NSEL Report Series
Report No. NSEL-053
June 2022

Sudden-event Monitoring of Civil Infrastructure using Wireless Smart Sensors

Yuguang Fu
and
Billie F. Spencer, Jr.

NSEL
NEWMARK STRUCTURAL ENGINEERING LABORATORY

Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign

UILU-ENG-2022-1801

ISSN: 1940-9826

The Newmark Structural Engineering Laboratory (NSEL) of the Department of Civil and Environmental Engineering at the University of Illinois at Urbana-Champaign has a long history of excellence in research and education that has contributed greatly to the state-of-the-art in civil engineering. Completed in 1967 and extended in 1971, the structural testing area of the laboratory has a versatile strong floor/wall and a three-story clear height that can be used to carry out a wide range of tests of building materials, models, and structural systems. The laboratory is named for Dr. Nathan M. Newmark, an internationally known educator and engineer, who was the Head of the Department of Civil Engineering at the University of Illinois [1956-73] and the Chair of the Digital Computing Laboratory [1947-57]. He developed simple, yet powerful and widely used, methods for analyzing complex structures and assemblages subjected to a variety of static, dynamic, blast, and earthquake loadings. Dr. Newmark received numerous honors and awards for his achievements, including the prestigious National Medal of Science awarded in 1968 by President Lyndon B. Johnson. He was also one of the founding members of the National Academy of Engineering.

Contact:
Prof. B.F. Spencer, Jr.
Nathan M. and Anne M. Newmark Endowed Chair in Civil Engineering
2213 NCEL, MC-250
205 North Mathews Ave.
Urbana, IL 61801
Telephone (217) 333-8630
E-mail: bfs@illinois.edu

This technical report is based on the first author's doctoral dissertation of the same title, which was completed in August 2019. The second author served as the dissertation advisor for this work.

Financial support for this research was provided in part by the NSF SBIR under Grant #1913947, Nazarbayev University Research Fund under Grant #SOE2017003, ZJU-UIUC Institute Research under Grant #ZJU083650, Federal Railroad Administration under Grant #DTFR53-17- C00007, and the China Scholarship Council.

The cover photographs are used with permission. The Trans-Alaska Pipeline photograph was provided by Terra Galleria Photography (http://www.terragalleria.com/).

ABSTRACT

Many of the civil infrastructure damage scenarios involve sudden events, including natural disasters and anthropogenic hazards. Due to their unpredictable nature, many of these events go unnoticed or unreported; but their consequence can be catastrophic, resulting in damage/failure in a matter of seconds or hidden damage that accelerates structural degradation. An efficient structural health monitoring (SHM) system is thus critical to not only send early warning of these events to facilitate appropriate emergency response, but also enable rapid damage assessment to make informed decisions. Traditional monitoring systems using wired sensors can be prohibitive, mainly due to high installation costs resulting from onerous and expensive cabling networks. Wireless smart sensors (WSS) are an attractive alternative, as they have the potential to significantly reduce the cost. However, this solution remains elusive because of three inherent challenges: (1) most wireless sensors are duty-cycled to preserve a limited battery power; as a result, they will miss the onset of events when they are in power-saving sleep mode; (2) the wireless communication throughput is strictly limited, so time-consuming data transmission can result in large delay for subsequent damage assessment; (3) sensor malfunction often occurs in wireless monitoring systems, so the existing faulty data may result in false condition assessment, negatively affecting informed decisions. To address these challenges, this research aims to develop an intelligent wireless monitoring system consisting of five main components: ultralow-power on-demand sensing prototypes, high-precision time synchronization strategies, high-throughput live-streaming framework, effective sensor fault management, and a graphic user interface for rapid condition assessment and real-time data visualization. The capabilities of the developed monitoring system are validated through lab and field tests, demonstrating that the proposed solution is able to detect sudden events, provide high-fidelity synchronized data, and present structural damage assessment in an efficient manner.

CONTENTS

Chapter 1

INTRODUCTION

Civil infrastructure (e.g., roads, bridges, schools, dams, electrical grid, etc.) is widely recognized as crucial to our quality of life (e.g., employment, overall economic growth, etc.) (ASCE Steering Committee, 2007). However, due to shifting demographics and population growth, the infrastructure is increasingly overburdened, and structural damage exists widely in aging civil infrastructure. Many structural damage scenarios involve sudden events, such as natural disasters (e.g., earthquakes) and human-induced hazards (e.g., collisions, explosions, acts of terrorism). Due to their unpredictable nature, many of these events go unnoticed or unreported, but their consequence can be catastrophic. For example, a river barge collided with a railroad bridge in Mobile, Alabama, in 1993, resulting in collapse of the bridge 20 minutes later when an Amtrak train crossed, killing 47 people (Garner & Huff, 1997). If this collision had been detected immediately and timely structural assessment of the bridge made, then the deaths of these individuals may have been prevented. Therefore, the status of civil infrastructure should be tracked continuously, not only to provide early warning of sudden events for emergency response before catastrophic failures occur, but also to enable rapid structural assessment to mitigate extended maintenance and downtime costs.

SHM systems have the potential to help engineers characterize the status of structures by providing a detailed and continuous source of structural information. The functionalities of SHM systems include measuring structural response using sensors, extracting structural features, assessing structural conditions, and finally assisting end users to make informed decisions (Sohn, et al., 2003). Traditional monitoring systems use wired sensors (Qu, et al., 2006; Celebi, 2006; Okada, et al., 2009), which are capable of continuous monitoring. However, high installation cost remains one of the main drawbacks of such conventional solutions, mainly due to onerous and expensive cabling networks, which often ranges from \$5 K to \$20 K per channel. For example, the Bill Emerson Memorial Bridge monitoring system installed in 2004 cost a total of \$1.3 M for 86 sensors (Celebi, 2006).

High-fidelity wireless sensors offer tremendous opportunities to reduce costs and realize the promise of pervasive sensing for structural condition assessment. For example, the Jindo bridge monitoring system only cost \$200/channel (Jang et al., 2010). However, sudden event detection using wireless sensors remains elusive. For example, the monitoring system installed on the Golden Gate Bridge in 2006 was unable to detect the three earthquakes that occurred during the three-month monitoring deployment, because the sensors were either asleep to save limited battery power or busy with data transmission of ambient vibration measurement prior to earthquakes (Cheng & Pakzad, 2009). Even if the events were detected, due to limited communication throughput, the transmission of 1600 seconds data from 64 nodes (~20MB) back to the base station took over 9 hours, making rapid damage assessment extremely difficult (Pakzad et al., 2008). More detailed discussion about the main challenges of using WSS for sudden-event monitoring is presented in following paragraphs.

The first challenge is the constraint in power resources, which may result in missing of sudden events. Most conventional wireless sensors rely on batteries for power supply. Even if periodically recharged and replaced, the battery power source poses limits for always-on monitoring needed for detecting sudden events (they can occur at any time). For example, if using a 10,000 mAh lithium polymer battery, the MICAz installed on the Golden Gate Bridge can only continuously measure bridge responses for approximate 5 days. Accordingly, duty-cycling is generally employed to preserve limited battery power for WSSs which wake up periodically from deep sleep. The battery lifetime can then be extent to 3 months, if a duty cycle of 5% is considered. While perfect for periodic monitoring, duty-cycling is fundamentally incompatible with sudden-event monitoring; wireless sensors will miss the occurrence of sudden events when they are in power-saving sleep mode. Because the duty cycle is typically below 5% (Guo et al., 2011), this scenario is quite likely to occur. Increasing the available power through energy harvesting from the ambient environment does not provide an efficient solution, because it is intermittent and time-varying, which is not reliable to support continuous monitoring of structures. On the other hand, developing ultralow power sensors to save energy also do not fully address the challenge, because most of these sensors do not maintain high-fidelity sensing. Therefore, because of constraints in power resources, detecting sudden events using wireless smart sensors is challenging.

The second challenge is limited network throughput, which makes it extremely difficult to support rapid condition assessment of civil infrastructure subjected to sudden events. For large-scale structures, a network of sensors is required to obtain meaningful characterization of structural response. During the process, two main steps must be re-examined: time synchronization and data collection, because they may result in significant delay for subsequent condition assessment of civil infrastructure.

(1) *Time synchronization*. Each WSS has its own clock which does not share global data acquisition time. If using unsynchronized data, structural condition assessment will be negatively affected. For example, 30μs synchronization error results in a noticeable error in mode shapes in modal analysis (Krishnamurthy et al. 2008). To ensure accurate damage assessment, high-precision time synchronization is required, which includes clock synchronization and data synchronization. (Nagayama and Spencer, 2007; Li et al., 2016). In the conventional pre-sensing time synchronization, initial clock synchronization is required prior to sensing to estimate clock drift, which may take as much as 30s to complete, resulting in high risk of initial data loss for sudden event monitoring. Alternatively, post-sensing time synchronization moves clock drift estimation after measurement. As a result, initial data loss is minimized, but the synchronization still cannot be completed during sensing process to save time for rapid condition assessment. In sum, real-time time synchronization is difficult to achieve to support rapid condition assessment.

(2) *Data collection*. WSS uses radio communication to transmit sensor data back to the base station for subsequent data analysis. The capability of wireless communication allows the ease of sensor installation and flexibility of network topology, but it is unreliable and inefficient in nature. The unreliable communication has been extensively investigated, and it can be resolved by various reliable communication protocols, e.g., using acknowledgement-based

protocols (Nagayama & Spencer, 2007). But the inefficiency issue has yet been fully addressed, due to sharing of limited bandwidth among sensor nodes in WSSNs. This issue becomes more serious for multi-hop WSSNs. For example, in the 46-hop WSSN on the Golden Gate Bridge, the transmission of 1600s data from 64 nodes back to the base station took over 9 hours, resulting in significant delay for damage assessment (Pakzad et al., 2008). To eliminate the delay, real-time data acquisition has been explored. However, because both scheduling conflict in each node and radio interference among multiple nodes have not been settled completely and efficiently, the throughput is relatively low, which is not practical for large-scale network.

In sum, time synchronization and data collection are essential steps in wireless data acquisition, but they may result in either initial data loss or significant delay for rapid damage assessment during sudden-event monitoring. Thus, a new approach for wireless data acquisition for a network of sensor nodes is needed for sudden-event monitoring.

In addition, to ensure correct decision making in sudden-event monitoring, the collected measurement data must be high-quality without faults and processed autonomously with minimal delay. The mass production and wide adoption of micro-electro-mechanical systems (MEMS) sensors significantly reduce the cost of sensor platforms. But these sensors are vulnerable to harsh environments, and thus, sensor malfunction often occurs in wireless smart sensors, which is one of the primary concerns for long-term deployment of WSSNs. Serious sensor faults in the raw data may negatively affect SHM analysis and subsequent informed decisions, e.g., falsely indicating structural damage. Therefore, sensor fault management is important to ensure high-quality data for reliable informed decisions. Conventional data quality inspection involves user interactions and subjectivity, which is inefficient for rapid damage assessment. On the other hand, many autonomous data fault management strategies are either inefficient or inadequate. Particularly, one of the most important faults, spikes, is difficult to be detected, especially when the sparsity is small. And some of the strategies require several rounds of test before the fault is identified. After high-quality data is ensured, effective online condition assessments are needed to support informed decisions, which should have essential features of autonomous operation and fast analysis with minimal delay. Though extensive study has been made to develop various online condition assessment approaches, very few have been implemented on WSS and tested in the laboratory or in the field. As condition assessment strategies highly depend on the types of sudden events and application requirements, more discussion will be made in the applications in Chapter 7. In sum, effective sensor fault management strategies are required to ensure high-quality data, and subsequently, efficient online condition assessment strategies are needed to support reliable informed decision making.

The objective of this research is to develop an intelligent wireless monitoring system as a cost-effective solution for sudden-event monitoring, addressing the above challenges. When deployed on in-service civil infrastructure, events of interest will be detected, assessed, and reported in real-time. Finally, early warning of sudden events will be sent to structure owners for emergency response purposes, and high-fidelity actionable information will be provided for maintenance decisions in an efficient manner. In particular, this study will address all the elements of system design and validation, including: (1) ultralow-power on-demand sensing prototype for event-triggered sensing,

(2) efficient time synchronization strategies for high-precision synchronized sensing, (3) a live streaming framework for high-throughput real-time data collection, (4) effective sensor fault management to ensure high-quality data for correct decision making, and (5) a MATLAB-based application for real-time damage assessment and data visualization. The capabilities of the proposed system are validated in a lab test of seismic building monitoring and full-scale field test of bridge/ship impact detection.

Chapter 2 provides the research background in the field of study. An overview of structural health monitoring is first given. Then, a succinct review and comparison of wireless smart sensors is made. In particular, a next-generation wireless sensor platform, Xnode, and its associated software tool suite is introduced, which is leveraged in this study. Afterwards, the gaps of knowledge towards the development of an intelligent wireless monitoring system are identified in following aspects: power management, time synchronization, data transmission, sensor fault management, and online condition assessment.

Chapter 3 presents the development of ultralow-power on-demand sensing prototype, termed *Demand-based WSS*, to address the challenges of limited energy. Hardware and software considerations are first described. In particular, a programmable event-based switch is designed and integrated in the Xnode, utilizing a low-power trigger accelerometer. The approach can rapidly turn on the WSS upon the occurrence of a sudden event and seamlessly transition from low-power acceleration measurement to high-fidelity data acquisition. Both a lab test and a field application demonstrate the capabilities of the proposed prototype, which can detect the occurrence of sudden events with minimal power budget and low response latency. In contrast with other event-driven monitoring, the proposed solution uniquely addresses the initial data loss due to cold booting, by stitching together the pre- and post-event data across sensors without the need for a priori coordination.

