of further monitoring data (such as acceleration data) can be implemented as additional specialized agents that are integrated into the multi-agent system. Also, damage detection algorithms, e.g. using fast Fourier transforms or autoregressive models, can be included in the multi-agent system as software agents. Being "wrapped" by an agent, software packages, such as MATLAB, can further be integrated into the multi-agent system.

The multi-agent system described herein is implemented in compliance with the specifications issued by the Foundation for Intelligent Physical Agents (FIPA) to ensure extensibility and interoperability. FIPA, the IEEE Computer Society standards organization for agents and multi-agent systems, promotes agent-based technology, interoperability of agent applications, and interoperability with other technologies. The FIPA agent communication specifications (FIPA, 2002), for example, ensure a common understanding of two or more communicating agents based on the FIPA Agent Communication Language (ACL) as used for the communication within the multi-agent system.

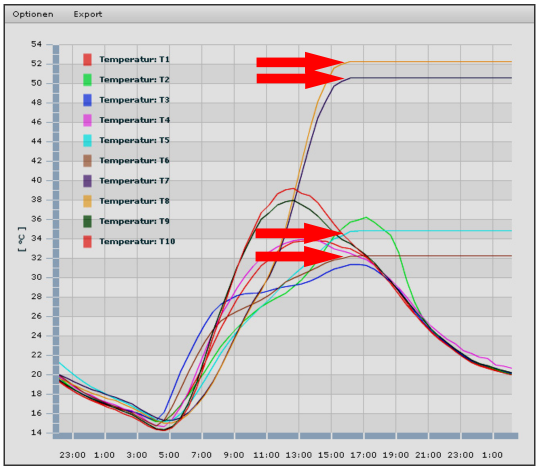
4 Validation of the agent-based approach

The multi-agent system is integrated into the existing SHM system to autonomously monitor the temperature DAUs instrumented on the wind turbine. The SHM system and the multi-agent system are installed at different geographical locations as follows:

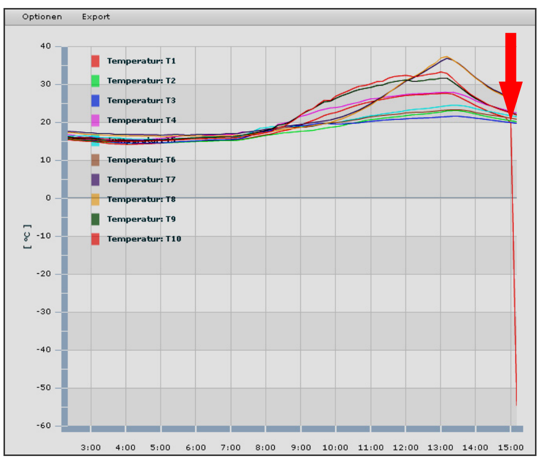
- The on-site hardware system (including sensors, DAUs and on-site computer) is installed at the wind turbine located in Dortmund (Germany).
- The components of the decentralized software system including the MySQL database system, reside on different computers at the Institute for Computational Engineering at the Ruhr-University Bochum (Germany).
- For the validation tests, the multi-agent system is distributively located at different institutes: the interrogator agent and the mail agent for detecting malfunctions and generating alerts are located at Stanford University (USA) while the email server, used by the mail agent at Stanford University, is situated at the Ruhr-University Bochum[1].

[1] For long-term operation, the multi-agent system has been installed at the Institute for Computational Engineering in Bochum.

For malfunction detection of the temperature DAUs, the interrogator agent is adjusted such that, based on experience, 150 consecutive equal values signal a malfunction. Figure 4 shows two malfunctions detected from the data sets by the data interrogator agent in July, 2010, (Figure 4a) and in November, 2011, (Figure 4b). Corresponding to the agent interaction described above, the mail agent generates an email alert to notify the authorized engineers immediately about the malfunction so that appropriate measures can be taken in order to avoid the loss of valuable monitoring data. Figure 4c reflects the detailed email information on the detected malfunction generated and sent by the multi-agent system.



(a) Malfunction occured on July 19, 2010 (affected sensors: T5, T6, T7, T8).



(b) Malfunction occured on November 5, 2011 (affected sensor: T1).

```
From:     monitoring@inf.bi.ruhr-uni-bochum.de
Subject: [Generated Email] Wind Turbine Safety Report
To:       smarsly@stanford.edu, [redacted]
--------------------------------------------------------------------
This email has autonomously been generated
and was sent by a software agent. Do not reply.
--------------------------------------------------------------------

(...)

Analysis properties:
Database driver:    com.mysql.jdbc.Driver
Database table:     pt104
Number of values observed per sensor: 86400
Number of significant values: 150
Sensors analyzed: t1;t2;t3;t4;t5;t6;t7;t8;t9;t10;

Anomaly detected:
DAU id:             pt104
Date:               Sat Nov 05 15:05:09 CET 2011
Sensor:             t1
Errornumber:        -1.0E30

--------------------------------------------------------------------
Properties:
IP address:         134.147.216.69
Internet host:      tunsql
Hap:                tunsql:1099/JADE
Responsible agent: mailAgent@tunsql:1099/JADE
Agent container:    tunsql:1099/JADE
Container state:    Ready(4)
Agent platform:     tunsql:1099/JADE
Mail host:          delta.inf.bi.rub.de
Port:               25
Debugging:          true
Protocol:           smtp
Sender:             monitoring@inf.bi.ruhr-uni-bochum.de
Receivers:          2
Local config file:  config\mail.shm
```

(c) Excerpt of an email notification sent in consequence of the malfunction shown in Fig. 4b.

