

Sensor technologies for civil infrastructures, Volume 2: Applications in structural health monitoring

Edited by Ming Wang, Jerome Lynch and Hoon Sohn

To be published by Woodhead Publishing Limited in 2014

Chapter 18 Sensor Data Management Technologies

Kincho H. Law, Stanford University, USA

Kay Smarsly, Stanford University, USA

Yang Wang, Georgia Institute of Technology, USA

Chapter Outline

Section 1 Introduction

Section 2 Sensor Level Data Processing and Management

Section 3 In-Network Data Communication and Management

Section 4 Persistent Data Management and Retrieval

Section 5 Summary and Discussion

Section 1 Introduction

Structural health monitoring (SHM) systems are being deployed to collect measurements of structural responses originated from ambient and/or external disturbances, and to draw conclusions about the state of health of a structure based on the measurement data. Typically, sensors are strategically placed in a structure to measure and record environmental and response data. The collected data, either in its raw form or being filtered or pre-processed, is then transmitted to another sensor node or to a data server for further processing and archival. Data management thus plays a very critical role for successful deployment of a SHM system.

An overview of data management issues has been discussed previously by McNeill (2009). As noted, data management issues include data collection and management at the site, data communication and transfer to off-site facilities, and data storage. Depending on the type of structure, the extent of information to be measured, and the purpose for monitoring, different system configurations, measurement equipment, data communication and storage apparatus may be used. To provide a comprehensive treatment of all the issues related to data processing, data communication and data management of SHM systems is a non-trivial endeavor. In this chapter, the discussion will focus on selected data issues for the deployment of wireless sensors and sensor networks, and persistent backend data storage, management and access.

Sensors are the most basic and primitive entities of a SHM system. They are instrumented in a structure and its vicinity to measure response data (e.g. strain, acceleration and displacement) as well as environmental data, such as temperature, wind speed, and wind direction. The past couple of decades have seen tremendous development of wireless sensors and communication technology. With embedded microcontrollers, wireless sensors and wireless sensor networks have opened many new and exciting

opportunities for their deployment in SHM systems. Wireless sensor technology not only eradicates cables and the associated material and labor cost, but also allows flexible system configurations. As many wireless sensors are designed to be powered by batteries, special considerations should be paid to minimize energy usages as much as possible. Furthermore, wireless sensors and wireless systems have their limitations, among many other issues, on communication range and possible data loss. Communication protocols must be designed to ensure that data can be transmitted in the monitoring environment and that data is being transmitted and received properly.

Measurement data is a valuable asset not only for structural health monitoring, but also for life-cycle assessment and management of the structure and the system as a whole. Persistent data storage that can effectively support data access is an important consideration of the overall data management system design. With the proliferation of the Internet and ubiquitous computing, advanced software technologies can be deployed to facilitate data access, to support maintenance and operation of the monitored structure, and to self-monitor the SHM system itself. The intriguing integration of data management and system monitoring is a subject that is worth consideration as a system solution to the monitoring of a civil infrastructure system.

This chapter is organized as follows. Section 2 discusses the data processing and management issues at the wireless sensor node, with special attention on reducing energy consumption inherent to wireless data communication. Section 3 discusses the data issues related to in-network communication. Specifically, the section examines the issues related to wireless communication range in a monitoring environment and robust communication protocol design. Furthermore, a dynamic code migration paradigm, taking advantage of wireless in-network communication, is introduced as a flexible approach towards data processing. Section 4 discusses persistent data management and retrieval, and presents software modules

developed to facilitate data access and to support system monitoring of a SHM system. Section 5 concludes the chapter with a brief summary and discussion.

Section 2 Sensor Level Data Processing and Management

When using wireless sensors for acquiring and transmitting measurements, issues that worth considering are the constraints regarding battery life and capacity, as well as communication range. These constraints could affect how data is being processed and managed at the sensor nodes. Generally speaking, wireless sensors consume less battery power to perform onboard data processing, thereby reducing the amount of data to be transmitted, than to transmit lengthy raw time-histories of sensor data. In other words, in terms of saving battery power, onboard processing is “cheaper” than wireless communication. This is particularly relevant for applications in civil structures, which typically involve long-range communication requiring signal boosting and/or multi-hopping and consume significant power by the wireless transceiver. Onboard processing measures can be adopted to preserve the life of portable batteries coupled with the wireless sensing node. These measures include embedding engineering analyses and performing data compression locally on the sensor node prior to data transmission. Using a wireless sensing node designed by Lynch (Lynch and Law 2002) as an example, this section briefly reviews a power efficiency study illustrating the demands for local data processing and data communication.

2.1 Power-efficiency of an example wireless sensing node

As an early research prototype effort, the wireless sensing node designed by Lynch (Lynch and Law 2002) features two onboard microcontrollers, a low-power 8-bit Atmel AVR AT90S8515 model and a high-performance 32-bit Motorola MPC555 PowerPC model. Normally on for maintaining the overall operation of the wireless sensing node, the AVR microcontroller draws 8 mA of current when powered by a 5 V source. On the other hand, the MPC555 microcontroller, which has a floating point unit and ample internal program memory, draws 110 mA of current when powered at 3.3 V. The MPC555 microcontroller is normally off to save power. When needed, the AVR microcontroller wakes up the MPC555 to perform computationally demanding task. For wireless communication, a Proxim

RangeLAN2 7911 wireless modem, operating on the 2.4 GHz FCC unlicensed band, is incorporated. Using a 1 dBi omni-directional antenna, open space ranges of 1,000 ft can be attained. To attain such large communication range, the wireless radio consumes a significant amount of power. When internally powered by 5 V, the wireless modem draws 190 mA of current during data transmission and reception, and draws 60 mA of current when idling.

To quantify power saving, the energy consumed by the wireless sensing node is experimentally measured using 7.5 V battery sources. Table 1 summarizes the expected operational life of a battery pack with Energizer AA L91 lithium-based battery cells when continuously drained (Lynch *et al* 2004). As shown in Table 1, the wireless modem consumes a relatively large amount of battery energy. To preserve battery life, the use of the modem should be minimized by limiting the amount of data wirelessly transmitted. When executing an embedded analysis task, the MPC555 consumes 363 mW by drawing 110 mA at 3.3 V, while for data transmission, the RangeLAN2 radio consumes 950 mW of power running 190 mA at 5 V. Clearly, the MPC555 is 2.6 times more power efficient than the wireless radio. To determine the total amount of energy saved, the time required to perform an embedded computational task needs to be calculated. The time for data transmission can be computed based on the wireless radio baud rate (19,200 bit per second).

Table 1. Duration of battery sources for various operational states.

Operational State	Circuit [mA]	Internal [V]	Energizer L91 [hours]
AVR On/MPC555 Off	8	5	500
AVR On/MPC555 On	160	5/3.3	15
RangeLAN Active	190	5	13
RangeLAN Sleep	60	5	40

2.2 Power saving measure I - embedded engineering analyses

With the MPC555 computational core, engineering analyses can be encoded and embedded in the wireless sensing node (Lynch *et al* 2004). To assess the energy saved by the sensing node by locally processing data, Fast Fourier transform (FFT) and auto-regressive (AR) time series analysis, which are two algorithms commonly used for system identification and damage detection in structural health monitoring, are used as illustrative examples. The first algorithm, FFT, can transform time series data into frequency domain for determining structural modal properties. In this experimental study, the Cooley-Tukey version of the FFT is embedded in the wireless sensing node to locally calculate the Fourier coefficients (Press 1995). The second algorithm, which is based on AR analysis and has been widely used in damage detection studies (Sohn and Farrar 2001), fits discrete measurement data to a set of linear AR coefficients weighing past time-history observations:

$$y_k = \sum_{i=1}^p b_i y_{k-i} + r_k \quad (1)$$

Here y_k denotes the response of the structure at the k -th discrete point, which is expressed as a function of p previous observations of the system response, plus a residual error term, r_k . Weights on the previous observations of y_{k-i} are denoted by the b_i coefficients. Burg's approach to solving the Yule-Walker equations can be used for calculating the weighting coefficients (Press 1995).

Table 2 presents the time associated with each analysis and the energy saved. The time series data is obtained during a field test conducted at the Alamosa Canyon Bridge in New Mexico (Lynch, *et al.* 2003). As compared to the transmission of time-history record, computational efficiency of the embedded FFT and transmission of (a few) modal frequencies can achieve major energy savings of over 98%. Calculation of AR coefficients is more complex and requires external memory for temporary data storage, resulting in longer execution times. Nevertheless, energy savings of over 50% can be achieved. Clearly, end-users of wireless

sensing nodes should be cognizant of the execution times of their analyses, but on average, significant energy can be saved by local data interrogation on the sensor nodes.

Table 2. Energy analysis of data interrogation versus transmission of time series record.