Chapter 4 describes efficient time synchronization strategies for a network of *Demand-based WSSs* for sudden-event monitoring, leveraging the capabilities of the Xnode. The node's clock behavior is examined, and an efficient two-stage time synchronization strategy for traditional SHM is implemented as a baseline. Then, event-triggered synchronization for sudden-event monitoring is classified into three categories that call for different strategies: offline synchronization strategies for short- and long-duration monitoring, and an online synchronization strategy for rapid data analysis in long-duration monitoring. These strategies are experimentally validated to provide efficient and accurate (<20μs maximum error) synchronized sensing for sudden-event monitoring. Uniquely, the proposed strategy addresses scheduling conflicts in single nodes which perform multiple operations concurrently using preemptive multitasking, and it finally achieve autonomous real-time time synchronization.

Chapter 5 discusses new technologies to reduce the response latencies of WSSN for rapid damage assessment under sudden events. Two sources of response latencies have been addressed, including data transmission and data processing. A high-throughput live streaming framework is first developed and implemented in the Xnode. In addition, two efficient online condition assessment approaches are developed to support early structural damage estimation, including interstory drift estimation using acceleration record and sudden damage detection using digital signal processing techniques. Finally, a MATLAB-based application is designed to realize real-time data processing and visualization for end users. In particular, compared to other real-time data acquisition schemes, the developed live-streaming framework fully resolves two fundamental issues: scheduling conflicts in single nodes and radio interference among multiple nodes. The former issue has been discussed in previous chapter, whilst the latter issue is addressed using adaptive time division media accessor strategy.

Chapter 6 presents an effective three-stage strategy for sensor fault management to ensure reliable informed decisions under sudden events. First, a distributed similarity test is employed to detect sensor faults; this test is based on the similarity of the power spectral density of the data among sensors within a cluster of nodes. Second, an artificial neural network model is trained to identify the types of sensor faults. Third, sensor data is recovered from the identified faults by applying a correction function or replacing faulty data with estimated values. The performance of the proposed strategies is validated in a numerical analysis, using a set of field measurements collected from the Jindo Bridge. Different from current strategies which can only detect some specific faults, the developed 3-stage strategy is a systematic solution that can detect, classify and recover all of the three main types of sensor faults in an efficient manner.

Chapter 7 builds an integrated system, referred to as a wireless intelligent sudden-event monitoring system (WISEMS), for sudden-event monitoring. Two variations of the proposed system architecture are described, including real-time monitoring (online) scheme and post-event monitoring (offline) scheme. The system performance is validated by two applications: seismic building monitoring and bridge/ship impact detection. The system is proven to be a cost-effective wireless solution, which can capture the occurrence of sudden events, providing high-fidelity actionable information for emergency response and maintenance decisions in an efficient manner. Comparing to current monitoring strategies for sudden events, the integrated wireless system is not only cost-effective to be deployed on more common structures, but also capable to detect sudden events and support rapid condition assessments.

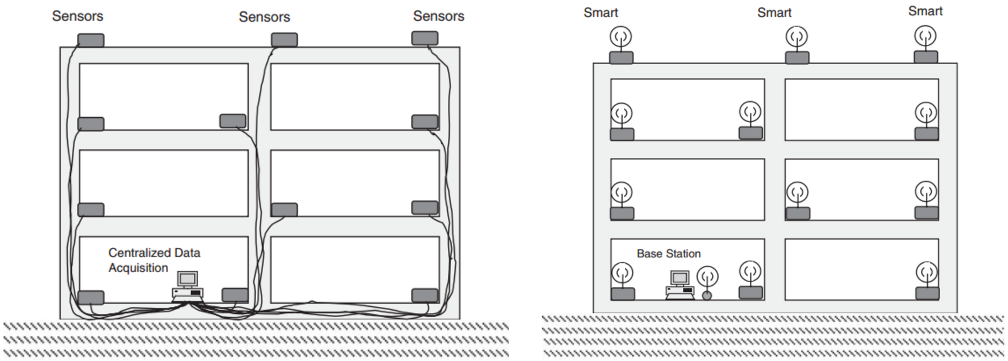
Chapter 8 summarizes the achievements reported in this thesis and discusses future research directions from system design to applications, based on the experiences of this study.

Chapter 2

BACKGROUND

2.1 Structural health monitoring

Structural health monitoring is the process of evaluating structural conditions from collected sensor data, by implementing damage detection strategies for aerospace, civil, and mechanical engineering infrastructure (Sohn, et al., 2003). An SHM system consists of multiple important components, including sensors, data acquisition systems, and a database for effective data management and health diagnosis (Ou & Li, 2010). The existing SHM systems can be categorized into two types, wired SHM systems and wireless SHM systems (Figure 2.1).



(a) (b)

Figure 2.1 SHM systems: (a) wired SHM system (b) wireless SHM system (Spencer, et al., 2004).

Wired SHM systems, the conventional approach, were exclusively used in the early stage of SHM development. Typical applications are health monitoring of the Golden Gate Bridge (Abdel-Ghaffar, et al. 1985), the Bill Emerson Memorial Bridge (Caicedo, et al., 2002; Celebi, 2006), and the Tsing Ma Bridge (Wong 2004, Wong 2007, Ni, et al. 2010). However, to investigate both global and local damage efficiently, a dense array of sensors is required. Such SHM systems need to be scalable in terms of cost and data management; therefore, wired SHM systems may no longer be attractive, primarily due to the high cost of installation and cabling for large scale structures (Spencer, et al., 2004). For example, the Bill Emerson Memorial Bridge monitoring system installed in 2004 cost a total of $1.3 M for 86 sensors, or an average installed cost of $15K per sensor (Çelebi, 2006). Advances in wireless communication and data processing techniques reduce the implementation cost and eliminate the onerous cabling work, making the WSS an attractive alternative to their wired counterparts.

2.2 Smart Sensor Platforms

2.2.1 Wireless Smart Sensors

A smart sensor is a device to intelligently acquire information (e.g., acceleration) from an object and transform it into electrical signals. Spencer, et al. (2004) summarize important features of a smart sensors to be (i) on-board central processing unit (CPU), (ii) small size, (iii) wireless, and (iv) the promise of being low-cost. As shown in Figure 2.2, smart sensors have three or four functional subsystems: sensing interface, computing core, wireless radio, and actuation interface (for some smart sensor platforms) (Lynch & Loh, 2006). A critical component of a smart sensor is its on-board microprocessor that has intelligent capabilities, such as digital processing, analog-to-digital conversions, and calculations (Kirianaki, et al.,2002).

Continuous efforts have been made over the past decade to develop several generations of wireless or smart sensor platforms and associated wing boards. Typical examples include: the MICA series — MICA (Hill and Culler, 2002), MICA2 (Crossbow, 2003), and MICAz (Crossbow, 2004), the Mote series — Imote (Kling, 2003; Zhao and Guibas, 2004; Kling et al. 2005), and Imote2 (Adler et al. 2005; Rice et al., 2010; Jo et al., 2011), and the WiMMS series — WiMMS (Straser et al., 1998), WiMMS II (Lynch et al., 2001), Narada (Swartz et al., 2005), and Martlet (Kane et al., 2014; Dong et al., 2014), which have been used in many SHM applications. Meanwhile, many wireless sensor platforms have been designed and provided commercially by Ember, MicroStrain, and Crossbow, etc. Typical sensor platforms are shown in Figure 2.3. Several applications of smart sensor platforms are shown in Figure 2.4. Most of these nodes (e.g., Narada, MICAz, and Imote2), however, are obsolete and no longer commercially available.

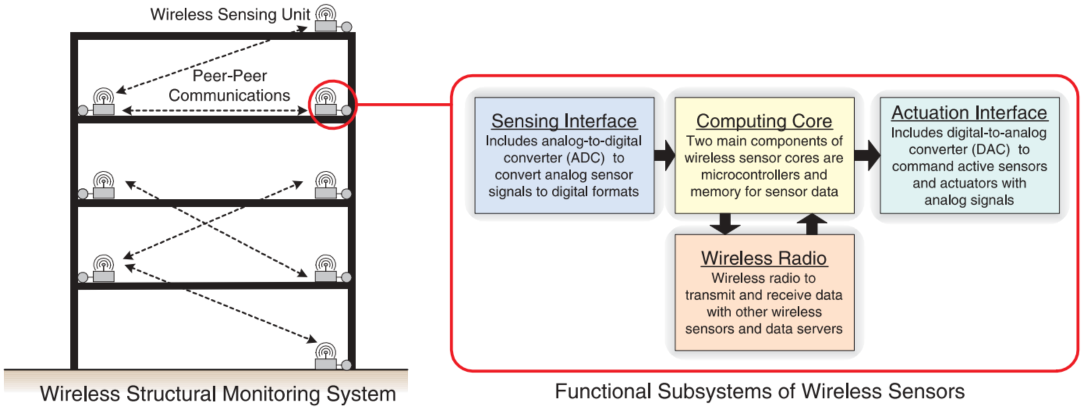

Wireless Sensing Unit
Peer-Peer Communications
Wireless Structural Monitoring System
Sensing Interface
Includes analog-to-digital converter (ADC) to convert analog sensor signals to digital formats
Computing Core
Two main components of wireless sensor cores are microcontrollers and memory for sensor data
Actuation Interface
Includes digital-to-analog converter (DAC) to command active sensors and actuators with analog signals
Wireless Radio
Wireless radio to transmit and receive data with other wireless sensors and data servers
Functional Subsystems of Wireless Sensors

Figure 2.2 Functional elements of a smart sensor platform (Lynch & Loh, 2006)

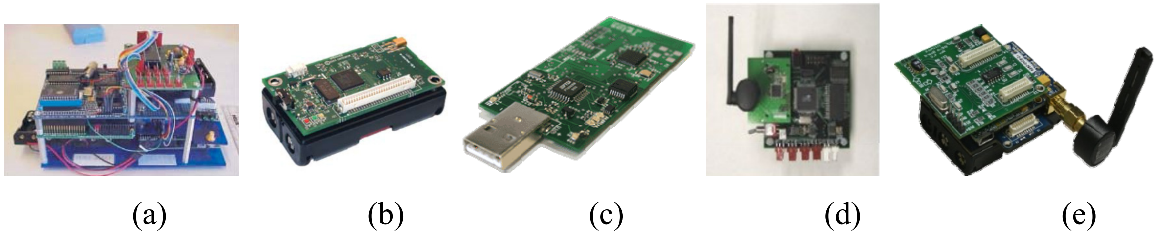


Figure 2.3 Typical smart sensor platforms: (a) WiMMS (Straser et al., 1998), (b) MICA2 (Crossbow, 2003), (c) Telos (Polastre et al., 2005), (d) Narada (Swartz et al., 2005), and (e) Imote2 (Rice et al., 2010).

Table 2.1 gives a succinct comparison between the major WSS platforms that have emerged in recent years. Realizing the demand of real-time application and high-speed on-board computation, Kane et al. (2014) proposed the design of a new wireless smart sensor platform, named Martlet. The node has two main critical features: (i) dual-core architecture well suited for structural control applications which require both standard data acquisition and real-time control execution, (ii) extensible hardware design enabling multiple wing boards to be interfaced with the motherboard. The AX-3D node developed by BeanAir, Inc. in 2016 is dedicated to vibration monitoring, especially in harsh environments. The node features an innovative antenna diversity design to improve radio communication, remotely programmable filters, and a supervision software (BeanScape) to provide real-time data visualization and automatic vibration analysis. The Waspmote v15 released in 2016 by Libelium, Inc. has several attractive functionalities, such as over-the-air programming, multiple radio module choices (XBee, 4G Cellular, WiFi, Bluetooth, and NFC/RFID), and cloud-based data management. These versatile functionalities enable its broad applications not only for SHM, but also for IoT markets. G-Link-200 from LORD Microstrain is another commercial node well-suited for SHM. In particular, the sensor node supports 20-bit resolution data acquisition with extremely low noise of $25\,\mu g / \sqrt{Hz}$ and a wireless communication range of up to 2km.

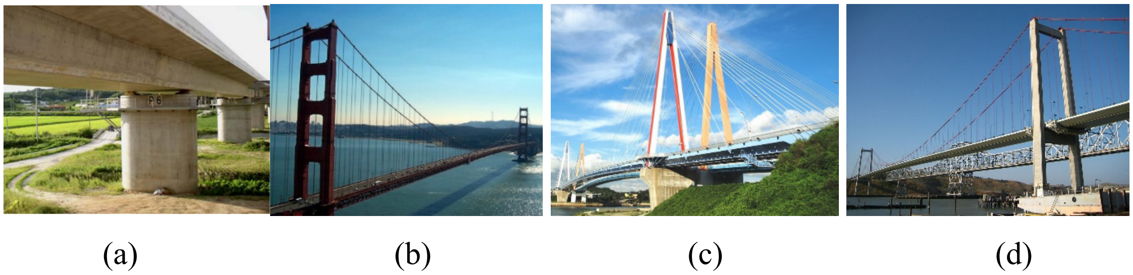


Figure 2.4 Applications of smart sensor platforms: (a) Geumdang Bridge (Lynch et al., 2006), (b) Golden Gate Bridge (Kim et al., 2007), (c) Jindo Bridge (Spencer et al., 2016), and (d) New Carquinez Suspension Bridge (Kurata et al., 2011).

2.2.2 Next-Generation Sensor Platform: Xnode

Though smart sensor platforms have been commercially available over a decade, only a few full-scale SHM applications have been realized. Typical applications are the Golden Gate Bridge monitoring (Kim, et al., 2007) and the Jindo Bridge monitoring (Rice, et al., 2010) (Figure 2.4). The obstacles to widely application of smart sensor platforms exist both in hardware and software. Lessons learned from those large-scale applications include: insufficient fidelity of data acquisition, unreliable communication, inefficient power management and data management (Spencer, et al., 2016). To address these obstacles, the Xnode, a next-generation sensor platform was designed and developed by Embedor Technologies.