Figure 4. Visual representation of two detected malfunctions and email notification generated and sent by the multi-agent system.

5 Conclusions

The implementation and the deployment of autonomous software for malfunction detection of a SHM system have been presented. The autonomous software, based on multi-agent technology, has been integrated into an existing decentralized SHM system, which is deployed for continuous real-time monitoring of a 500 kW wind turbine. Since its initial deployment in December, 2009, the agent-based autonomous software is in continuous operation and has reliably detected all malfunctions occurred. Once having detected a malfunction of a sensing unit, the autonomous software has immediately notified the responsible engineers by email, whereupon the affected sensing units were remotely restarted in a timely manner. The ability to automate the (2nd level) monitoring of the SHM system and its components is important to ensure the reliability and integrity of the overall system.

This work has demonstrated that multi-agent technology can be flexibly deployed for long-term monitoring practice. Software agents, because of their modularity and extensibility, can be installed on any computer external to the SHM system. Representing another distinct advantage, various algorithms for malfunction detection and for further monitoring functions (e.g. damage detection of the wind turbine) can easily be embedded into the software agents and added into the SHM system. In short, the multi-agent approach has proven to be a promising paradigm for the implementation of autonomous software in structural health monitoring.

Acknowledgements

The authors gratefully acknowledge the financial support of the German Research Foundation (DFG) through the grants SM 281/1-1 awarded to Dr. Kay Smarsly and HA 1463/20-1 awarded to Professor Dietrich Hartmann. The authors also acknowledge partial support by the U.S. National Science Foundation (NSF) through grant CMMI-0824977 awarded to Professor Kincho H. Law.

References

BESNARD, F. (2009). On Optimal Maintenance Management for Wind Power Systems. *Licentiate Thesis*. KTH Royal Institute of Technology, Stockholm, Sweden.

BLOOMBERG, L.P. (2011). Global Trends in Clean Energy Investment: Clean Energy Investment, Storms to New Record in 2010. *Fact Sheet*. New York City, NY, USA: Bloomberg L.P.

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS – FIPA (2002*). SC00061:2002 FIPA ACL Message Structure Specification*. Alameda, CA, USA.

GERMAN FEDERAL MINISTRY OF JUSTICE – BMJ (2009). *EEG – Gesetz für den Vorrang erneuerbarer Energien.* (BGBl. I S. 2074 and BGBl. I S. 643), Berlin, Germany.

HAMEED, Z., HONG, Y.S., CHO, Y.M., AHN, S.H. and SONG, C.K. (2009). Condition monitoring and fault detection of wind turbines and related algorithms: a review. *Renewable and Sustainable Energy Reviews*, 13(1), pp.1-39.

HARTMANN, D. and HÖFFER, R. (2010). Lifespan assessment of wind energy converters through system identification (Lebensdauerabschätzung von Windenergieanlagen mit fortlaufend durch Systemidentifikation aktualisierten numerischen Modellen). *Research project funded by the German Research Foundation (DFG) through the research grant HA 1463/20-1*. Ruhr-University Bochum, Bochum, Germany.

MINISTRY OF INNOVATION, SCIENCE, RESEARCH AND TECHNOLOGY OF THE STATE OF NORTH RHINE-WESTPHALIA – MIWF (2009). Das Cluster EnergieForschung.NRW. *Report*. Düsseldorf, Germany.

NILSSON, J. and BERTLING, L. (2007). Maintenance Management of Wind Power Systems Using Condition Monitoring Systems – Life Cycle Cost Analysis for Two Case Studies. *IEEE Trans. on Energy Conversion*, 22(1), pp.223-229.

SMARSLY, K., LAW, K.H. and HARTMANN, D. (2010). A Multi-Agent-Based Collaborative Framework for a Self-Managing Structural Health Monitoring System. *ASCE Journal of Computing in Civil Engineering*, 10.1061/(ASCE)CP.1943-5487.0000107 (May 20, 2011).

WOOLDRIDGE, M., and JENNINGS, N.R. (1995). Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 10(2), pp. 115-152.

WWEA (World Wind Energy Association), (2011). World Wind Energy Report 2010. *Report*. Bonn, Germany: WWEA. [Online]. Available at <http://www.wwindea.org/home/images/stories/pdfs/worldwindenergyreport2010_s.pdf> [Accessed November 18, 2011].