Analysis	Length of Record N	Time of MPC555 Calculation [sec]	Energy Consumed MPC555 [J]	Time for Wireless Transmission [sec]	Energy Consumed Radio [J]	Energy Saved [%]
FFT	1024	0.0418	0.0152	1.7067	1.6213	99.1
FFT	2048	0.0903	0.0328	3.4133	3.2426	99.0
FFT	4096	0.1935	0.0702	6.8267	6.4854	98.9
AR (10 Coef.)	2000	1.3859	0.5031	3.3333	3.1666	84.1
AR (20 Coef.)	2000	2.8164	1.0224	3.3333	3.1666	67.7
AR (30 Coef.)	2000	4.2420	1.5398	3.3333	3.1666	51.4
AR (10 Coef.)	4000	2.7746	1.0072	6.6667	6.3333	84.1
AR (20 Coef.)	4000	5.6431	2.0484	6.6667	6.3333	67.7
AR (30 Coef.)	4000	8.5068	3.0879	6.6667	6.3333	51.2

2.3 Power saving measure II - data compression

Time series data can be compressed by exploiting natural internal structures of data prior to transmission, so that wireless transmission can be reduced and power can be saved. Compression algorithms can be broadly categorized into two classes: lossless and lossy compression. Lossless compression guarantees the integrity of the data without distortion. In contrast, lossy compression reduces data with reasonable distortions but can achieve higher compression rates. There are many lossless and lossy compression techniques (Salomon and Motta 2010). A simple and computationally inexpensive compression technique, known as Huffman coding, is selected for illustration (Sayood 2000).

Lossless Huffman coding exploits statistical relationships in the data, and pairs short symbols to data values with high probability of occurrence and long symbols to those with low probability. For example, if the 16-bit integer value “0x2342” was the most commonly occurring data sample, a short 1-bit symbol can be given to it, such as “0”. Next, if “0x2455” was the second most occurring data sample, it might be given the 2-bit

symbol “10”. Hence, provided the probability mass density of the data, a compact binary representation of variable length can be used for compressed coding. Prior to generation of a Huffman lookup table, inherent structures in data can be exploited for achieving greater compression rates by using a de-correlation transform, such as (Daubechies) wavelet transform. The data compression and de-compression process is shown in Fig. 1.

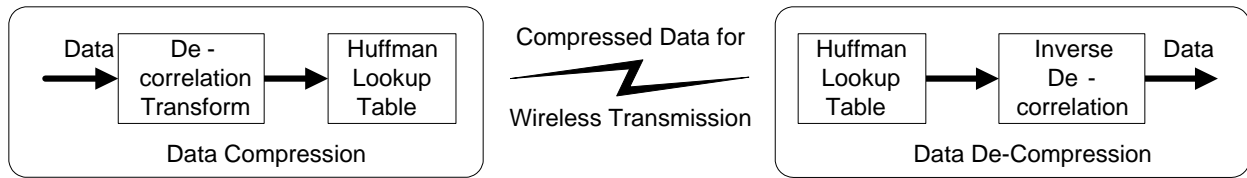


Fig. 1. Huffman compression of sensor data using wireless sensing nodes.

In the illustration example, Huffman coding is performed for data compression prior to wireless transmission. The compression results reported in Table 3 are obtained from the acceleration response acquired from a shake table test on a 5 degree-of-freedom laboratory structure subjected to sweeping sinusoidal and white noise inputs (Lynch and Law 2002). The acceleration data is recorded by the wireless sensing node using an effective 12-bit A/D converter. Table 3 shows the performance of lossless compression and the amount of energy saved through compressing data with the MPC555 and wirelessly transmitting the compressed record. Huffman coding compression is performed with and without wavelet transform for de-correlation. For the case of sweep excitation input, a compression rate of 61% was achieved after the initial record is de-correlated using wavelet transform. If the record is not de-correlated and internal statistical structures are not exploited in the creation of the Huffman coding lookup table, compression rate of approximately 71% can still be attained. However, for the white noise excitation, the response lacks an inherent structure that the de-correlation transform can leverage for compression. Since the time required to

compress data is practically negligible, the compression rate essentially determines the energy saved by the wireless radio in transmitting the compressed record.

Table 3. Compression of structural response data using Huffman coding.

Excitation Type	De-correlation	A/D Resolution [bits]	Total Record Size [bytes]	Compressed Record Size [bytes]	Compression Rate [%]	Energy Saved [%]
Sweep	None	12	1024	733	71.6	28.4
Sweep	Wavelets	12	1024	626	61.2	38.8
White	None	12	1024	795	77.6	22.4
White	Wavelets	12	1024	791	77.3	22.7

Lossy compression techniques are also widely available to further compress the data with “reasonable” data distortion within certain tolerance for a particular application. The time series data can be compressed by adding signal quantization between the de-correlation (or inverse de-correlation) filter and the Huffman coder. Table 4 shows the data compression results on a 16-bit time series data set collected from a high-speed boat (Sohn, *et al.* 2001), using a simple uniform quantizer, $x_q = \text{round}(x/q)$, with quantization (sampling) factors of 2 and 4. The results demonstrate that lossy compression can significantly reduce the size of the time series data. However, the distortion errors can propagate to subsequent analyses performed, in this example shown as the mean-square error (MSE) on the AR coefficients computed with the compressed data series.

Table 4. Lossy compression of measurement data using uniform quantization.

Compression Scheme	Compress Rate [%]	Data (MSE/Mean)	AR Coeff. (30) (MSE)
Lossless	83.0	0.0	0.0
Lossy (2)	65.6	10^{-8}	10^{-6}
Lossy (4)	58.8	10^{-7}	10^{-4}

In summary, with the availability of microcontroller onboard with the sensor nodes, significant energy can be saved by pre-processing the measurement data (i.e. directly embedding engineering analyses and/or compressing the raw sensor measurement data), prior to wireless transmission.

Section 3 In-Network Data Communication and Management

Compared to cable-based systems, wireless structural monitoring systems have a unique set of advantages and technical challenges. Besides the desire for portable long-lasting energy sources, such as batteries, reliable data communication is another key data management issue to be considered in the design and implementation of wireless structural monitoring systems. Section 3.1 discusses the important communication issues and metrics, such as transmission latency and communication range, for adopting wireless sensors in a structural health monitoring application. Furthermore, communication constraints need to be considered carefully in the selection of hardware technologies and the design of software/algorithmic strategies. Section 3.2 presents the utilization of a state machine concept to design reliable communication protocols in a wireless sensor network. In addition to data transmission, software code/programs can be wirelessly transmitted and dynamically migrated to sensor nodes on an as-needed basis. This dynamic code migration strategy can greatly ease the onboard memory limitations on the sensor nodes, and save efforts in the field for programming and incorporating new algorithmic developments on the sensor nodes. Section 3.3 describes the concept of dynamic code migration illustrated with an example prototype implementation.

3.1 Communication constraints in wireless sensor network

To quantify transmission latency in wireless sensor networks, it is necessary to first review the overall architecture of a wireless sensor node. A wireless sensor node usually consists of three functional modules, including sensor interface, computational core, and wireless communication (Lynch and Loh 2006). The sensor interface converts analog sensor signals into digital data, which is then transferred to the computational core. Besides a microcontroller/processor executing embedded program, external Static Random Access Memory (SRAM) is often integrated with the computational core to facilitate local data storage and analysis. A complete wireless communication process includes not only wireless data

exchanging among transceivers of different wireless nodes, but also onboard data exchanging between the computational core and the transceiver inside each wireless node.

The time required for a single transmission is defined as the wireless transmission latency between the sender and the receiver ends of the communication. A single transmission time refers to the period starting when the sender's processor begins pushing data to its transceiver, and ending when the receiver's processor obtains all data from its transceiver. As shown in Fig. 2, $T_{Onboard}$ represents the time required to transfer the data packet onboard, between the processor and wireless transceiver. Typical interfaces for the onboard transfer are either serial peripheral interface (SPI) or universal asynchronous receiver/transmitter (UART) interface. Once the sender's wireless transceiver obtains the first bit of data from the its processor, the transceiver starts preparing wireless transmission of the data packet. The communication latency, $T_{Latency}$, refers to the period starting when the sender side's processor begins pushing the first bit of data to its transceiver, and ending when the receiver side's transceiver is able to push out first bit of the data to its processor. Assuming both the sender and the receiver have the same onboard interfaces between their processor and wireless transceiver, $T_{Onboard}$ is usually the same for both sides.