Table 2.1 Comparison of advanced wireless platforms for SHM applications.

	Martlet (Kane et al. 2014)	AX-3D (BeanAir, 2016)	Waspmote v1.5 (Libelium, 2016)	G-Link-200 (Microstrain, 2017)	Xnode (Spencer et al., 2016)
DATA ACQUISITION					
Sensing channels	9	3	7	3	8
Sample rate	up to 3MHz	3.5kHz	0.5-1kHz	0-4kHz	1-16kHz
A/D resolution	12-bit	16-bit	16-bit	20-bit	24-bit
Time synch error	30us	2.5ms	-	50us	10us
Acquisition schemes	periodic sensing	periodic sensing	periodic /trigger sensing	periodic /trigger sensing	periodic /trigger sensing
DATA TRANSMISSION					
LOS range	500m	650m	1.6km	2km	>1km
Data rate	250kbps	250kbps	250kbps	NA	250-1000kbps
Transmission protocol	IEEE 802.15.4	IEEE 802.15.4	Zigbee/ IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.4
DATA PROCESSING/MANAGEMENT					
Bus size	16-bit	NA	8-bit	NA	32-bit
Clock speed	80MHz	NA	32kHz	NA	12-204MHz
Data memory	32GB	1 million data points	2GB	8 million data points	4GB
Operating system	State-machine	Proprietary	Proprietary	Proprietary	FreeRTOS
Fault tolerance[1]	×	×	×	×	√
Edge computing	×	×	NA	√[2]	√
POWER MANAGEMENT					
Sensing power	~190mA	~30mA	~18mA	NA	~170mA
Sleep power	~0.5mA	~0.03mA	~0.04mA	NA	~0.3mA
Energy harvesting	×	√	√	√	√

[1] The fault tolerance refers to sensor fault management, i.e., automatic detection & recover of sensor faults.

[2] It can output derived vibration parameters, including velocity, amplitude and crest factor.

As shown in Figure 2.5, the Xnode consists of three printed circuit board, including a processor board, a radio/power board, and a sensor board, which are designed to realize three functional subsystems, i.e., computational core, wireless transceiver, and sensing interface, respectively. While inheriting most advantages from the Imote2 in many aspects, the Xnode has several significant hardware improvements. Particularly, to enhance the capabilities of computation and concurrent execution, the Xnode was designed around a 32-bit NXP LPC4357 processor, based on an asymmetric dual-core ARM Cortex M4F/M0 architecture. The main core can be clocked up to 204MHz and features a hardware coprocessor that dramatically speeds up floating-point numerical computations. In addition, the newer Cortex processor features a large number of built-in peripherals and interfaces. The second core, M0, is considerably less capable, yet has access to the full array of peripheral devices and can perform many functions that on the Imote2 had to be performed by the main core. In addition, the comparison between the Xnode and other advanced sensor platforms can be found in Table 2.1.

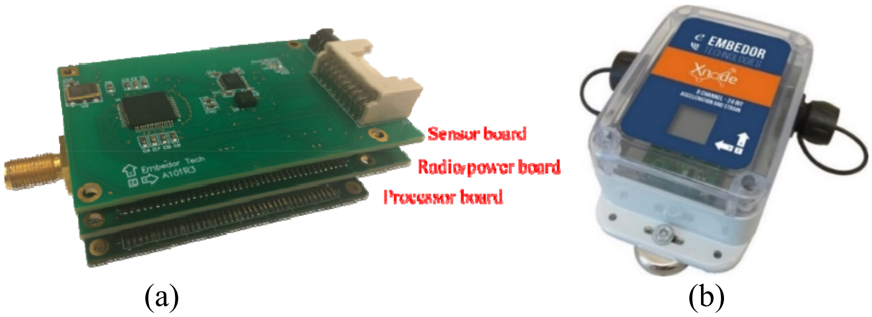


Figure 2.5 Xnode: (a) stacked modular boards; (b) weather-proof enclosure (Spencer, et al., 2016).

Many wireless platforms adopt event-driven operating systems to manage computing resources and host applications efficiently. Typically, TinyOS is the most popular event-driven operating system, which is designed for wireless platforms with extreme limited resources (Levis,, et al. 2005). However, these operating systems have problematic characteristics, including static resource allocation, a single-application focus, lack of real-time scheduling support, and dependence on a non-standard programming language. To address these limitations, the Xnode has been developed based on a real-time operating system (FreeRTOS), commonly used for industrial control systems and similar applications (Fu, et al., 2016).

More details of Xnode performance assessment and full-scale validation can be found in Fu et al. (2019).

2.2.3 The ISHMP Service Toolsuite

The ISHMP Services Toolsuite (http://shm.cs.uiuc.edu/), developed through collaboration between researchers in the Smart Structures Technology Laboratory and the Open Systems Laboratory at University of Illinois at Urbana-Champaign, is the software foundation for the iMote2-based remote sensing platform operating on TinyOS (Rice & Spencer, 2009). This toolsuite provides a suite of open-source services, employing service-oriented architecture (SOA) design principles. It consists of foundation service, application services and tools, and utilities services. The description of some typical services is summarized in Table 2.2.

Table 2.2 Typical services of ISHMP toolsuite (Rice & Spencer, 2009; Sim & Spencer, 2009).

Services		Functionalities & Characteristics
Foundation Services	Time Synchronization	A network-wide service for synchronizing local clocks in sensor nodes
	Unified Sensing	A sensing interface for Imote2 that support for various sensor boards
	Reliable Communication	A acknowledge-based protocol to ensure reliable radio communication in a wireless sensor network
	RemoteCommand	An efficient messenger for delivering commands, response or measured data between the gateway node and sensor nodes
Application Services	SyncSensing	A service to resample time-stamped sensor data from a node in a synchronized sensor network
	CFE	A numerical service to return the Correlation Function Estimate (CFE) via Fast Fourier Transform (FFT)
	ERA	A numerical service to perform Eigensystem Realization Algorithm (ERA)
	SSI	A numerical service to perform the covariance-driven Stochastic Subspace Identification (SSI) algorithm
	SDLV	A numerical service to perform Stochastic Damage Locating Vector (SDLV)
	FDD	A numerical service to perform Frequency Domain Decomposition (FDD)
Tools & Utilities	LocalSensing	A sensing service to allow sensor data to be collected while a single Imote2 is connected directly to the PC
	Autocomm	A basic terminal program for interfacing with the Imote2 through the Interface Board's USB port.
	TestServices	A numerical service that combines application services: CFE, ERA & SDLV
	SensingUnit	A service component for the coordination between the gateway node and multiple sensor nodes during a sensing process
	TestRadio	An application to test raw bidirectional radio communication between a sender node and a group of receiving nodes
	RemoteSensing	A network-wide distributed application to collect measurement data from multiple sensors
	DecentralizedData-Aggregation	An application to realize data acquisition and processing in a decentralized, hierarchical sensor network

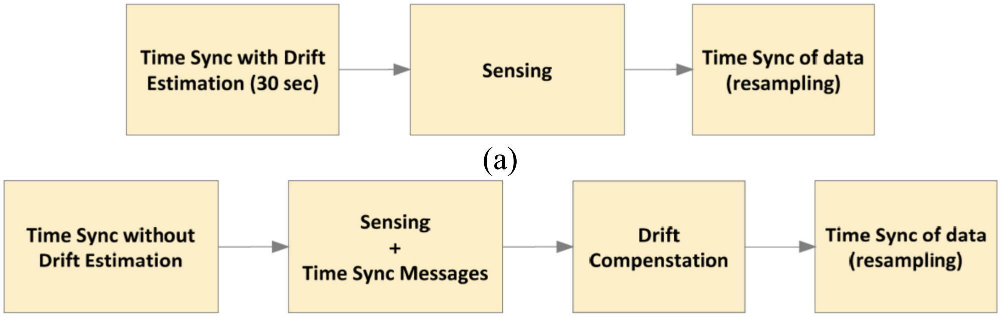
2.3 Networking and Implementation

2.3.1 Time Synchronization

In a WSSN, each WSS has its own clock, rather than sharing a global data acquisition time as in a wired data acquisition (DAQ) system. If the analysis is carried out using unsynchronized data, structural condition assessment is negatively affected. For example, 1ms time synchronization error will result in 3.6-degree phase delay of a mode at 10Hz measurement, or 36-degree phase delay at 100Hz measurement (Nagayama et al., 2007). The required synchronization level for SHM depends on the applications, and it is generally much more precise than typically available "for free" on a WSS with off-the-shelf integrated radio modules.

Continuous efforts have been made to develop various time synchronization strategies/protocols, which can be broadly classified into two categories: clock synchronization, and data synchronization. More precisely, the clock synchronization refers to a strategy in which sensor nodes exchange clock information to adjust and finally synchronize their local clocks to a global reference time. Well-known examples include the Reference Broadcast Synchronization (RBS) (Elson, 2003), Time-sync Protocol for Sensor Networks (TPSN) (Ganeriwal et al. 2003), and Flooding Time Synchronization Protocol (FTSP) (Maroti et al., 2004). However, clock synchronization does not guarantee the synchronization of sensor data obtained from each node (Nagayama & Spencer, 2007). In particular, most WSSs adopt event-driven operating systems which can introduce large uncertainties in task scheduling. Although the clock of sensor nodes is synchronized with the reference clock, the start-up time for sensing will not be the same across different sensor nodes. Other factors that introduce the synchronization errors include slight differences in sampling rates among sensor nodes and temporal variation in sampling frequency. Accordingly, data synchronization is required in addition to clock synchronization to postprocess the sensor data to eliminate the errors introduced by the above-referenced factors. To this end, Nagayama and Spencer (2007) proposed a resampling-based approach. Specifically, data samples are stored in blocks; each block is packed with a global time stamp which is obtained from local clocks after clock synchronization. These global time stamps are used to determine corresponding time offset and sampling rate for each data block. Subsequently, the data in the blocks are resampled through a finite impulse response filter and then combined to represent a complete synchronized measurement record. Another example of data synchronization is based on phase locking (Dragos et al., 2018), in which the expected relationship between the phase angles of acceleration measurement from multiple sensor nodes are utilized to estimate true time lags in data samples.

Several researchers have developed and implemented time synchronization strategies on various WSS platforms for SHM applications, as summarized in Table 2.3. The synchronization error compared here is the average error in single-hop (star topology) networks. Strictly speaking, precision is a function of many factors, as measurement length and temperature variation. Most studies only report the error of clock synchronization as an aggregate metric. Crucially, however, the data synchronization error can dominate the synchronization results, particularly when the clock synchronization error is minimal. Among the studies listed in the table, only three have implemented both clock synchronization and data synchronization, achieving high-synchronization precision. In Nagayama & Spencer (2007), a 30s clock synchronization period is set aside before sensing starts to estimate and then compensate local clock drift in each sensor node through linear regression; after sensing completes, time-stamped data is resampled to achieve data synchronization (see Figure 2.6a). However, this strategy is not feasible to deal with nonlinear clock drift due to temperature variation over an extended sensing period (Li et al., 2012). To address the challenge, Li et al. (2016) later proposed a post-sensing time synchronization, comprising two implementations. In the first implementation, beacon messages containing the global clock information from the reference node are broadcasted periodically during sensing to sensor nodes, where clock offsets are continuously estimated over the sampling period; finally, the recorded clock offsets are applied to achieve data synchronization through nonlinear regression analysis (see Figure 2.6b). As a result, the precision of time synchronization can be improved by up to 40% when nonlinear clock drift occurs in long-duration measurement. In the second one, a single beacon message is broadcasted from the reference node to sensor nodes before and after sensing, respectively, to estimate clock drift and offset. This simpler implementation is applicable when the radio environment is unstable over time during sensing, because massive beacon messages are not required to be transmitted during sensing. Using the same idea of integration of clock and data synchronization, Kim et al. (2016) developed a scalable time synchronization strategy using low-cost Global Positioning System (GPS) receivers. When a large structure is installed with multiple WSSNs, each of which has a gateway node, a GPS receiver inside each gateway node is utilized to trigger and synchronize multiple gateway nodes; clock and data synchronizations are then performed to achieve high-precision time synchronization within each WSSN.


Time Sync with Drift Estimation (30 sec)
Sensing
Time Sync of data (resampling)
Time Sync without Drift Estimation
Sensing + Time Sync Messages
Drift Compenstation
Time Sync of data (resampling)

(a)

(b)

Figure 2.6 Synchronized sensing procedures in ISHMP: (a) pre-sensing time synchronization (Nagayama & Spencer, 2007), (b) post-sensing time synchronization (Li et al., 2016).

Table 2.3 Summary of time synchronization of WSS for SHM applications.

Study	Platform	Clock synchronization		Data synchronization	
		strategy	error (µs)*	strategy	error (µs)
Ganeriwal et al. (2003)	MICA	TPSN	16.9	-	
	MICA	RBS	29.1	-	
Maroti et al. (2004)	MICA2	FTSP	1.5	-	
Wang et al. (2007)	Prototype	FTSP[+]	20	-	
Nagayama et al. (2007)	MICA2	TPSN	50	-	
	Imote2	FTSP	-	resampling	30
Kim et al. (2010)	Narada	FTSP[+]	7.4	-	
Sazonov et al. (2010)	WISAN	FTSP[++]	<23	-	
Bocca et al. (2011)	ISMO-2	u-Sync	10	-	
Kane et al. (2014)	Martlet	FTSP[1]	<30	-	
Li et al. (2016)	Imote2	FTSP	-	resampling	~16[+++]
Kim et al. (2016)	Imote2	FTSP	-	resampling	~30[+++]

[+] It is not the full-version of FTSP, as it does not compensate for clock drift.

[++] A modified version of FTSP: nodes are divided into clusters and cluster heads are synchronized by GPS.

[+++] The paper provides errors for different sensing duration, and the approximate average error estimated.

[*] Small clock synchronization error does not necessarily prevent significant data synchronization errors.