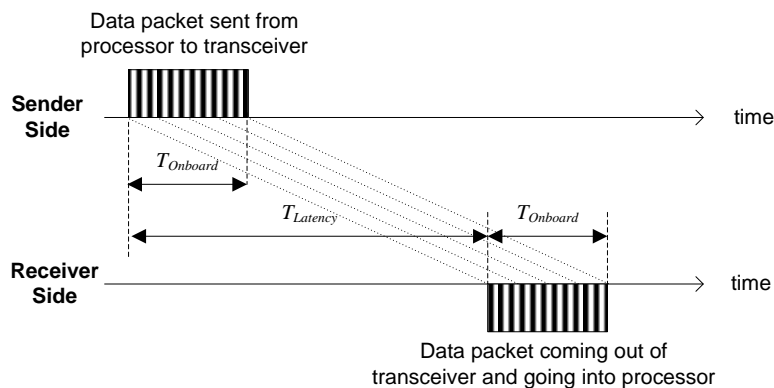


Fig. 2. Illustration of time consumption by a single wireless transmission.

Wireless transmission latency, $T_{Latency}$, is among the basic characteristics of a wireless transceiver. For example, latency of the 24XStream and 9XCite transceiver from Digi International Inc. are measured to be about 15 ms and 5 ms, respectively (Wang and Law 2007), while the Chipcon CC2420 wireless transceiver offers a lower latency of less than 2 ms (Swartz and Lynch 2009; Wang and Law 2011). The onboard data transfer time, $T_{Onboard}$, is determined by the transfer rate of the onboard interface. For example, a UART interface can be used with a data rate of R_{UART} at 38,400 bps (bits per second). UART is usually set to transmit 10 bits for every one byte (8 bits) of sensor data, including one start bit and one stop bit. Therefore, the data rate is equivalent to $R_{UART}/10$ bytes per second, or $R_{UART}/10,000$ bytes per ms. If a data packet to be transmitted contains N bytes, the single transmission time of the data packet can be estimated as:

$$T_{SingleTransm} = T_{Latency} + \frac{10,000N}{R_{UART}} [\text{ms}] \quad (1)$$

In the analysis herein, it is assumed that the bandwidth of the wireless transceiver is higher than or equal to the onboard transfer rate R_{UART} . Otherwise, wireless bandwidth becomes a bottleneck in the communication, and the wireless baud rate should be used to replace R_{UART} in Eq. (1). This amount of single transmission time typically has minimal effect in most monitoring applications, but can have noticeable effect when in-network analysis is performed. An in-network analysis may require frequent exchange of data among wireless nodes, while harnessing the embedded computing power of microcontrollers for collaborative data processing.

The other constraint, achievable wireless communication range, is related to the attenuation of the wireless signal traveling along the transmission path. The path loss $PL[\text{dB}]$ of a wireless signal is measured as the ratio between the transmitted power, $P_{TX}[\text{mW}]$, and the received power, $P_{RX}[\text{mW}]$ (Molisch 2005):

$$PL[\text{dB}] = 10 \log_{10} \frac{P_{TX}[\text{mW}]}{P_{RX}[\text{mW}]} \quad (2)$$

Path loss generally increases with the distance, d , between the transmitter and the receiver. However, the loss of signal strength varies with the environment along the transmission path and is difficult to quantify precisely. Experiments have shown that a simple empirical model may serve as a good estimate to the mean path loss (Rappaport and Sandhu 1994):

$$\overline{PL}(d)[\text{dB}] = PL(d_0)[\text{dB}] + 10n \log_{10} \left(\frac{d}{d_0} \right) + X_\sigma [\text{dB}] \quad (3)$$

Here $PL(d_0)$ is the free-space path loss at a reference point close to the signal source (d_0 is usually selected as 1 meter for ease of calculation). Parameter X_σ represents the variance of the path loss, which is a zero-mean log-normally-distributed random variable with a standard deviation of σ . Parameter n is the path loss exponent that describes how fast the wireless signal attenuates over distance. In essence, Eq. (3) indicates an exponential decay of signal power:

$$P_{RX}[\text{mW}] = P_0[\text{mW}] \left(\frac{d}{d_0} \right)^{-n} \quad (4)$$

where P_0 is the received power at the reference distance d_0 . Typical measured values of n are reported to be between 2 and 6 (Rappaport and Sandhu 1994).

A link budget analysis can be used to estimate the range of wireless communication (Molisch 2005). To achieve a reliable communication link, it is required that following inequality holds.

$$P_{TX}[\text{dBm}] + AG[\text{dBi}] \geq PL(d)[\text{dB}] + RS[\text{dBm}] + FM[\text{dB}] \quad (5)$$

where AG denotes the total antenna gain for the transmitter and the receiver, RS the receiver sensitivity, FM the fading margin to ensure the quality of service, and $PL(d)$ the realized path loss at some distance d within an operating environment. Table 5 summarizes the link budget analysis for a 900 MHz and a 2.4

GHz transceiver (both operating in a free unlicensed frequency band in the United States), and their estimated indoor ranges. The transmitted power for both transceivers, P_{TX} , is set at 1 Watt (30 dBm), the maximum power allowed by the Federal Communications Committee.

Table 5. Link budget analysis to two types of wireless transceivers (1 Watt transmitted power)

	900 MHz	2.4 GHz
P_{TX} [dBm]	30.00	30.00
AG [dBi]	4.00	4.00
RS [dBm]	-104.00	-105.00
FM [dB]	22.00	22.00
$\overline{PL} = P_{TX} + AG - RS - FM$ [dB]	116.00	117.00
$PL(d_0)$ [dB], $d_0 = 1$ m	31.53	40.05
$\overline{PL} - PL(d_0)$ [dB]	84.47	76.95
n	2.80	2.80
\bar{d} [m]	1039.72	560.23

In Table 5, a total antenna gain AG of 4 dBi is employed by assuming that low-cost 2 dBi whip antennas are used at both the transceiver and receiver. The receiver sensitivity RS and fading margin FM of the two wireless transceivers are chosen as typical values available from commercial transceivers. The free-space path loss at d_0 is computed using the Friis transmission equation (Molisch 2005):

$$PL(d_0)[\text{dB}] = 20 \log_{10}(4\pi d_0 / \lambda) \quad (6)$$

where λ is the wavelength of the corresponding wireless signal. Rappaport and Sandhu (1994) reported values of path loss exponent, n , that are measured for a number of different building types. The value for soft-partitioned office building ($n = 2.8$) is listed in Table 5 for both transceivers. Finally, assuming that the variance X_σ is zero, the mean communication range \bar{d} can be derived from Eq. (3) as shown at the bottom row of Table 5:

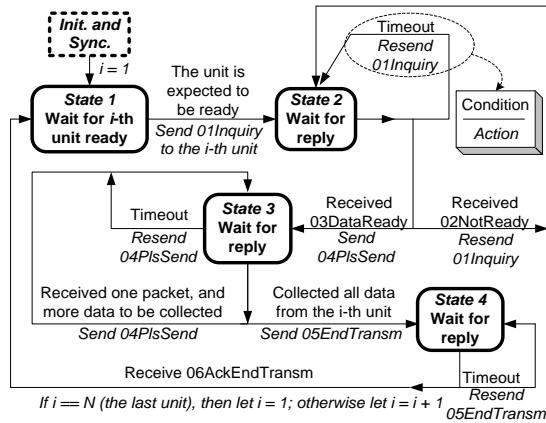
$$\bar{d} = d_0 10^{(\overline{PL} - PL(d_0)) / (10n)} \quad (7)$$

It is important to note the sensitivity of the communication range with respect to the path loss exponent n in Eq. (7). For instance, if the exponent of 3.3 for indoor traveling (through brick walls, as reported by Janssen & Prasad (1992) for 2.4 GHz signals) is used for the 2.4 GHz transceiver, the mean communication range reduces to 214.7m. Our experience in the design of wireless sensors and their placement has demonstrated the importance of considering the path loss estimation in physical implementation of a SHM system with reliable data collection (Wang and Law 2007).

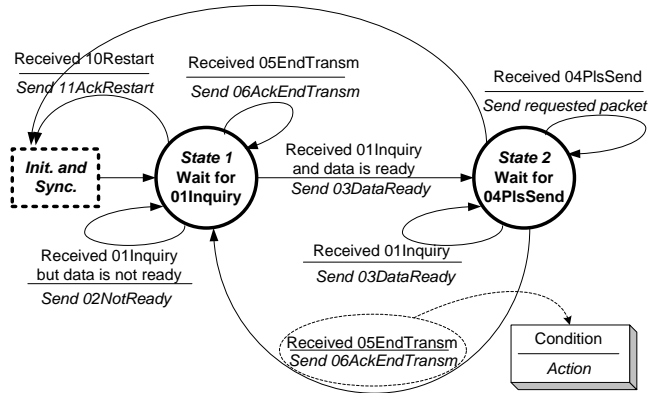
3.2 Communication protocol development through state machine concept

To ensure reliable communication between a wireless server and wireless nodes, or among multiple wireless nodes, data communication protocol needs to be carefully designed and implemented. A commonly used network communication protocol is the Transmission Control Protocol (TCP). Establishing a full duplex virtual connection between two endpoints, TCP is a sliding window protocol that handles both timeouts and retransmissions. Although TCP is highly reliable, it is usually too general and cumbersome to be employed by low-power and low data-rate wireless nodes. If the complete TCP suite is adopted in a low-power wireless sensor network, protocol overhead can cause long latency while transmitting each wireless packet, and significantly slow down the communication throughput. While general purpose operating systems for wireless sensors are available to conveniently implement TCP, for practical and efficient application in a wireless structural sensor network, a simpler communication protocol is much desirable to minimize the transmission overhead without overwhelming the limited resources of the sensor nodes. Furthermore, the protocol needs to be designed to ensure reliable wireless transmission by properly addressing possible data loss. Nevertheless, a light-overhead communication protocol designed for a wireless structural sensor network can still inherit many useful features of TCP, such as data packetizing, sequence numbering, timeout checking, and retransmission.

Finite state machine concepts can greatly facilitate the designing and programming of an efficient communication protocol. A finite state machine consists of a set of discrete states and specified transitions between the states (Tweed 1994). Taking the protocol between a wireless server and multiple wireless sensing nodes as an example, the state machine of either the server or a wireless node can only be in one of the possible states at any point in time. In response to different events, each machine transits between states. Fig. 3 illustrates part of a communication protocol for one round of sensor data collection, as described by state machine diagrams for the server and for the wireless nodes; note that the server and the nodes have separate sets of state definitions. During each round of data collection, the server collects sensor data from all wireless nodes. The detailed communication protocol for initialization and synchronization has been discussed by Wang, *et al.* (2007).



(a) State diagram of the server.



(b) State diagram of a wireless sensing node.

Fig. 3. Communication state machine diagrams for wireless structural health monitoring.

As shown in Fig. 3, at the beginning of data collection, the server and all the wireless nodes are all set in their own State 1. Starting with the first wireless node in the network, the server queries each node for the availability of data by sending the ‘01Inquiry’ command. If the data is not ready, the node replies ‘02NotReady’; otherwise, the node replies ‘03DataReady’ and transits to State 2. After the server ensures that the data from this wireless node is ready for collection, the server transits to State 3. To request a data segment from a node, the server sends a ‘04PlsSend’ command that contains a packet sequence number. One round of data collection from one wireless node is ended with a two-way handshake, where the server and the node exchange ‘05EndTransm’ and ‘06AckEndTransm’ commands. The server then moves on to the next node and continuously collects sensor data round-by-round.

Since the typical objective in structural monitoring is to transmit sensor data or analysis results to the server, in this set of state machine designs, the server is assigned the responsibility for ensuring reliable wireless communication. If the server sends a command to a wireless sensing node and does not receive

an expected response from the node within a certain time limit, the server will resend the last command again until the expected response is received. However, after a wireless sensing node responds to the server, the node does not check if the message has successfully arrived at the server, because the server is assigned with the overall responsibility. The wireless sensing node only becomes aware of data loss when the server queries the node for the same data again.

3.3 Dynamic wireless code migration

In addition to transmitting data sets, algorithms and software programs can also be transmitted over the wireless network. Specifically, the concept of dynamic wireless code migration is a powerful approach towards resource-efficient and reliable data processing and management at the sensor node. Wireless code migration, i.e. software programs (including dynamic behavior, actual state, and specific knowledge) physically migrating from one sensor node to another, has been used in a number of areas, such as mobile commerce, medical applications, and distributed traffic detection (Chen *et al.* 2009, Herbert *et al.* 2006, Mihailescu *et al.* 2002). The concept can be beneficial for wireless structural health monitoring as well.

Relatively sophisticated wireless sensing nodes are now available to support dynamic code migration. For example, Smarsly *et al.* (2011a) have employed a Java-based wireless sensing node, the SunSPOT node developed by Sun Microsystems, for a prototype implementation. Unlike common embedded applications for wireless sensor networks, which are usually written in low-level native languages (such as C/C++ and assembly language), the SunSPOT node contains a fully capable Java ME, which is widely used, for example, on advanced mobile phones. The computational core is an Atmel AT91RM9200 system-on-a-chip (SoC) incorporating a 32-bit ARM920T ARM processor with 16 kB instruction and 16 kB data cache memories and executing at 180 MHz maximum internal clock speed. The SunSPOT node also incorporates a Spansion S71PL032J40 chip that consists of 4 MB flash memory and 512 kB RAM.

Smarsly *et al.* (2011a) have coupled the migration-based approach with mobile multi-agent technology for wireless structural health monitoring. The wireless monitoring system consists of a 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 that collect structural sensor data (Fig. 4). The base node is connected to a local computer which includes a database and an information pool providing global information on the monitoring system and on the structure. Example information includes modal properties of the structure under monitoring, sensor nodes installed, and a catalog of data analysis algorithms.

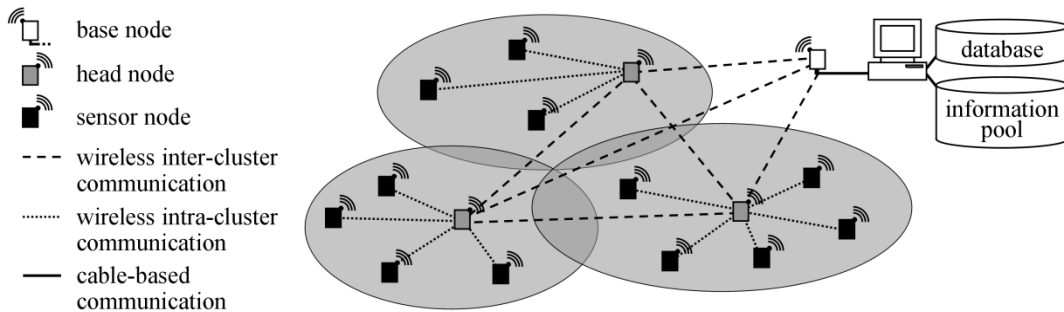


Fig. 4. Hierarchical architecture of the monitoring system.

Two basic types of mobile software programs are designed, namely “on-board agents” permanently residing at the head and sensor nodes, and “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 programs 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 record sensor data from the monitored structure, to perform simple routines for detecting suspected structural abnormalities, and to aggregate the sensor 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 (here a Cooley-Tukey FFT algorithm as introduced in the previous section). In the prototype implementation of dynamic code migration, a 96.4% reduction of wirelessly transferred data has been achieved compared to transferring the collected raw sensor data to a central server (Smarsly *et al.* 2011b). Furthermore, the memory consumption on a sensor node utilizing the code migration strategy can potentially be reduced comparing to the conventional execution of embedded algorithms.

Section 4 Persistent Data Management and Retrieval

This section provides an overview on data management and retrieval in SHM systems by means of a wind turbine monitoring system as an illustrative example (Smarsly *et al.* 2012a, Hartmann *et al.* 2011). Section 4.1 gives a brief description of the monitoring system. The basic steps of data collection, processing and archiving are discussed in Section 4.2, followed by an example of remote communication with the monitoring system in Section 4.3. Finally, in Section 4.4, specific features of the monitoring system are illuminated, such as the autonomous detection of sensor malfunctions.

4.1 A wind turbine monitoring system

A wind turbine monitoring system is prototypically implemented on a 500 kW wind turbine located in Germany. The system is designed to systematically assess the condition of the wind turbine, to detect damages and deteriorations, and to estimate its service lifespan. The monitoring system consists of an on-site hardware system and a software system for supporting distributed and remote access (Fig. 5). Installed in the 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 sensor data collected from the wind turbine.

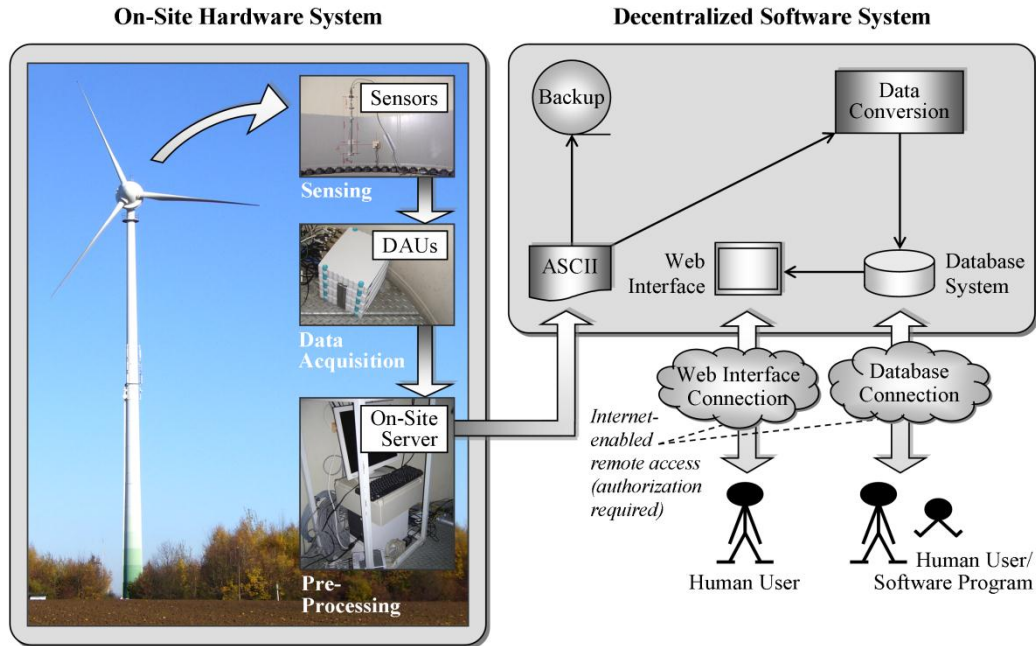


Fig. 5. Architecture of the wind turbine monitoring system.