2.3.2 Data Acquisition Schemes

In contrast to their wired counterparts using cables for both power supply and data transmission, WSSs are independently-powered using batteries and communicate with each other wirelessly. Though the removal of cables allows flexible network topology and ease of installation, concerns of packet loss and response latency arise during wireless data transmission. More precisely, due to unreliable wireless radio communication, data loss occurs frequently; also, because of limited bandwidth, data transmission is generally time-consuming, resulting in delay for subsequent rapid damage assessment. These concerns are exacerbated in large-scale WSSNs, because more nodes transmit large amounts of data to the base station. This section presents a review of wireless data acquisition schemes.

Traditional WSSNs employ centralized data acquisition, mimicking the concept of a tethered monitoring system (see Figure 2.7a). In particular, sensor nodes store the data in their local memory after measurement is completed. The raw sensor data from each sensor node is then transmitted back to the base station in sequence. Subsequently, the base station collects the raw time history data and carries out data analysis (e.g., damage detection). This data transmission scheme is widely adopted in campaign-type monitoring and long-term periodic monitoring. However, it can take significant amount of time and result in severe data congestion.

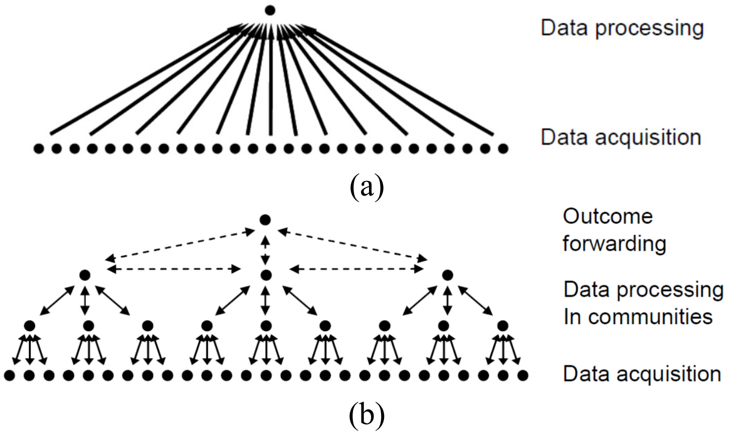


Figure 2.7 Data acquisition schemes: (a) centralized data acquisition, (b) decentralized data acquisition (Sim, 2011).

A typical example is the full-scale monitoring of Jindo Bridge using Imote2 smart sensors from 2009 to 2012 (see Figure 2.8). The first deployment has 70 sensor nodes, and it was expanded to a 113-node network in the second deployment. Most of the time, data acquisition was conducted in a centralized way using single hop communication, during which the *ReliableComm* service developed by Nagayama & Spencer (2007) was implemented to avoid data loss. The total transmission of 200s 50Hz measurement data from 23 sensor nodes took approximately 30 minutes (Rice & Spencer, 2009). This issue of time-consuming transmission is more serious for a WSSN employing multi-hop communication. For example, a network of 64 MicaZ smart sensors was installed on Golden Gate Bridge in 2006 (see Figure 2.9a). Due to the large span of the bridge and short range of node communication, the WSSN employed 46 hops and Straw service to reliably transmit measurement data back to the base station. In this multi-hop network, the bandwidth was reduced from 1.2kbps to 0.4kbps, as hops increased (see Figure 2.9b). As a result, 64 sensor nodes took over 9 hours to transmit 20MB data which correspond to 1600s 50Hz measurement data. To reduce the time for multi-hop communication, Nagayama et al. (2010) developed a multi-hop bulk data transfer protocol, named as Single-Sink Multi-hop (SSMH), allowing multiple neighbor pairs of nodes to transmit simultaneously by using multiple RF channels. Though a tenfold increase is achieved in the bandwidth of up to 58kbps, data transmission is still time-consuming. In addition, because radio communication is one of the main power-hungry processes, long data transmission time also reduces the battery life.

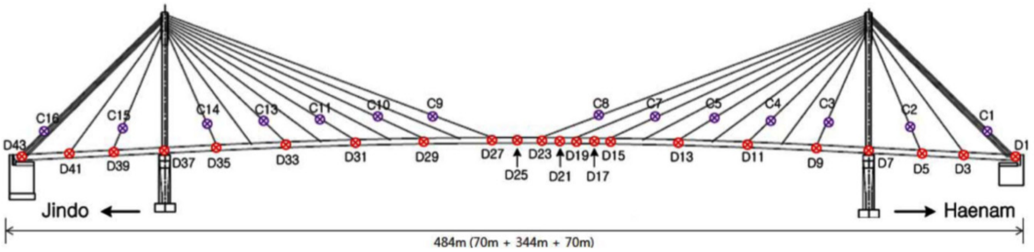


Figure 2.8 First deployment of a WSSN in Jindo Bridge (side view, Nagayama et al., 2010).

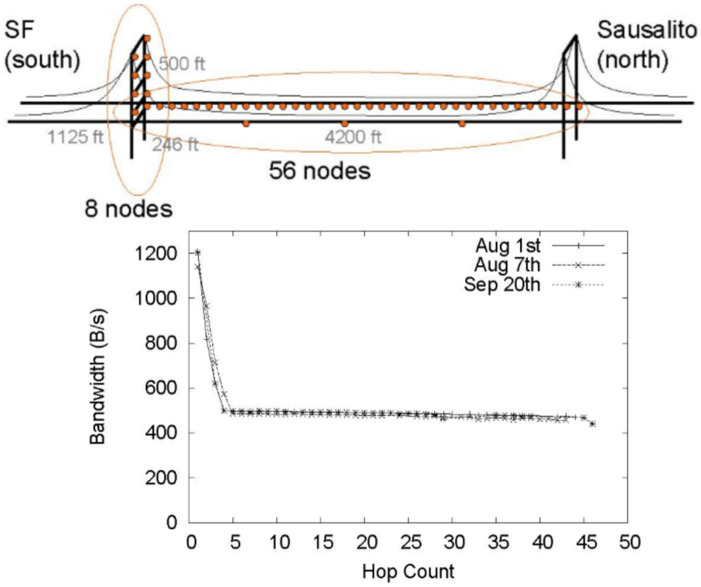


Figure 2.9 WSSN in Golden Gate Bridge (Kim et al., 2007): 64-node 46-hop deployment, and bandwidth of Straw service.

Decentralized data acquisition has been explored extensively over the years to reduce transmission data by processing raw measurement data on the sensor node and only transmitting useful information, which can expedite the data retrieval (see Figure 2.7b). Depending on the type of processing strategy involved, the decentralized data acquisition can be divided into two ways: decentralized independent processing and decentralized coordinated processing. For decentralized independent processing, measurement data is analyzed on each sensor node without communicating with the other nodes. After processing, the data size is significantly reduced; the process results (e.g., FFT) are then transmitted back to the base station in an efficient manner. For example, Sim et al. (2013)

developed autonomous cable tension monitoring system, in which each WSS calculates cable tension independently and sends the results back to the base station. Decentralized coordinated processing is employed to capture local spatial information for more complicated analysis. In this approach, the WSSN is organized in a hierarchical manner with three layers: leaf nodes, cluster heads, and the gateway node. Based on physical locations or vibration levels, a cluster is built among several leaf nodes, one of which is selected as the cluster head. The cluster head coordinates the data transmission and processing within the cluster, and it finally transmits the results back to the gateway node. Various high-efficiency SHM applications have been developed by employing decentralized coordinated processing, such as automated decentralized modal analysis (Sim et al., 2010), decentralized stochastic modal identification (Jo et al., 2011), distributed computing strategy (DCS) for damage detection (Gao & Spencer, 2005; Nagayama & Spencer, 2007), decentralized receptance-based damage detection (Jang et al., 2012), holistic approach for damage detection by integrating a decentralized computing architecture with the Damage Localization Assurance Criterion algorithm (Hackmann et al., 2012). The benefit of using decentralized data acquisition is significant. For example, Sim (2011) demonstrated that the decentralized Random Decrement Technique based modal analysis can reduce the transmission data by 95.7%, compared to its centralized counterpart. Compared to centralized damage detection strategies, the decentralized monitoring system developed by Hackmann et al. (2012) can reduce the latency and energy consumption by 65.5% and 64.0% respectively, significantly increasing the system's projected lifetime.

Real-time data acquisition is an attractive alternative, because it can eliminate delay for data transmission, which is essential for applications with tight timing requirement (e.g. real-time data visualization). However, due to constraints in operating systems (OS) and stringent synchronization requirements, real-time data acquisition is still a big challenge for WSSNs. In particular, as mentioned in Section 2.2.2, the widely adopted event-driven OS in WSSNs has very limited support for real-time applications, because the scheduler runs in a FIFO manner and uncertain delay of task executions is inevitable. These OS constraints result in the realized real-time data acquisition having a much lower bandwidth than theoretically predicted. A wireless sensor prototype was developed using a multithreaded OS (Lynch et al., 2006; Wang et al., 2007), in which interrupts preempt any non-real-time task to service the software routine associated with the sensing process to ensure a precise schedule without delay. Ring buffers are utilized to realize real-time data acquisition; however, the bandwidth is limited to prevent the overwrite of the buffer. Dual-stack memory is later implemented, in which data is stored in one stack while data is sent from the other stack. The role of two stacks alternates as soon as one stack is filled with newly collected data (see Figure 2.10). The system was estimated to achieve near-synchronized 16-bit data acquisition of up to 24 sensing channels at a sampling rate of 50Hz in real-time, which has yet to be validated in tests. Based on transmission scheduling, the system developed by Whelan & Janoyan (2009) was validated to support real-time data acquisition of 12-bit 40 channels at a sampling rate of 128Hz. The scheduling of transmission is based on time synchronization, which is however only precisely maintained for several minutes. After comprehensive timing analysis, Linderman et al. (2013) implemented Time-Division Medium Access (TDMA) on Imote2s to enable high-throughput real-time data acquisition of 12 channels at 40Hz sampling rate (see Figure 2.11). The framework was based on TinyOS which imposed limits for throughput, and the system does not guarantee data lossless transfer. In addition, Linderman et al. (2015) designed a SHM-SAR board to reduce the latency of filter from 30ms to approximate 200us for real-time data acquisition. Kane et al. (2014) developed a dual-core wireless sensing node, Martlet, supporting not only real-time data acquisition but also structural control, but the maximum throughput of the node was not reported. A comparison of various real-time data acquisition strategies using WSS is presented in Table 2.4.

In sum, a high-throughput high-efficiency data acquisition for WSSN is still challenging.

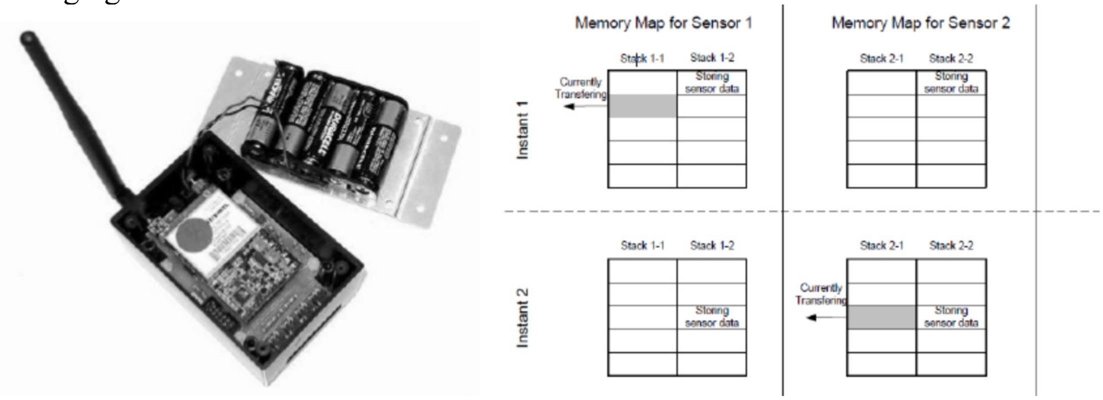


Figure 2.10 Wireless node prototype and dual-stack memory allocation (Wang, et al., 2007).

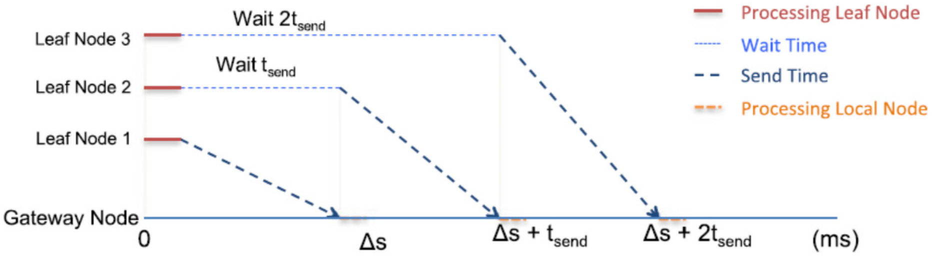


Figure 2.11 TDMA implementation for real-time data acquisition (Linderman, et al., 2013).

Table 2.4 Summary of real-time data acquisition services using WSS.

Study	Platform	Data delivery	Data bit	Channels	Sampling rate (Hz)	Throughput (kbps)
Wang et al. (2007)	Prototype	Real-time	16	24	50	19.2
Niu et al. (2009)	IRIS	Reliable real-time	10	34	50	17
Whelan & Janoyan (2009)	Prototype	Reliable real-time	12	40	128	61.4
Linderman et al. (2013)	Imote2	Real-time	16	12	40	7.68
		Near-real-time	16	72	40	46

2.4 Data Processing and Analysis

2.4.1 Sensor Fault Management

In long-term SHM using WSSN, sensor nodes are expected to work autonomously, and malfunctioning of sensors are likely to occur frequently, due to issues with battery, harsh environment or calibration errors. There are two levels of Malfunctioning: sensor faults and sensor failure (Lo, et al., 2015). More specifically, in the first level, sensors continue to work but provide abnormal data; in the second level, sensors become inactive. This study is focused on sensor faults.