The on-site hardware system is implemented to collect structural, environmental and operational data for assessing the condition of the wind turbine. 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. 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. In addition, six three-dimensional PCB-3713D1FD3G piezoelectric accelerometers, manufactured by PCB Piezotronics, are placed at five levels in the tower. On 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. Fig. 6 shows components of the on-site hardware system installed in the wind turbine tower as well as the anemometer.

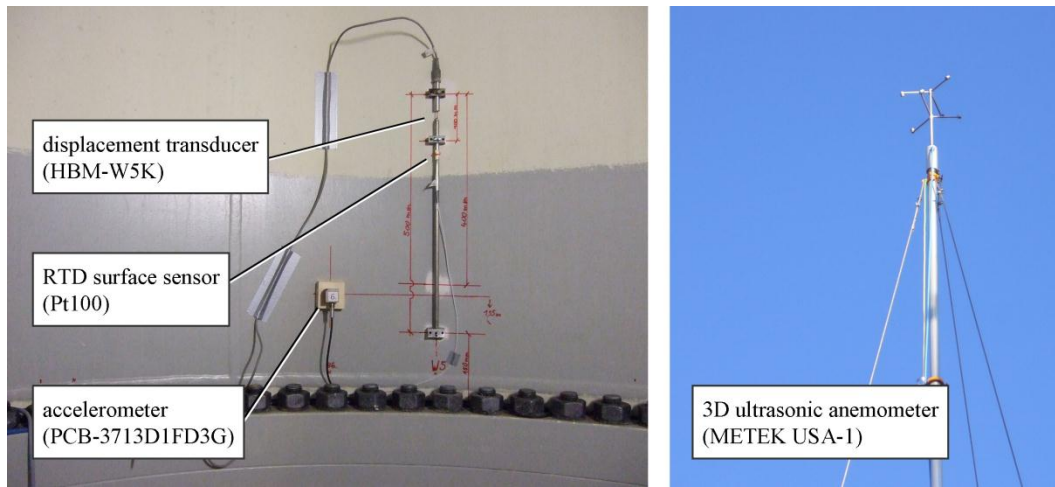


Fig. 6. Sensors inside the wind turbine tower (left) and three-dimensional anemometer (right) being part of the on-site hardware system.

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. For the acquisition of temperature data, three 4-channel Picotech RTD input modules PT-104 are deployed; for the acquisition of acceleration and displacement data, four Spider8 measuring units are used. Each Spider8 unit has separate A/D converters ensuring simultaneous measurements at sampling rates between 1 Hz and 9,600 Hz. All data sets, being sampled and digitized, are continuously forwarded from the DAUs to the on-site server for temporary storage.

The software system is installed on different computers at the Institute for Computational Engineering (ICE) in Bochum (Germany). The data collected by the on-site hardware system is forwarded to the software system using a permanently installed DSL connection. As shown in Fig. 5, the software system is designed to provide a persistent data storage, and to support remote access to the data sets and to the monitoring system. For data storage, the software system 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. A web interface connection is designed to facilitate interactions by the human users to remotely monitor the wind turbine. In addition, a database connection can be utilized by both human users and software programs to remotely access the monitoring system and to download and analyze the data.

4.2 Data processing and management

The data sets collected on the wind turbine are stored temporarily on the on-site server. The on-site server automatically creates local backups of all recorded data sets and, through a permanently installed DSL connection, transmits the raw data to the MySQL database installed on a main server at ICE. The data transmission is automatically executed by a “Cron” job scheduler, which is a time-based Unix utility running on the on-site server to ensure the periodic execution of tasks according to specified time intervals. When uploading the collected raw data to the main server for persistent archival, metadata is added to provide definitions of installed sensors, DAU IDs, output specification details, date and time formats, etc. This data conversion process is implemented using a commercial, open-source tool “PDI” (Pentaho Data Integration), which offers metadata-driven conversion and data extraction capabilities (Roldan 2009, Castors 2008). Fig. 7 shows the basic tasks defined for executing the data conversion process, which includes (i) starting the conversion service, (ii) data processing and (iii) moving the input data files for storage on a database.

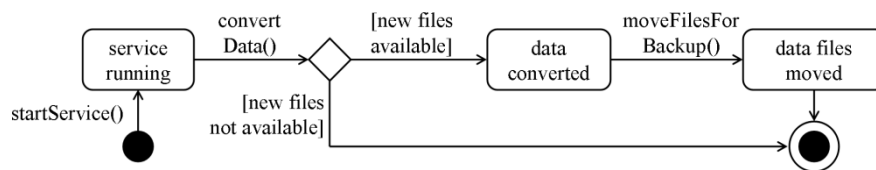


Fig. 7. Abridged illustration of the automated data conversion expressed in terms of a state machine protocol.

Once the data is successfully converted and stored in the MySQL database, an acknowledgement is sent from the main server at ICE to the on-site server in the wind turbine, whereupon the corresponding data set on the on-site server is deleted. During the conversion process, all data sets involved are automatically “locked” and cannot be accessed by software programs or by human users in order to avoid inconsistencies. Performance tests validating the automated data conversion process have been documented (Smarsly and Hartmann 2009a, 2009b, 2010). After being stored in the database at ICE, the data is available for remote access.

Fig. 8 shows the basic structure of the database. Database tables (such as “pt104”, “spider8” and “usa”) are defined for the DAUs (PT-104 modules, Spider8 measuring units, USA-1 anemometer) installed in the wind turbine. Each field in a database table represents one sensor connected to the DAU. For example, the database table “pt104”, shown in Fig. 8, comprises 10 fields which correspond to the data recorded by the PT-104 modules through the temperature sensors T1 to T10.

Monitoring Database			
Table	Records	Type	Size
spider8	1,724,319,519	MyISAM	432.6 GiB
usa	877,378,974	MyISAM	57.2 GiB
pt104	17,574,367	MyISAM	1.9 GiB

pt104

Field	Type	Null	Default	Comments
time	bigint(20)	No		
t1	float	No		
t2	float	No		
t3	float	No		
t4	float	No		
t5	float	No		
t6	float	No		
t7	float	No		
t8	float	No		
t9	float	No		
t10	float	No		

usa

Field	Type	Null	Default	Comments
time	bigint(20)	No		
v	int(11)	No		
d	int(11)	No		
z	int(11)	No		
t	int(11)	No		

spider8

Field	Type	Null	Default	Comments
time	bigint(20)	No		
B1_x	float	No		
B1_y	float	No		
B1_z	float	No		
B2_x	float	No		
B2_y	float	No		
B2_z	float	No		
B3_x	float	No		
B3_y	float	No		
B3_z	float	No		
B4_x	float	No		
B4_y	float	No		
B4_z	float	No		
B5_x	float	No		
B5_y	float	No		
B5_z	float	No		
B6_x	float	No		
B6_y	float	No		
B6_z	float	No		
W1	float	No		
W2	float	No		
W3	float	No		
W4	float	No		
W5	float	No		
W6	float	No		
B7	float	No		
B8	float	No		
B9	float	No		

Fig. 8. Structure of the monitoring database (extract).

During the automated conversion process, the basic statistics of the data sets, such as quartiles, medians and means, are computed at different time intervals and stored in the MySQL database (Table 6). As an example, Fig. 9 shows the database table “usa_3” which summarizes the statistics of the data sets collected by the USA-1 anemometer over a period of 3 seconds, as indicated by the suffix “_3”. The data sets and statistics are made available for analyzing the physical and operational states of the wind turbine structure.

Table 6. Characteristic values describing the secondary monitoring data.

Description	Suffix
Least value not considered an outlier	_least_non_outlier
First (lower) quartile Q1	_lower_quartile
Second quartile Q2 (median)	_median
Mean value	_mean
Third (upper) quartile Q3	_upper_quartile
Largest value not considered an outlier	_largest_non_outlier

Monitoring Database (aggregated)			
Table	Records	Type	Size
pt104_3	17,806,670	MyISAM	4.4 GiB
pt104_60	946,176	MyISAM	237.7 MiB
pt104_360	148,397	MyISAM	37.3 MiB
pt104_600	94,617	MyISAM	23.8 MiB
pt104_1800	29,686	MyISAM	7.5 MiB
pt104_21600	2,477	MyISAM	638.3 KiB
spider8_3	17,475,701	MyISAM	12.9 GiB
spider8_60	944,449	MyISAM	712.8 MiB
spider8_360	145,640	MyISAM	109.9 MiB
spider8_600	94,446	MyISAM	71.3 MiB
spider8_1800	29,135	MyISAM	22.0 MiB
spider8_21600	2,434	MyISAM	1.8 MiB
usa_3	17,740,632	MyISAM	2.0 GiB
usa_60	946,001	MyISAM	107.7 MiB
usa_360	147,847	MyISAM	16.8 MiB
usa_600	94,600	MyISAM	10.8 MiB
usa_1800	29,575	MyISAM	3.4 MiB
usa_21600	2,467	MyISAM	289.0 KiB

usa_3				
Field	Type	Null	Default	Comments
time	bigint(20)	No		
v_least_non_outlier	float	No		
v_lower_quartile	float	No		
v_median	float	No		
v_mean	float	No		
v_upper_quartile	float	No		
v_largest_non_outlier	float	No		
d_least_non_outlier	float	No		
d_lower_quartile	float	No		
d_median	float	No		
d_mean	float	No		
d_upper_quartile	float	No		
d_largest_non_outlier	float	No		
z_least_non_outlier	float	No		
z_lower_quartile	float	No		
z_median	float	No		
z_mean	float	No		
z_upper_quartile	float	No		
z_largest_non_outlier	float	No		
t_least_non_outlier	float	No		
t_lower_quartile	float	No		
t_median	float	No		
t_mean	float	No		
t_upper_quartile	float	No		
t_largest_non_outlier	float	No		

Fig. 9. Tertiary monitoring data composed of characteristic values (database structure).

4.3 Remote access to the monitoring system

Remote access to the monitoring data is provided through the web interface and through the 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. Fig. 10 shows the web interface displaying the monitoring data

collected during 24 hours on April 24, 2011. As shown in the left pane of the web interface, the monitoring data is graphically displayed as selected by the user through the control panel on the right pane. Using the control panel, the user can select the data collected by specific sensors, specify the time intervals to be plotted, conduct online data analyses, and export data sets. In Fig. 10, a time history of acceleration data is displayed, collected by a Spider8 measuring unit through a 3D accelerometer installed at the height of 63 m in the wind turbine tower (sensor B1). The web interface and the database connection provide easy access to the users and software modules for interacting with the monitoring system and conducting monitoring tasks.

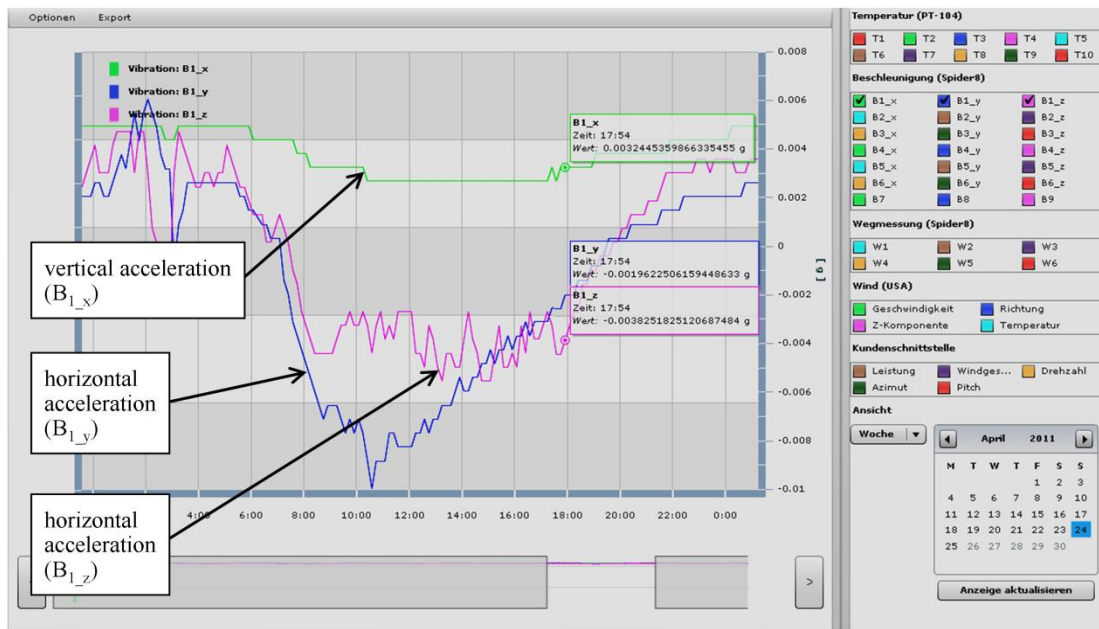


Fig. 10. Data sets remotely accessed via the web interface provided by the monitoring system (collected during 24 hours on April 24, 2011).

4.4 Detection of data anomalies and sensor malfunctions

One of the key issues in a SHM system is to ensure that the collected sensor data is reliable and, furthermore, the sensors or the DAUs are in good conditions. Typical malfunctions may be caused by

communication problems due to long-distance lines, breakdowns of sensors, or temporary power blackouts that affect the computer systems. A flexible and extendible multi-agent system is designed and connected to the existing monitoring system through the database connection (Smarsly *et al.* 2011c, 2011d, 2012b). The purpose of the multi-agent system is to self-detect malfunctions and to enable corrective actions. Once a malfunction of a DAU is observed, the system informs the responsible individuals about the detected defects through email alerts. The affected DAUs can be restarted remotely or replaced in a timely manner.

A multi-agent system consists of multiple interacting software components or “agents”. Software agents are characterized by two basic capabilities: *autonomy* and *flexibility*, which make multi-agent technology well suited for implementing distributed, real-time applications. An “autonomous” software agent is able to operate without any direct intervention by humans or other software systems, to control its actions, and to decide independently which actions are appropriate for achieving prescribed goals (Wooldridge 2009, Russell and Norvig 1995). In this implementation, the multi-agent system is developed based on the widely used Java Agent Development Framework JADE (Bellifemine *et al.* 2003, 2004, 2007). To ensure extensibility and interoperability, the multi-agent system is implemented in compliance with the specifications issued by the Foundation for Intelligent Physical Agents (FIPA). FIPA, the IEEE Computer Society standards organization for agents and multi-agent systems, promotes agent-based technology, interoperability of agent applications and the interoperability with other technologies (FIPA 2004, 2002a, b, c, d). Besides ensuring extensibility and interoperability, adhering to the FIPA specifications provides considerable advantages with respect to performance and robustness of the implemented multi-agent system. To illustrate the integration of data management system with self-monitoring functions, the following briefly describes two specific agents, namely an “interrogator agent” for sensor malfunction detection, and a “mail agent” for sending email alerts to responsible personnel.

One typical potential DAU malfunction, that causes an interruption of the data acquisition process, is often implicitly indicated by anomalies in the data sets, such as a long consecutive sequences of unusual, identical measurements. To detect such data anomaly, the interrogator agent at certain time intervals extracts and analyzes the data sets stored in the MySQL monitoring database (Smarsly *et al.* 2012a). The interrogator agent is connected to the monitoring database through the Java Database Connectivity (JDBC). Security of database requests and data transmissions is ensured by security mechanisms provided by the MySQL database, which requires password and username as well as secure drivers to be specified by the interrogator agent when trying to access the database. A set of configuration files defining interrogation parameters, database URL, database driver, sensor specifications, scheduled interrogation intervals, etc., is predefined and stored in the multi-agent system (Smarsly and Law 2012, Smarsly *et al.* 2012a). Fig. 11 shows an abridged example of how a connection is established using the required specifications of URL, database driver, username and password (`connect` method). Similarly, other complex queries can be executed by the agent (`interrogate` method). As shown in the figure, SQL command queries, issued to request for the sensor data or for performing data analyses, can be dynamically executed.