Extensive research has been conducted to handle sensor faults in WSSNs. A comprehensive review of fault diagnosis techniques in WSSNs can be found in references (Sharma et al., 2010; Mahapatro & Khilar, 2013). Most of the research is directed toward static or pseudo-static measurands (e.g., temperature (Moustapha & Selmic, 2008)) or other general-purpose applications (e.g., health-care (Bourdenas & Sloman, 2010)), which is quite distinct from the type of data required for most SHM applications. In the context of WSSN in SHM applications, several fault diagnosis techniques have been developed for sensor fault management, which can be classified as centralized and decentralized. In centralized methods, a base station collects the information of all sensors and execute fault diagnosis process. Most of centralized methods employ models to predict sensor outputs. A sensor is considered to be faulty, if the measured and predicted outputs are significantly different. For example, Chang et al. (2017) utilized autoregressive modelling techniques to develop a bank of Kalman estimators which was used to estimate the sensor outputs. The residuals between the estimated and the measured responses were then applied to identify three types of sensor faults, including the additive, multiplicative, and slowly drifting faults. In addition, machine learning techniques have been applied to classify sensor faults directly. Zhao et al. (2011) developed a fault diagnosis model using a support vector machine (SVM), of which the parameters were determined by chaos particle swarm optimization algorithm. Four types of faulty sensor readings (shock, bias, short circuit and shift) were successfully recognized using pre-trained SVM. Instead of examining the output directly, Yu et al. (2014) collected the symptoms of faulty data using rough set theory to train an SVM. The faulty data was finally classified through the trained SVM model. In addition, some techniques have been proposed to deal with the denoising problems or sensor outliers (Yang & Nagarajaiah, 2014). Although these methods have high accuracy, they require transmission of numerous data packets and hence large amount of energy is consumed. Moreover, they are difficult to be implemented in WSSNs running decentralized SHM applications.

In decentralized methods, sensor fault diagnosis is conducted within a group of sensors. Recently, several decentralized model-based methods have been published in the context of SHM applications. Lo et al. (2015) proposed a method of Kalman filter based group test, which was verified to detect various faults, such as spike, non-linearity, mean-drift and excessive noises. Their method achieved similar accuracy but required fewer number of tests, compared to non-group testing method. Lo et al. (2016) also presented a decentralized approach carried out locally within a pair of sensor nodes to detect and isolate nonlinearity faults. In particular, they presented a low-complexity algorithm to solve the largest empty rectangle which could achieve high accuracy to detect the nonlinearity faults (see Figure 2.12). These two strategies both utilized raw sensor data in time domain for fault diagnosis. In the paper (Smarsly & Law, 2014), an Artificial Neural Network (ANN) model was trained to predict sensor outputs by utilizing the correlations between multiple sensor outputs in time domain. This method was implemented on sensor nodes, and it was validated by identifying drift fault and bias fault. Considering that using raw sensor outputs in time domain requires large data traffic between sensor nodes, Dragos & Smarsly (Dragos & Smarsly, 2016) improved the ANN model by using the correlation between the Fourier amplitudes of peaks at resonant frequencies (see Figure 2.13). The improved method was able to detect and identify bias, drift, complete failure, gain, and precision degradation. However, the efficiency of this method highly depends on the magnitude of amplitude residuals of peak value in the frequency domain; it may not be able to identify spike faults found in the Jindo Bridge data, because the change of the peak values due to spike faults is considered to be small. In some other methods, models are not applied, and sensor measurement in close proximity are assumed to have similar characteristics (Ding et al., 2005; Luo et al., 2006). A sensor is considered to be faulty if its behavior is significantly different from other sensors. However, most of them requires large number of tests based on the order of size of network (Ni et al., 2009), which may not be practical for large-scale WSSNs. In addition, most of the approaches are merely designed for fault detection and identification, but little attention has been paid to develop the techniques of fault recovery.

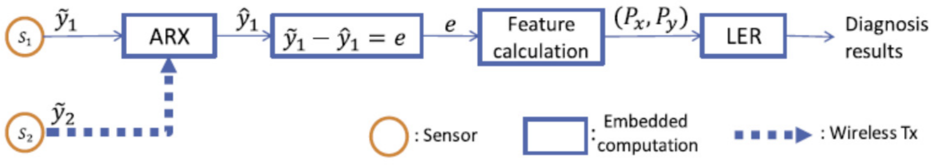


Figure 2.12 Distributed model-based sensor fault diagnosis: (Lo, et al., 2016).

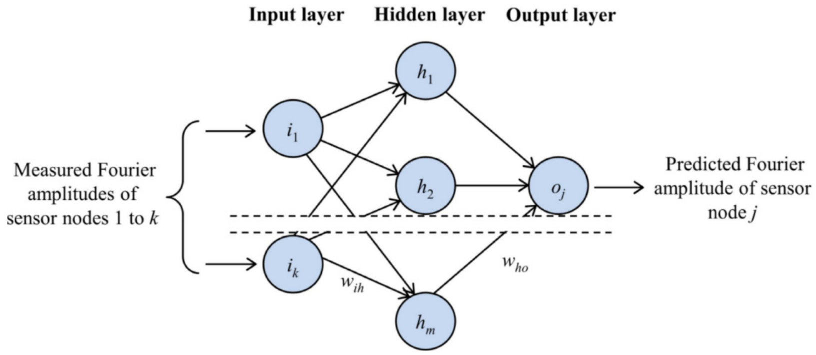
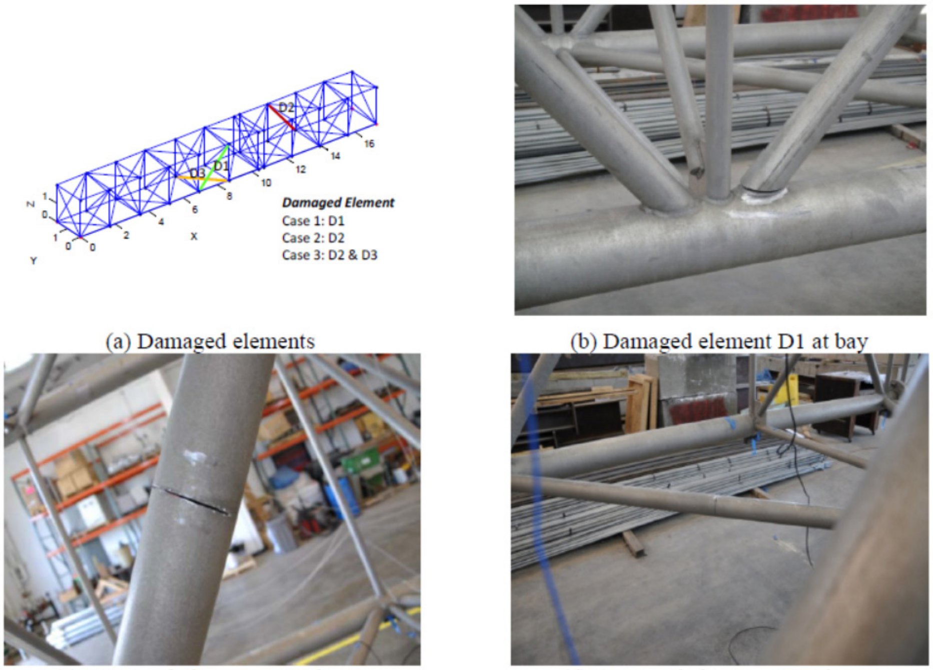

Input layer
Hidden layer
Output layer
h_1
i_1
h_2
o_j
i_k
h_m
w_{ih}
w_{ho}
Measured Fourier amplitudes of sensor nodes 1 to k
Predicted Fourier amplitude of sensor node j

Figure 2.13 Neural network model for estimating virtual sensor outputs (Dragos & Smarsly, 2016).

2.4.2 Online Condition Assessment

Condition assessment in most applications is generally a process of extracting information of interest from raw time history data for end users to make informed decisions. Conventional condition assessment is carried out off line and usually long time after data is collected. In particular, user interactions and subjectivity are involved in most conventional condition assessment strategies, which is in hence impractical for online structural condition assessment.

Various efforts have been made to realize online structural condition assessment using wireless smart sensors. For example, Gao & Spencer (2005) extended a flexibility-based damage detection technique, the damage locating vector (DLV) method (Bernal, 2002), for continuous online damage diagnosis. The key part of the online extension is to eliminate user interactions both for excitation control and data analysis. In particular, an approximate flexibility matrix for the damaged structure from ambient vibrations is constructed by utilizing the modal normalization constants from the undamaged structure. The strategy is verified for a numerical simulation of a 14-bay truss structure, but it has yet been implemented on WSSNs. Sun et al. (2015) proposed a strategy capable of online and distributed multi-step damage detection: the Angle-between-String-and-Horizon (ASH) flexibility-based algorithm as the Step I and the Axial Strain (AS) flexibility-based algorithm as the Step II. Most of the previous efforts are generally verified in either numerical simulation or small-scale lab tests with "fake" damages. In contrast, the strategy proposed by Sun et al. was implemented on Imote2 smart platforms and tested on a full-scale sign support truss structure which was previously mounted over Inter-state I-29 near Sioux City in Iowa. In particular, their strategy successfully detected damages on the structure for various cases (see Figure 2.14).



(a) Damaged elements

(b) Damaged element D1 at bay

(c) Damaged element D2 at bay 9

(d) Damaged element D3 at bay 5

Figure 2.14 Damage cases for experimental study (Sun et al. 2015).

Early damage detection relied on modal properties identifications are difficult to be implemented for real-time damage assessment, because they need a segment of data for repeated analysis for better estimation of modal properties. In addition, the damage detection results may be significantly affected by environmental and operational variabilities. To achieve robust damage detection, researchers developed several strategies based on "statistical pattern recognition" paradigm.

Sohn et al. (2001) developed a statistical time-series approach. First, a series of auto-regressive (AR) process models is built to fit the measured data; the residual errors of the AR models are sensitive to damage but affected by external inputs. So, the residual errors are then normalized by utilizing autoregressive with exogenous inputs (ARX) models, serving as damage sensitive features regardless of operational states. Finally, statistics of the residual error in the ARX models reveal the presence of damages. Lynch et al. (2004) implemented this two-tiered time-series (AR-ARX) damage detection algorithm in WSSs in an autonomous way. As shown in Figure 2.15, the coefficients of an AR model are determined in each local sensor node and transmitted to the data server. In the server, the corresponding ARX model is identified and transmitted back to the sensor node. Finally, the residual errors of the ARX model is used to determine if the damage is present. A lab test of a lumped-mass test structure is performed to validate the capabilities of the time-series algorithm, demonstrating that leveraging on-board computational capabilities for damage detection analysis mitigates the data transmission and finally save the power for WSSs. Lu et al. (2008) implemented this efficient online algorithm on wireless smart sensors and instrumented them on near full-scale single-story RC-frames for real-time damage detection during shaking table tests. To enhance the performance of the statistical methods, Yao et al. (2012) proposed new feature extraction techniques by using model spectra and residual auto-correlation, together with resampling-based threshold construction methods. Their first strategy employs Ljung-Box statistic of AR model residual sequence as damage index, and their second strategy uses Cosh spectral distance of the estimated AR model spectrum. Compared with existing algorithms based on statistical pattern recognition, the Ljung-Box statistic provides a more accurate account of the structural damage and Cosh spectral distance is less sensitive to changes in excitation sources.

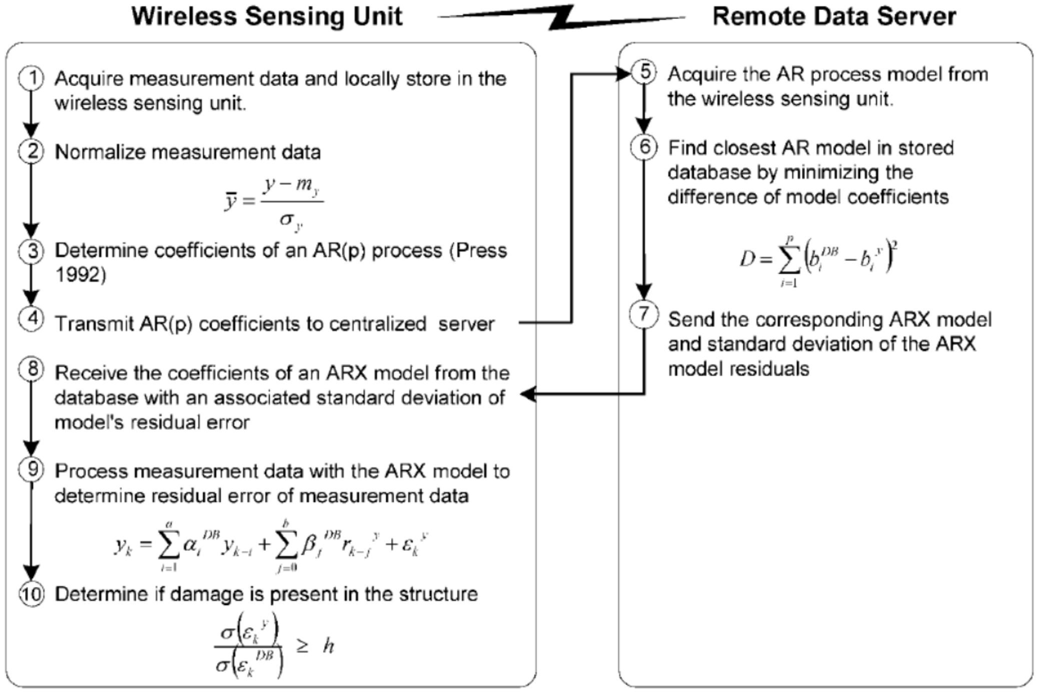

Wireless Sensing Unit
Remote Data Server
1 Acquire measurement data and locally store in the wireless sensing unit.
2 Normalize measurement data
$\bar{y} = \frac{y - m_y}{\sigma_y}$
3 Determine coefficients of an AR(p) process (Press 1992)
4 Transmit AR(p) coefficients to centralized server
5 Acquire the AR process model from the wireless sensing unit.
6 Find closest AR model in stored database by minimizing the difference of model coefficients
$D = \sum_{i=1}^{p} \left(b_i^{DB} - b_i^{y}\right)^2$
7 Send the corresponding ARX model and standard deviation of the ARX model residuals
8 Receive the coefficients of an ARX model from the database with an associated standard deviation of model's residual error
9 Process measurement data with the ARX model to determine residual error of measurement data
$y_k = \sum_{i=1}^{a} \alpha_i^{DB} y_{k-i} + \sum_{j=0}^{b} \beta_j^{DB} r_{k-j}^{y} + \varepsilon_k^{y}$
10 Determine if damage is present in the structure
$\frac{\sigma\left(\varepsilon_k^{y}\right)}{\sigma\left(\varepsilon_k^{DB}\right)} \geq h$

Figure 2.15 Flowchart of autonomous damage detection procedure (Lynch et al., 2004).

All of the aforementioned strategies are only capable for damage detection in first or second levels, namely damage presence detection and localization. To extend the level of damage detection, Hsu et al. (2011) proposed a new method by checking the change of frequency response function, which supports online damage localization and quantification. The data interrogation algorithms were embedded in WSSs to extract the frequency spectrum segments around eigenfrequencies, automatically from measured structural response. Particularly, this strategy needs structural measurements both prior and posterior to an occurrence of damage. Therefore, automatic triggering mechanism is required to justify the online damage detection capability, which is yet elaborated in the paper.

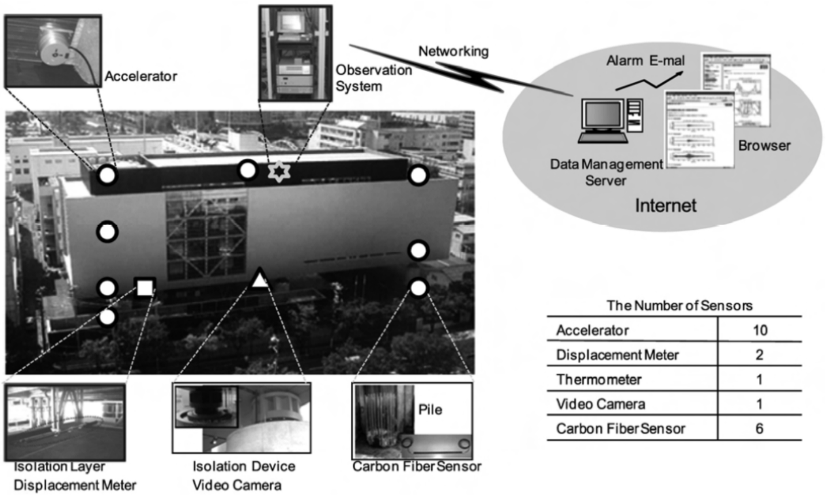
A common feature of online strategies is distributed computing, that is, embedding algorithms on the computational core of the wireless sensing units. As a result, the limited power resource is significantly saved by reducing data transmission. To fully eliminate the data transmission and synchronization for more efficient damage detection, Avci et al. (2018) implemented a 1D Convolutional Neural Network (CNN) on each sensor node of the network for real-time damage detection. In this strategy, each trained 1D CNN processes local data only, without the need of data transmission and synchronization. The efficiency of the strategy is validated in a lab test, in which damage assessment is conducted at 0.5-s intervals automatically.

Though many online condition assessment strategies are validated in either lab tests or field applications, they are merely examined on very short time scales, and very few considerations have been made to address the practical concerns (e.g. power consumption) in long-scale tests. For example, most online condition assessment strategies require continuous data acquisition to feed their algorithms with measurement data. A short-time lab test is viable, because battery power is adequate to support wireless smart sensors for always-on monitoring in minutes or even days. However, because of limited battery power supply, these algorithms are no longer practical for long-term condition assessments. In contrast, event-triggered data acquisition promises to be more efficient for online condition assessments. Though some strategies are claimed to work in line with event-triggered sensing (e.g., the algorithm developed by Hsu et al. (2011)), very little research has been focused on the development and validation of online condition assessments.

2.5 Sudden-Event Monitoring

Sudden events in structural engineering encompass natural disaster and anthropogenic hazards. Those events are unpredictable, transient, and potentially catastrophic. Yamazaki (2001) points out that “the lack of information at an early stage causes significant delays to emergency response activities.” During an earthquake, for example, once the event is detected, emergency actions during the event is possible, such as deactivating elevators and releasing dampers (Okada, et al., 2009). Also, if structural response is recorded during the earthquake, a series of important analysis can be conducted after the event, including base isolation evaluation, structural condition assessment (Celebi, et al., 2004), damage detection (Ji, et al., 2011), and improvement of future seismic design (Yamazaki, et al., 2000).

To date, wired sensors are used conventionally to monitor sudden events in various fields, such as impact detection for satellites (Hedley, et al., 2004) and space aircraft (Scheerer, et al., 2013), damage detection under strong wind for large-scale space grid structures (Qu, et al., 2006), and earthquake detection for gas pipelines (Yamazaki, et al., 1994), traffic systems (Yamazaki, et al., 2000), bridges (Celebi, 2006), and other structures (Okada, et al., 2009). In all these wired sensor systems, sensors are continuously measuring structural response, and emergency actions (e.g., email alerts and shut down structural systems) can be timely made, as shown in Figure 2.16. Although wired SHM systems can carry out continuous measurement, they suffer from significant cost of installation and extreme data inundation (Rice, et al., 2010). These two issues become more serious for sudden event detection, because a dense array of sensors should be installed to capture the location of sudden events, and continuous collection of data is required to capture the time instant of the events onset.



The Number of Sensors

Accelerator	10
Displacement Meter	2
Thermometer	1
Video Camera	1
Carbon Fiber Sensor	6

Figure 2.16. Wired monitoring and control systems for earthquake monitoring (Okada, et al., 2009).

High-fidelity wireless sensors offer tremendous opportunities to reduce costs and realize the promise of pervasive sensing for structural condition assessment. However, sudden event detection using wireless sensors remains elusive. For example, the monitoring system installed on the Golden Gate Bridge was unable to detect the three earthquakes that occurred during the three-month monitoring deployment (Cheng & Pakzad, 2009). Two main challenges to detect sudden events are apparent:

(i) Limited energy. Most wireless sensors are duty-cycled to preserve limited battery power; as a result, wireless sensors will miss the occurrence of sudden events when they are in power-saving sleep mode. Because the duty cycle is typically below 5% (Guo et al., 2011), this scenario is quite likely to occur.

(ii) Response latency. Response of WSS from sleep mode to data acquisition may take over a second, resulting in the loss of critical information in short-duration events (e.g., earthquakes and collisions). Moreover, even if awake, sensors may be busy with other tasks (e.g., data transmission); therefore, they will be unable to respond immediately to the occurrence of sudden events, and hence miss the short-duration events.

Addressing these challenges is critical to realizing a WSS for sudden event detection.

One intuitive strategy is to provide sustainable power for WSS to enable continuous monitoring of structures subjected to sudden events, emulating traditional wired monitoring systems. For example, Potenza et al. (2015) installed a wireless SHM system consisting of 17 WSSs on a historical church, which was damaged during the 2009 L'Aquila earthquake. The nodes were powered by the existing electrical lines, which guaranteed the continuity of operation and successfully detected several earthquakes over a 3-year monitoring period. Their strategy of using electrical lines to power WSS does not retain the inherent advantages of WSS, and thus may not be practical for other sudden-event monitoring applications. Energy harvesting and wireless power transfer technologies also do not provide an efficient solution. Although technologies such as solar and wind energy harvesting have been developed and validated to power WSS for periodic monitoring (Miller et al., 2010; Jang et al., 2010; Park et al., 2012), the challenge is that energy harvesting from the ambient environment is intermittent and time-varying, which is not reliable to support continuous monitoring of structures. Radio frequency (RF) energy transfer and harvesting is another wireless power technique in which WSS convert the received RF signals into electricity. The energy can be transferred reliably over a distance from a dedicated energy source to each node, or dynamically exchanged between different nodes (Lu et al., 2015a). However, the energy harvesting rate is on the order of micro-watts with low efficiency (Lu et al., 2015b) and is insufficient for high-power high-fidelity monitoring of sudden events.

On the other hand, power consumption can be reduced by employing various energy-saving mechanisms, which help to mitigate, but do not fully address the challenge of limited power for WSS. For example, Jalsan et al. (2014) proposed layout optimization strategies for wireless sensor networks to prolong the network lifetime by optimizing communication schemes without compromising information quality. Other examples of energy-saving mechanisms include data reduction, radio optimization, and energy-efficient routing. More detailed discussion can be found in Rault et al. (2014). Most of these strategies are designed to reduce power consumption for wireless transmission, which does not help energy conservation for continuous sensing, because most of the power draw comes from the sensor being always-on.

Recent developments in event-triggered sensing present both opportunities and challenges to realize sudden-event monitoring using WSS. In event-triggered sensing, the WSS only initiates measurement in response to signaling of events, which helps to save both energy and memory resources, and thus prolong the lifetime of WSS. Research has been conducted to implement low-power components (sensors and radios) that enable continuous operation and triggering mechanisms inside each sensor node. Lu et al. (2010) designed the TelosW platform, which is an upgrade of the TelosB platform (Polastre, et al., 2005), by adding ultra-low power wake-on sensors and wake-on radios. The wake-on sensor is able to wake up the microcontroller (MCU) on occurrence of events with a predetermined threshold. Additionally, the wake-on radio can wake up the MCU when a triggering radio message is received. Similarly, Sutton et al. (2016) presented a heterogeneous system architecture which included a low-power event detector circuit and low-power wake-up receivers. Although these two technologies achieve low power consumption, they do not satisfy the high-fidelity requirement of sudden-event monitoring for civil infrastructure. For example, the TelosW's analog to digital converter has only 12-bit resolution. Event-triggered sensing is also developed and implemented to facilitate railway bridge monitoring, because strain cycles and vibrations induced by trains are the most important data for bridge condition assessment (e.g., fatigue), but the arrival time of trains is generally unpredictable. Bischoff et al. (2009) deployed a wireless monitoring system which provided strain measurement and fatigue assessment of the Keraesjokk Railway Bridge. Each node was triggered independently by a low-power microelectromechanical systems (MEMS) accelerometer which operated continuously and detected an approaching train. Bias due to the transient start-up nature of the strain gage was removed by a post-processing technique. Liu et al. (2018) developed an on-demand sensing system, named ECOVIBE, to monitor train-induced bridge vibrations. In each wireless node, a passive event detection circuit, powered by vibration energy harvester, was designed to monitor bridge vibration, and another adaptive logic control circuit powered off the node once the designated tasks were finished. While effective for some applications, all the aforementioned approaches will lose critical data between the occurrence of the event and the time that data begins to be collected. Conversely, response times of wireless sensors from a cold boot to data acquisition are typically well over a second, making this problem particularly acute for short-duration sudden events (e.g., impacts can last only fractions of a second).

Moving the triggering mechanism to outside the sensor nodes provides a solution to address the challenge of data loss. In general, a separate trigger node or system is used to monitor the events continuously and notify of events to sensor nodes which are in power-saving mode most of the time. The trigger node/system is required to send notifications with a certain amount of time before the arrival of events at the structure, compensating the response latency of other sensor nodes. For example, an event-driven wireless strain monitoring system was implemented on a riveted steel railway bridge near Wila, Switzerland (Popovic et al., 2017). Two trigger nodes, referred to as sentinel nodes, were placed at 50 and 85 m away from the bridge, detecting approaching trains and sending alarm messages using a reliable flooding protocol to wake up sensor nodes on the bridge, before the train arrived. In a 47-day deployment, the system successfully detected 99.7% of train-crossing events. Likewise, to detect earthquakes and initiate seismic structural monitoring, Hung et al. (2018) developed an intelligent wireless sensor network embedded with an earthquake early warning (EEW) system which was able to detect P-waves before earthquakes arrived. In addition, each sensor node was implemented with a wake-on radio which supported ultralow-power periodic listening of wake-up commands, while the main sensor node was in deep sleep mode. Once the P-wave was detected, the gateway node, integrated with the EEW system, sent wake-up commands to sensor nodes approximately 2 seconds ahead of earthquakes. Subsequently, sensor nodes started measurement with a latency time of only 229 milliseconds. Despite successful detection of train-crossing and seismic events, the aforementioned methods do not provide a universal solution to address the challenge of data loss for many other sudden events, e.g., bridge impact by over-height trucks and ships which can hardly be detected ahead of impacts.

In addition, some progress has been made in addressing the challenge of response latency to sudden events, when WSS are awake but not in sensing mode. Cheng & Pakzad (2009) proposed a pulse-based media access control protocol. When an earthquake occurs, a trigger message with high priority is propagated from an observation site across the WSS network to preempt current tasks; sensors will be forced to conduct measurement to capture the structural response under the earthquake. Dorvash et al. (2012) developed the Sandwich node to reduce the response latency for unexpected events. A smart trigger node continuously measures the structural response; it will broadcast a proper message across a network of Sandwich nodes in the case of occurrence of events. Sandwich nodes keep listening to the trigger message; they will preempt current tasks once the trigger message is received. Response delay of Sandwich nodes is around 8 milliseconds to the occurrence of events. Although response delay is reduced in these two strategies, the wireless sensor's radio must always be on to listen for messages from a trigger node. Unless employing an ultralow-power wake-up radio, these strategies will result in a significant power draw.