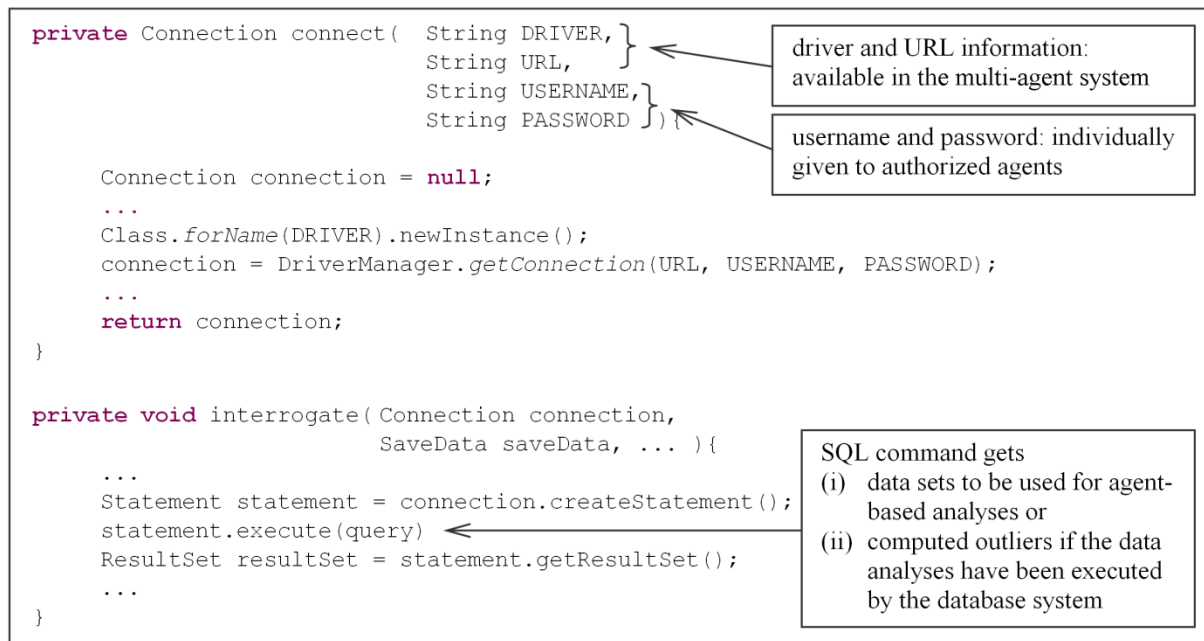


Fig. 11. Abridged example of establishing and utilizing a database connection.

Upon detecting possible data anomalies, the responsible individuals are immediately notified by the mail agent. On behalf of the interrogator agent the notification is 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 information from the interrogator agent about the observed anomaly. Furthermore, metadata stored in the configuration files, such as addresses of the recipients or the email server to be used, is acquired to compose the email, which is then automatically sent to the email clients using the Simple Mail Transfer Protocol (SMTP). For that purpose, the mail agent communicates with a SMTP server that is specified in the configuration files (and can be changed any time by the users, as approved by and registered with the multi-agent system). Secure email messages are ensured by username- and password-based authentications that the mail agent, like a human user, needs to specify when trying to access the SMTP server.

Since its initial deployment in 2009, the multi-agent system has reliably detected all malfunctions occurred and has notified the human individuals via email alerts. To illustrate the remote access to the monitoring system using the web interface, the detection of a malfunction of a temperature DAU as occurred on April 17, 2011, is shown in Fig. 12. Identical temperature measurements have been repeatedly stored by the DAU for a long period of time while other DAUs show variations on the temperature measurements. Fig. 13 shows a printed excerpt of the corresponding email, assembled and sent by the mail agent, which includes detailed information on the revealed anomaly. An internal system malfunction in the DAU (one of the PT-104 input modules installed in the wind turbine) was identified as the cause and the engineer, after having received the email alert, remotely restarted the DAU in a timely manner.

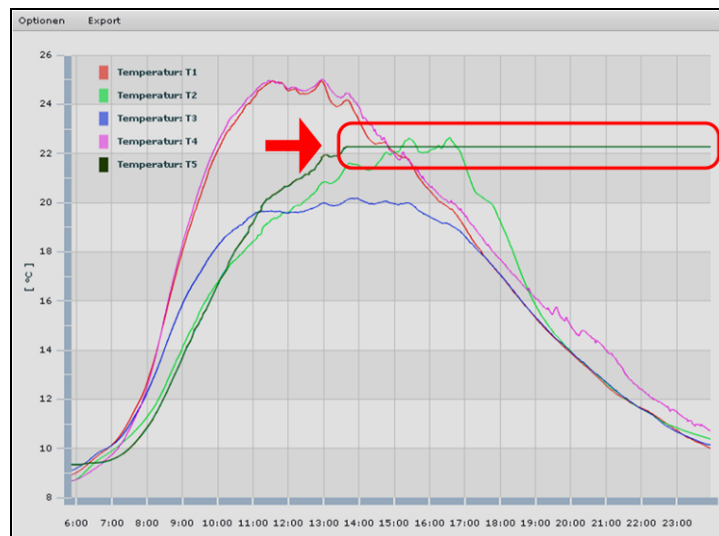


Fig. 12. Malfunction detected by the multi-agent system on April 17, 2011.

From: monitoring@inf.bi.ruhr-uni-bochum.de
Date: Sun, 17 Apr 2011 18:48:04 -0700
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.

(...)
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:
Database: lwea
DAU id: pt104
Date: Sun Apr 17 15:37:53 CEST 2011
Sensor: t5
Errornumber: 22.26




Fig. 13. Excerpt of the email alert, assembled and sent by the multi-agent system.

One distinct advantage of the agent-based approach is that the multi-agent system can be easily extended to accommodate additional functions. Variety of self-sensing and self-detection mechanisms can be built and data interrogation methods, such as fast Fourier transforms and autoregressive models, can be implemented as specialized agents readily integrated into the multi-agent system.

Section 5 Summary and Discussion

Measurement data collection, data transmission and persistent storage for data retrieval and access are among the key issues for successful deployment of a structural health monitoring system. With a few selected examples from prior wireless sensor research and from system deployment efforts, this chapter has discussed a number of issues faced by sensor data management of SHM systems. Issues have been discussed and demonstrated at three different operational levels: sensor level, in-network level, and system and database level.

- (a) At the sensor level, more specifically for wireless sensor nodes, energy constraints are an important issue when handling collected measurement data. To reduce the energy-demanding wireless transmission of data, onboard data processing and compression are viable approaches to optimize the performance at the sensor level. A dual-microcontroller architecture has been shown allowing the power-consuming and more sophisticated microcontroller to be switched on only when complicated computing tasks are needed. Algorithms can be embedded to pre-process the data prior to transmission.
- (b) With respect to data management occurring at the sensor network level, communication bandwidth and range are important issues to consider. Robust and light-overhead end-to-end communication protocols are necessary to achieve reliable data exchange. For flexible onboard data processing, wireless exchange of analysis code, i.e. dynamic wireless code migration, offers new opportunities for conveniently providing wireless sensor nodes with a wide variety of engineering analysis algorithms on demand.
- (c) At the system and database level, persistent data storage and convenient data retrieval from potentially off-site database is fundamentally important for the maintenance and operation of the monitored structure. Java Database Connectivity (JDBC) has been illustrated to provide online interfaces for querying a relational database. A multi-agent system allows a flexible and modular

approach for implementing external functions for processing the data and for performing operational activities, including self-monitoring and diagnosis.

In today's rapidly evolving technological world, new hardware/software platforms will continue to emerge. Although the technologies used as examples in this chapter are not necessarily the latest, the data management and processing issues discussed are commonly shared by different generations of structural health monitoring systems. Future energy harvesting technologies may alleviate energy constraints at the sensor node level, and empower more sophisticated in-network analysis. New data modeling techniques, such as the hierarchical data format HDF5 (HDF-Group 2011), provide enhanced capabilities to support complex and large-volume data sets. New technologies and analyses will eventually enable low-cost, pervasive, and ubiquitous sensing that supports data-rich SHM systems. It can be expected that future SHM systems will provide a much more detailed and accurate understanding of structural performance than today's systems.

Acknowledgements

This research was partially funded by the National Science Foundation under grants CMS-9988909 and CMMI-0824977 (Stanford University), CMMI-0928095 (Georgia Institute of Technology), and by the German Research Foundation (DFG) under grant SM 281/1-1, awarded to Dr. Kay Smarsly. Dr. Yang Wang was supported by an Office of Technology Licensing Stanford Graduate Fellowship. Prof. Jerome P. Lynch at University of Michigan kindly provided insightful advices to the development of the prototype wireless sensing system and Mr. Arvind Sundararajan, a former student at Stanford University, contributed to the data compression results presented in Section 2.3. In addition, the authors would like to express their gratitude to Professor Dietrich Hartmann of the Ruhr-University Bochum, Germany, for his generous support and for his leadership in the development of the wind turbine monitoring system, which was part of the DFG-funded research project HA 1463/20-1.