2.6 Summary

This chapter presents a review of structural health monitoring using wireless smart sensors in several key aspects and associated gaps of knowledge, which are critical towards the development of wireless monitoring systems for civil infrastructure under sudden events. The use of wireless smart sensors is a cost-effective solution for sudden-event monitoring. However, the constraints in battery energy makes always-on monitoring extremely difficult, leading to incapability of sudden event detection. Time synchronization and data acquisition are two important steps for the network of wireless smart sensors, before users can collect measurements and make informed decisions. However, limited technologies are available to reduce the latencies from these two steps. Furthermore, emergency response and informed decisions highly depend on high-quality data and efficient online damage detection algorithms. However, lack of data recovery and real-time processing support in wireless smart sensors pose the limits for rapid damage assessment of structures under sudden events. The research presented in the following chapters is aimed to address the above challenges and finally develop a wireless smart monitoring system to not only provide early warning of sudden events, but also enable rapid damage assessment of structures for end users.

DEMAND-BASED WSS FOR SUDDEN EVENT DETECTION

This chapter proposes a new approach for monitoring civil infrastructure subjected to sudden events, aimed at detecting events of any duration and capturing complete transient response of any length. Particularly, a demand-based wireless smart sensor (WSS) is developed. As opposed to periodic monitoring, the *Demand-based WSS* only wakes up and initiates sensing in response to specific conditions, such as sudden events. The results of laboratory experiments and a field experiment show that the proposed approach can capture the occurrence of sudden events and provide high-fidelity data for structural condition assessment in a timely and power-efficient manner.

3.1 Demand-based WSS Development

The primary issues that must be overcome to use wireless sensors for sudden-event monitoring of civil infrastructure are: (i) the sensor must operate on battery power, (ii) high-fidelity data appropriate for SHM applications must be obtained, and (iii) the WSS node must have sufficient computational power to translate the collected data into actionable information. This section describes a *Demand-based WSS* prototype that can address these issues.

3.1.1 Ultralow-Power Trigger Accelerometer for Continuous Monitoring

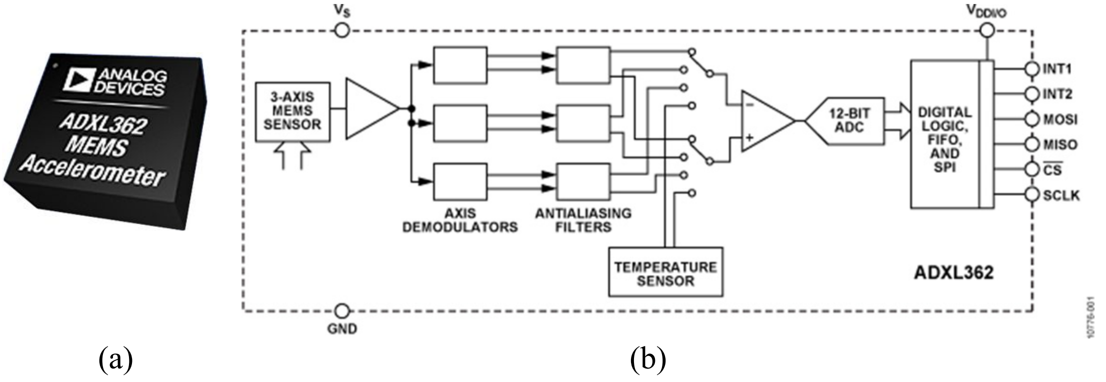
To ensure that the occurrence of sudden events is not missed, the monitoring system must be continuously in an on state. A wireless node that is always on would quickly deplete its battery. Therefore, the solution proposed herein is to use an ultralow-power trigger accelerometer that can continuously monitor the vibration of structures; the data from the accelerometer is stored in a First-In-First-Out (FIFO) buffer. When an event occurs, the data in the FIFO buffer will be frozen, and the sensor triggers an interrupt signal to wake up the main sensor platform and start sensing. Such a trigger accelerometer should have low power consumption to enable continuous monitoring for several years, good sensing characteristics, including a high sampling rate and adequate resolution, and a large FIFO buffer to ensure data is not lost after the triggering event.

Trigger accelerometers in the market today were compared and the candidates that satisfied the basic needs of sudden-event monitoring are listed in Table 3.1. The power consumption reported in the table correspond to the ultralow-noise mode of each sensor. More specifically, the ADXL362, developed by Analog Devices, consumes much less power than the other trigger accelerometers. The ADXL372, an updated high-g version of ADXL362, has a larger sampling rate and measurement range, but with a sensing resolution of only 100 mg. The LIS3DSH from STMicromechanics has high resolution of 0.06 mg, but it has a high-power draw and an inadequate FIFO buffer. Finally, the MPU6050 developed by InvenSense features a large FIFO buffer and high resolution, but it consumes substantial power.

Table 3.1 Comparison of trigger accelerometers in the market.

	ADXL362 (Analog Devices, 2016)	ADXL372 (Analog Devices, 2017)	LIS3DSH (STMicroelectronics, 2017)	MPU6050 (InvenSense, 2013)
Manufactures	Analog devices	Analog devices	STMicroelectronics	InvenSense
Supply voltage (V)	1.6–3.5	1.6–3.5	1.7–3.6	2.4–3.5
Power consumption (uA)	13	33	225	500
Sampling rate (Hz)	12.5~400	400~6400	3.125~1600	4~1000
Measurement range (g)	±2, ±4, ±8	±200	±2, ±4, ±8, ±16	±2, ±4, ±8, ±16
Resolution (mg)	1 mg	100 mg	0.06 mg	0.06 mg
Spectral noise (µg/√Hz)	175–350	5300	150	400
Buffer size (samples)	512	512	32	512

In sum, based on application needs, the ADXL362 has been selected for this study (Figure 3.1); it integrates a three-axis microelectromechanical systems (MEMS) accelerometer with a temperature sensor, an analog-to-digital converter, and a Serial Peripheral Interface (SPI) digital interface. The ADXL362 consumes only 13 µA in ultralow-noise mode at 3.3 V, which theoretically could work continuously for over two years on a single coin-cell battery. A sampling rate up to 400 Hz and a resolution of 1 mg is supported, satisfying many SHM applications. The large FIFO buffer allows the sensor to save up to 512 samples, which corresponds to 1.7 s for all three axes sampled at 100 Hz. Moreover, it has built-in logic for acceleration threshold detection; a detected event can be used as a trigger to wake up the primary sensor node.



(a) (b)

Figure 3.1 ADXL362: (a) sensor chip; (b) Functional block diagram (Analog Devices, 2016).

3.1.2 High-Fidelity Sensor Platform for Sudden-Event Monitoring

To provide high-quality measurement data and enable rapid condition assessment of structures subjected to sudden events, the sensor platform should have following features: (i) sensors and a data acquisition system that can obtain high-quality data at a high sampling rate for the event; (ii) powerful microcontroller to acquire and analyze sensor data in near real time. Other important features include: reliable communication, open-source software, and efficient data and power management.

A summary of the most advanced wireless sensor platforms available in the market currently is given in Table 2.1. The Xnode, developed by Embedor Technology, has a 24-bit Analog-to-Digital Converter (ADC), which is the best in its class. The microprocessor unit (MCU) of Waspmote (Libelium, Zaragoza, Spain) is not able to support rapid processing of large amount of data. The MCU information for the AX-3D (BeanAir, Berlin, Germany) and the G-Link-200 (LORD Sensing, Williston, VT, USA) is unavailable, and the operating systems of these platforms are proprietary. Note that several high-performance wireless sensor platforms (e.g., Imote2) are no longer commercially available, and hence not compared herein.

Because of its high sensing resolution, high sampling rate, powerful microprocessor, and open-source software, the Xnode Smart Sensor (Spencer et al., 2016) has been selected as the host wireless sensor platform in this study. The standard Xnode consists of three modular printed circuit boards (PCB): (i) the processor board, (ii) the radio/power board, and (iii) the sensor board (Figure 2.5). In particular, it employs an 8-channel, 24-bit ADC (Texas Instruments ADS131E8), allowing a maximum sampling rate up to 16 kHz and an NXP LPC4357 microcontroller operating at frequencies up to 204 MHz, which can be used to execute data-intensive on-board computation. Moreover, it implements open-source middleware services (Fu et al., 2016), which facilitates custom application development. In addition, it possesses two SPI controllers, making it possible to communicate with the selected trigger accelerometer, the ADXL362.

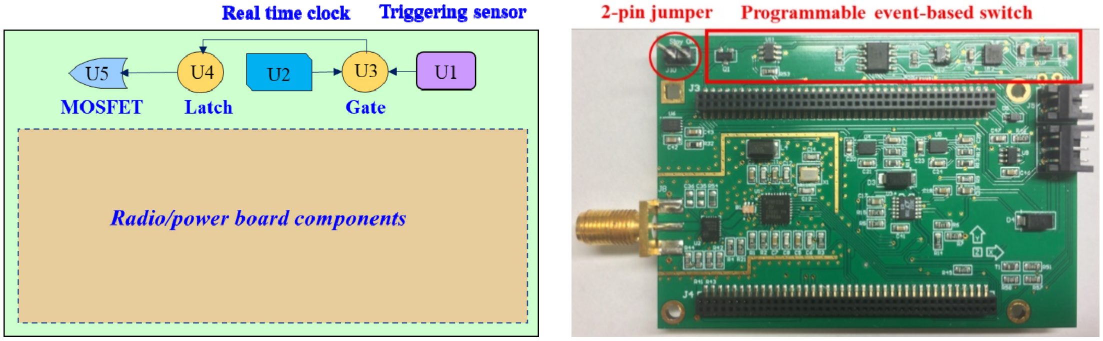
3.1.3 Integration of Trigger Sensor and High-Fidelity Sensor Platform

To capture the entire event without loss of critical information, the ADXL362 accelerometer and the Xnode must be carefully integrated to build a *Demand-based WSS*. The first-generation integration can be found in Fu et al. (2018), which realizes basic functionalities of event-triggered sensing but consumes more current draw of 10.2 mA than expected. Afterwards, the second-generation integration is developed and discussed in the remainder of this section, in terms of hardware and software.

(1) Hardware consideration

To address the challenge of physical integration of the ADXL362 accelerometer into the Xnode, a programmable event-based switch was designed and implemented on the radio/power board of the Xnode in the *Demand-based WSS*. When a sudden event occurs, and the vibration exceeds a user-defined threshold, an interrupt pin in the ADXL362 generates a triggering signal. This signal is connected to a MOSFET to flip its state, turning on the Xnode and initiating high-fidelity sensing. When the event ends (lack of acceleration above a threshold), the other interrupt pin in the ADXL362 generates a signal to notify the Xnode to stop high-fidelity sensing. After data acquisition is completed, the triggering signal is cleared, and the MOSFET turns off the Xnode. The communication of control messages between the ADXL362 and the Xnode is carried out via SPI bus through a four-wire connection. In addition to enable event-triggered sensing, the proposed switch should be designed to retain traditional functionality for periodic monitoring. Specifically, sensor nodes are operated on low duty cycles, and the base station can access the network of nodes at random to initiate operations or measurements in the network. To achieve this goal, a real-time clock, DS3231, is employed in the proposed switch. When a user-defined period passes or at a specific time of day, the DS3231 sends a triggering signal which flips the MOSFET switch and turn on the Xnode. Then the node remains awake for a short period of time to listen for messages from the base station. Once a command is obtained, the node carries out the required task (e.g., sensing, battery check). The communication of control messages between the DS3231 and the Xnode is conducted through the I2C bus. In addition, the DS3231 can help to record the onset time of a sudden event, in the form of 24-hour clock and Gregorian calendar.

Figure 3.2a illustrates the design concept of the proposed switch. Five major components are implemented, including a trigger sensor ADXL362 (U1), a real-time clock DS3231 (U2), an AND gate (U3), a latch (U4), and a MOSFET (U5). Interrupt pins from the ADXL362 and the DS3231 are connected to the MOSFET through the AND gate. This circuit enables the MOSFET to be triggered by either the real-time clock or the trigger sensor. In addition, a latch component is added between the AND gate and the MOSFET, to keep the power supply stable. Figure 3.2b shows the realized PCB for the proposed switch. The five major components, as well as companion resistors and capacitors, are all soldered on the edge of top side. In some use cases such as downloading code to the board and debugging, the sensor platform should always be on and therefore the designed switch needs to be bypassed. To achieve this goal, a 2-pin jumper is added. When the two pins on the jumper are not connected, the proposed switch works as designed, otherwise, it is bypassed.



(a) (b)

Figure 3.2 PCB design for the event-based switch: (a) design concept (b) realized PCB.

(2) Software consideration

In addition to hardware development of the prototype, an effective application framework is required to control the behavior of *Demand-based WSS* to realize event-triggered sensing. Figure 3.3 shows a flowchart of the application framework for the *Demand-based WSS*. More specifically, when users turn on the physical switch of a main sensor platform, the Xnode first initializes itself and sends commands which contain configuration parameters (e.g., threshold, timers, and data buffer size) to the event-based switch discussed in the previous section. Once the commands have been received, the switch completes configuration of the device settings. Subsequently, the ADXL362 starts measurement in ultralow-noise mode, and the Xnode is turned off. If a sudden event occurs and the acceleration obtained in the ADXL362 exceeds the user-defined threshold, an interrupt pin, INT1 on the ADXL362 sends a trigger signal to turn on the Xnode. Concurrently, the ADXL362 saves 512 data samples into its FIFO buffer surrounding the onset of the event and waits for the Xnode to retrieve the data. The Xnode starts high-fidelity data acquisition using its built-in high-power high-accuracy MEMS accelerometer. When the event stops and the acceleration obtained in the ADXL362 is lower than a user-defined threshold for a certain period of time, the other interrupt pin, INT2, in the ADXL362 is triggered. Subsequently, the Xnode stops high-fidelity sensing. After sensing is completed, the Xnode reads data from the FIFO buffer of the ADXL362 and fuses it with the Xnode data. In addition, when the Xnode is busy with other tasks (e.g., data transmission of a previous event), but another sudden event occurs, the INT2 pin can be configured to interrupt undergoing tasks and force the Xnode to start high-fidelity sensing immediately. In addition, timing analysis results for each stage of a *Demand-based WSS* are presented in the left of Figure 3.3.