References

- Bellifemine, F., Caire, G., Poggi, A. and Rimassa, G. (2003). "JADE - A white paper." *EXP in search of innovation*, 3(3), 6-19.
- Bellifemine, F., Caire, G. and Greenwood, D. (2004). *Developing Multi-Agent Systems with JADE*, John Wiley & Sons, West Sussex, UK.
- Bellifemine, F., Caire, G., Trucco T. and Rimassa, G. (2007). *JADE Programmer's Guide*, [online]. <http://jade.tilab.com/doc/programmersguide.pdf>.
- Castors, M. (2008). *PDI Performance Tuning Check-List, Technical Report*, Pentaho Corporation, Orlando, FL, USA.
- Chen, B., Cheng, H. H., and Palen, J. (2009). "Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems." *Transportation Research Part C-Emerging Technologies*, 17(1), 1-10.
- Foundation for Intelligent Physical Agents - FIPA (2002a). *SC00001:2002 FIPA Abstract Architecture Specification*, FIPA, Alameda, CA, USA.
- Foundation for Intelligent Physical Agents - FIPA (2002b). *SC00026:2002 FIPA Request Interaction Protocol Specification*, FIPA, Alameda, CA, USA.
- Foundation for Intelligent Physical Agents - FIPA (2002c). *SC00037:2002 FIPA Communicative Act Library Specification*, FIPA, Alameda, CA, USA.
- Foundation for Intelligent Physical Agents - FIPA (2002d). *SC00061:2002 FIPA ACL Message Structure Specification*, FIPA, Alameda, CA, USA.
- Foundation for Intelligent Physical Agents - FIPA (2004). *SC00023K:2004 FIPA Agent Management Specification*, FIPA, Alameda, CA, USA.

Hartmann, D., Smarsly, K. and Law, K. H. (2011). "Coupling sensor-based structural health monitoring with finite element model updating for probabilistic lifetime estimation of wind energy converter structures." *Proceedings of the 8th International Workshop on Structural Health Monitoring*. Stanford, CA, USA, September 13-16, 2011.

HDF-Group (2011). *Hierarchical Data Format HDF5* [online]. <http://hdfgroup.org/HDF5/>.

Herbert, J., O'Donoghue, J., Ling, G., Fei, K. and Fok, C.-L. (2006). "Mobile agent architecture integration for a wireless sensor medical application." *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Hong Kong, China. December 18-22, 2006.

Janssen, G.J.M. and Prasad, R. (1992). "Propagation measurements in an indoor radio environment at 2.4 GHz, 4.75 GHz and 11.5 GHz," *Proceedings of IEEE 42nd Vehicular Technology Conference*, 2: 617-620, Denver, CO, May 10 - 13, 1992.

Lynch, J.P. and Law, K.H. (2002). *Decentralization of Wireless Monitoring and Control Technologies for Smart Civil Structures*. 140, John A. Blume Earthquake Eng. Ctr., Stanford University, Stanford, CA.

Lynch, J.P., Sundararajan, A., Law, K.H., Kiremidjian, A.S., Carryer, E., Sohn, H. and Farrar, C.R. (2003). "Field validation of a wireless structural monitoring system on the Alamosa Canyon Bridge," *Proceedings of the SPIE 10th Annual International Symposium on Smart Structures and Materials*, San Diego, CA, USA, March 2 - 6, 2003.

Lynch, J.P., Sundararajan, A., Law, K.H., Kiremidjian, A.S. and Carryer, E. (2004). "Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency," *Smart Materials and Structures*, 13(4): 800-810.

Lynch, J.P. and Loh, K.J. (2006). "A summary review of wireless sensors and sensor networks for structural health monitoring," *The Shock and Vibration Digest*, 38(2): 91-128.

McNeill, D.K. (2009). "Data management and signal processing for structural health monitoring of civil infrastructure systems," in *Structural Health Monitoring of Civil Infrastructure Systems*, V.M. Karbhari and F. Ansari (eds.), CRC Press, Boca Raton, FL.

Mihailescu, P., Binder, W. and Kendall, E. (2002). "MAE: A mobile agent platform for building wireless M-commerce applications." *Proceedings of the 8th ECOOP Workshop on Mobile Object Systems: Agent Applications and New Frontiers*, Malaga, Spain, June 10-14, 2002.

Molisch, A.F. (2005). *Wireless Communications*, John Wiley & Sons, IEEE Press, Chichester, West Sussex, England.

Press, W.H. (1995). *Numerical Recipes in C*, Cambridge University Press, Cambridge, England.

Rappaport, T.S. and Sandhu, S. (1994). "Radio-wave propagation for emerging wireless personal-communication systems," *Antennas and Propagation Magazine, IEEE*, 36(5): 14-24.

Roldan, M.C. (2009). *Pentaho Data Integration (Kettle) Tutorial, Technical Report*, Pentaho Corporation, Orlando, FL, USA.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Salomon, D. and Motta, G. (2010). *Handbook of data compression*, Springer, London ; New York.

Sayood, K. (2000). *Introduction to Data Compression*, Morgan Kaufmann Publishers, San Francisco, CA, USA.

Smarsly, K. and Hartmann, D. (2009a). "Real-time monitoring of wind energy converters based on software agents." *Proceedings of the 18th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar, Germany, July 7-9, 2009.

Smarsly, K. and Hartmann, D. (2009b). "Multi-scale monitoring of wind energy plants based on agent technology." *Proceedings of the 7th International Workshop on Structural Health Monitoring*, Stanford, CA, USA, September 9-11, 2009.

Smarsly, K. and Hartmann, D. (2010). "Agent-oriented development of hybrid wind turbine monitoring systems." *Proceedings of ISCCBE International Conference on Computing in Civil and Building Engineering and the EG-ICE Workshop on Intelligent Computing in Engineering*, Nottingham, UK, June 30 - July 2, 2010.

Smarsly, K., Law, K.H. and König, M. (2011a). "Resource-efficient wireless monitoring based on mobile agent migration." *Proceedings of the SPIE: Health Monitoring of Structural and Biological Systems 2011*, San Diego, CA, USA, March 7-10, 2011.

Smarsly, K., Law, K.H. and König, M. (2011b). "Autonomous structural condition monitoring based on dynamic code migration and cooperative information processing in wireless sensor networks." *Proceedings of the 8th International Workshop on Structural Health Monitoring*, Stanford, CA, USA, September 13-15, 2011.

Smarsly, K., Law, K.H. and Hartmann, D. (2011c). "Implementing a multiagent-based self-managing structural health monitoring system on a wind turbine." *Proceedings of the 2011 NSF Engineering Research and Innovation Conference*, Atlanta, GA, USA, January 4-7, 2011.

Smarsly, K., Law, K.H. and Hartmann, D. (2011d). "Implementation of a multiagent-based paradigm for decentralized real-time structural health monitoring." *Proceedings of the 2011 ASCE Structures Congress*, Las Vegas, NV, USA, April 14-16, 2011.

Smarsly, K. and Law, K.H. (2012). "Advanced structural health monitoring based on multi-agent technology." in *Computation for Humanity: Information Technology to Advance Society*, J. Zander and P. Mostermann (eds.) Taylor & Francis Group, LLC., Boca Raton, FL, USA.

Smarsly, K., Law, K.H. and Hartmann, D. (2012a). "A multiagent-based collaborative framework for a self-managing structural health monitoring system". *ASCE Journal of Computing in Civil Engineering*, 26(1), 76-89.

Smarsly, K., Hartmann, D. and Law, K.H. (2012b). "Structural health monitoring of wind turbines supervised by autonomous software components – 2nd level monitoring." *Proceedings of the 14th International Conference on Computing in Civil and Building Engineering*, Moscow, Russia, June 26-29, 2012.

Sohn, H., Farrar, C.R., Hunter, N. and Worden, K. (2001). *Applying the LANL Statistical Pattern Recognition Paradigm for Structural Health Monitoring to Data from a Surface-Effect Fast Patrol Boat*. LA-13761-MS, Los Alamos National Laboratory, Los Alamos, NM, USA.

Sohn, H. and Farrar, C.R. (2001). "Damage diagnosis using time series analysis of vibration signals," *Smart Materials and Structures*, 10(3): 446-451.

Swartz, R.A. and Lynch, J.P. (2009). "Strategic network utilization in a wireless structural control system for seismically excited structures," *Journal of Structural Engineering*, 135(5): 597-608.

Tweed, D. (1994). "Designing real-time embedded software using state-machine concepts," *Circuit Cellar Ink*, (53): 12-19.

Wang, Y. and Law, K.H. (2007). *Wireless Sensing and Decentralized Control for Civil Structures: Theory and Implementation*. 167, John A. Blume Earthquake Eng. Ctr., Stanford University, Stanford, CA.

Wang, Y., Lynch, J.P. and Law, K.H. (2007). "A wireless structural health monitoring system with multithreaded sensing devices: design and validation," *Structure and Infrastructure Engineering*, 3(2): 103-120.

Wang, Y. and Law, K.H. (2011). "Structural control with multi-subnet wireless sensing feedback: experimental validation of time-delayed decentralized H_∞ control design," *Advances in Structural Engineering*, 14(1): 25-39.

Wooldridge, M. (2009). *An Introduction to Multiagent Systems*, 2nd Ed., John Wiley & Sons, West Sussex, UK.