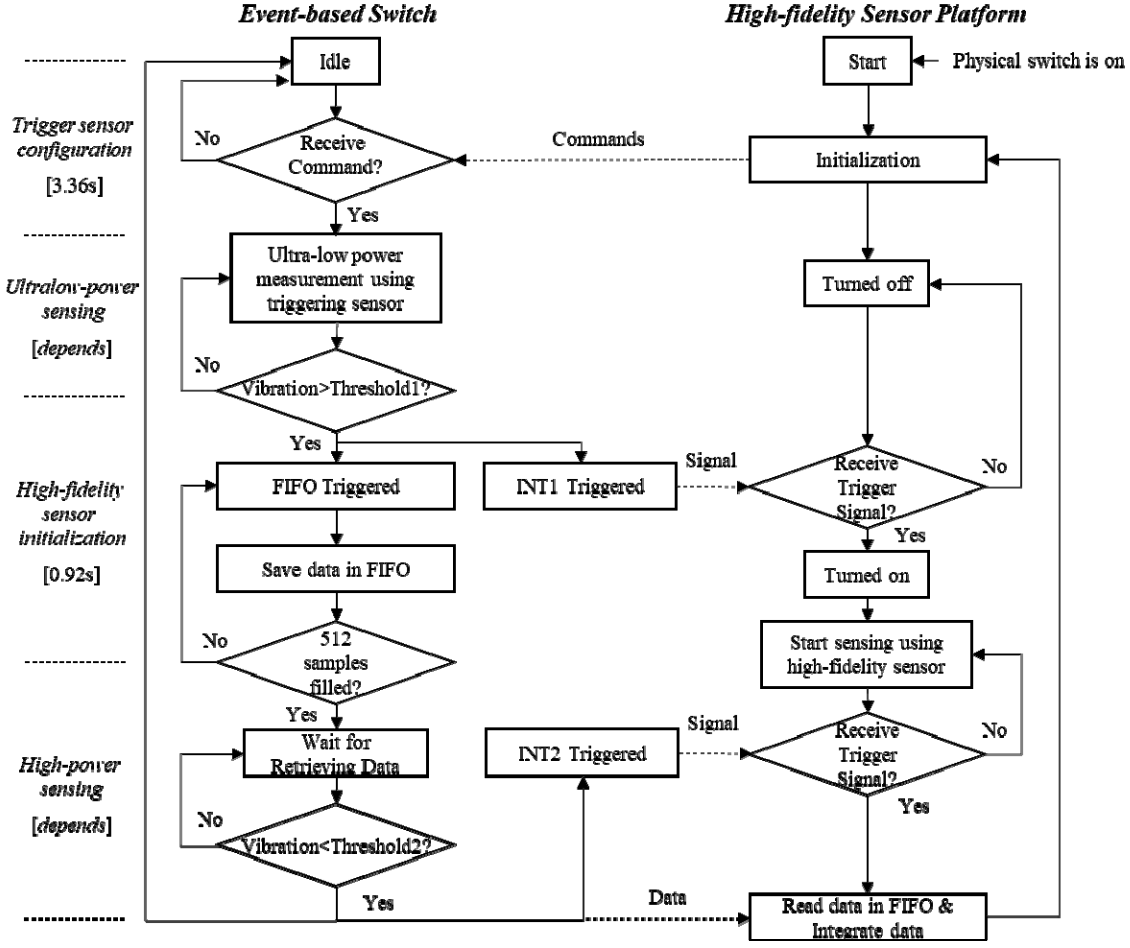


Figure 3.3 Flowchart of *Demand-based WSS* for event-triggered sensing.

For sudden events that are rare, e.g., earthquakes, the thresholds can be determined based on priori information about the sudden events that are monitored. The priori information can be estimated by numerical analysis, the data in the history, or the measurement data in a preliminary test. For some events that occur frequently, e.g., railway bridge impacts from over-height vehicles, the thresholds can be determined adaptively, starting from a relatively low value during a "training phase" and then adjusted until the detection errors are minimized.

3.2 Data Fusion to Address Data Loss

The objective of the *Demand-based WSS* is to obtain the data from before the trigger event occurs until the structural accelerations stop. Specifically, the ADXL362 can record structural measurements surrounding the onset of a sudden event, whilst the Xnode starts sampling the data approximately 0.9 s after being triggered. Therefore, the ADXL362 data and the Xnode data must be synchronized and fused to produce a complete representation of the acceleration record. The following paragraphs describe the challenges encountered in this process, along with the associated resolutions.

To fuse the two overlapping data streams, two main challenges should be addressed, including (i) differences in the sampling rate between the ADXL362 and the Xnode, and (ii) synchronization error between the ADXL362 data and the Xnode data. More precisely, the first challenge results from the differences between the clock rates of the ADXL362 and the Xnode. The internal clock rate in the ADXL362 has a standard deviation of approximately 3%. One approach might be to calibrate the ADXL362 incorporated in each Xnode; however, this approach is not practical, as the clock rate will change with temperature, invalidating the initial calibration. The second challenge is due to the variance in start-up time of the sensing task on the Xnode. As a result, a random offset will exist between the two data streams.

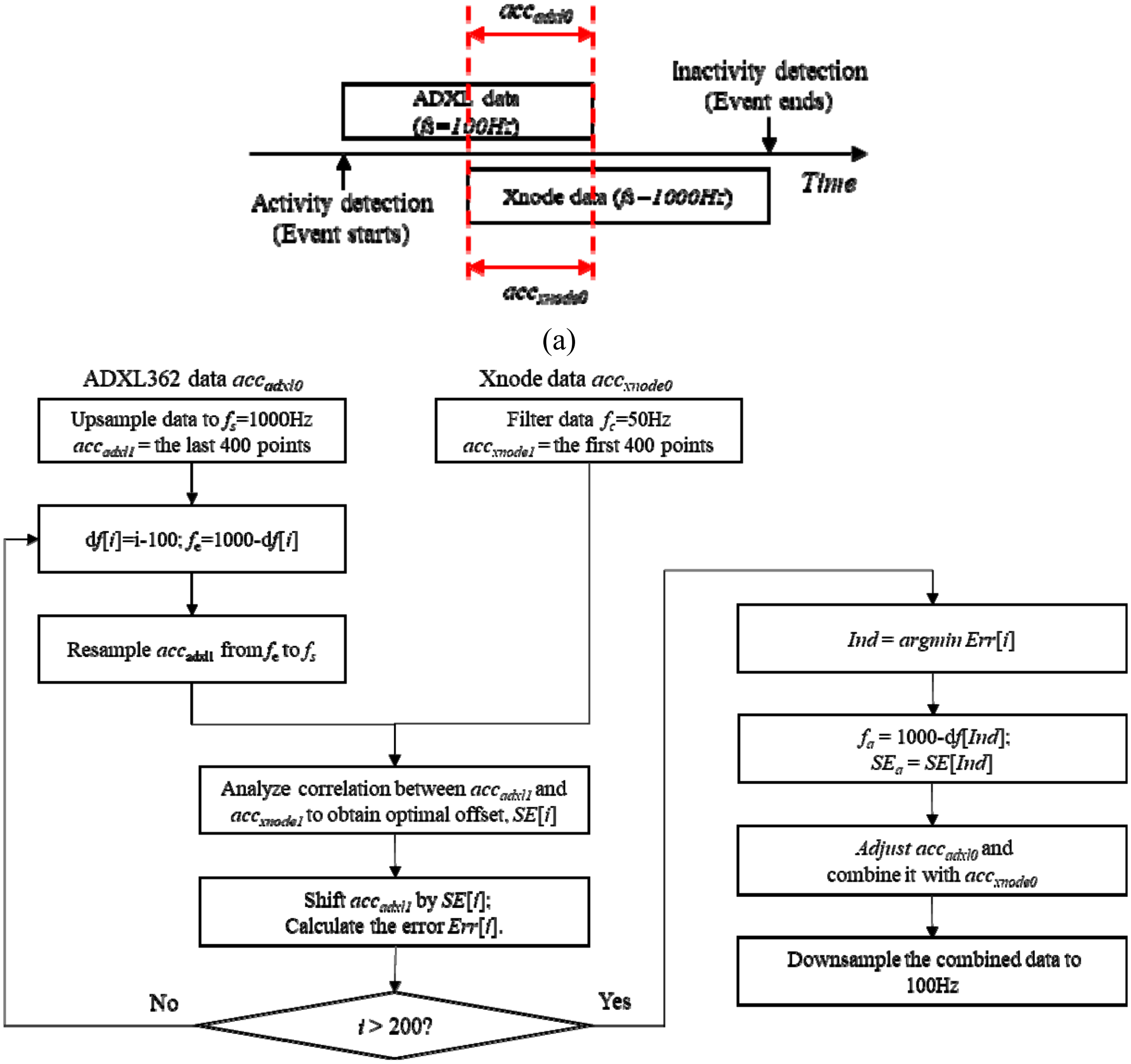
To tackle the challenges identified in the previous paragraph, the beginning of the Xnode data, which is overlapped with the ADXL362 data, was utilized to calibrate the entire ADXL362 data stream. Figure 3.4 shows a flowchart of this approach. More specifically, the ADXL362, acc_{adxl0}, with a nominal sampling rate of 100 Hz, were first up-sampled to f_s (1000 Hz). The last 400 data points of the acc_{adxl0} were chosen as acc_{adxl1}, which was assumed to be approximately overlapped with the beginning of the Xnode data. In the meantime, the Xnode data, acc_{xnode0}, was sent through an 8-pole elliptic low-pass filter with a cutoff frequency of 50 Hz, to have the same bandwidth with acc_{adxl0}. The first 400 data points of acc_{xnode0} were considered as acc_{xnode1}. Based on the datasheet of ADXL362 (Analog Devices, 2016), the clock frequency deviation from the ideal value was within the range of −10% and 10%. Therefore, to find the actual sampling frequency of acc_{adxl1}, exhaustive search was applied from 900 Hz to 1100 Hz. For Step *i*, the estimated sampling frequency (f_e) of the ADXL362 data was set as,

$$f_e = 1000 - df[i] \tag{3.1}$$

where, $df[i] = i - 100,\ i \in [0, 200]$. acc_{adxl1} was resampled from f_e to f_s using resampling-based approach (Nagayama & Spencer, 2007). Then, to estimate the synchronization error between acc_{adxl1} and acc_{xnode1}, the cross-correlation between the two data segments was calculated. The optimal offset, $SE[i]$, was obtained, for which the cross-correlation reaches its maximum value. Afterwards, acc_{adxl1} was shifted by $SE[i]$, and then the data fusion error (Err) was calculated as,

$$Err[i] = \left\| acc_{adxl1} - acc_{xnode1} \right\|_2 \tag{3.2}$$

where, $\| \; \|_2$ represents Euclidean norm. After completing these steps, the best estimations of sampling frequency f_a and synchronization error SE_a were obtained in the step that achieves minimal Err. Subsequently, f_a and SE_a were applied to calibrate the original data set, acc_{adxl0}. Finally, acc_{adxl0} and acc_{xnode0} were combined and down-sampled to 100 Hz for ensuing analysis.


acc_adxl0
Inactivity detection
(Event ends)
ADXL data
(fs=100Hz)
Time
Activity detection
(Event starts)
Xnode data (fs=1000Hz)
acc_xnode0
ADXL362 data acc_adxl0
Xnode data acc_xnode0
Upsample data to fs=1000Hz
acc_adxl1 = the last 400 points
Filter data fc=50Hz
acc_xnode1 = the first 400 points
df[i]=i-100; fe=1000-df[i]
Resample acc_adxl1 from fe to fs
Analyze correlation between acc_adxl1 and acc_xnode1 to obtain optimal offset, SE[i]
Shift acc_adxl1 by SE[i];
Calculate the error Err[i].
i > 200?
No
Yes
Ind = argmin Err[i]
fa = 1000-df[Ind];
SEa = SE[Ind]
Adjust acc_adxl0 and combine it with acc_xnode0
Downsample the combined data to 100Hz

(a)

(b)

Figure 3.4 Post-sensing data fusion: (a) illustration of two data sources, (b) flowchart of data fusion strategy.

3.3 Experimental Validation

To validate the performance of the *Demand-based WSS*, laboratory tests were carried out for data fusion, earthquake monitoring, and impact detection. The detailed test setup and results are presented in this section. The performance of the *Demand-based WSS* is discussed, in terms of power consumption, sensing characteristics, and data quality for sudden-event monitoring.

3.3.1 Validation of Data Fusion

A lab test was conducted to illustrate the challenge of data fusion between the ADXL362 data and the Xnode data. Specifically, a *Demand-based WSS* was located at 10th floor of an 18-story building model shown in Figure 3.5. The ADXL362 was configured to capture samples at 100 Hz, starting at 0.2 s before the triggered event and continuing until 1.5 s after the event. The event-triggering threshold was set to 150 mg, at which time, the Xnode was turned on and 1000 Hz high-fidelity measurement was started. To reduce false positives, two consecutive data points exceeding the threshold were required to cause triggering. In addition, a wired piezoelectric accelerometer, model PCB353B33 (PCB Piezotronics, Inc., Depew, NY, USA), was installed on the same floor and sampled at a frequency of 128 Hz. The acceleration from these sensors served as reference data. A sudden event was simulated by manually shaking the building model in horizontal direction.

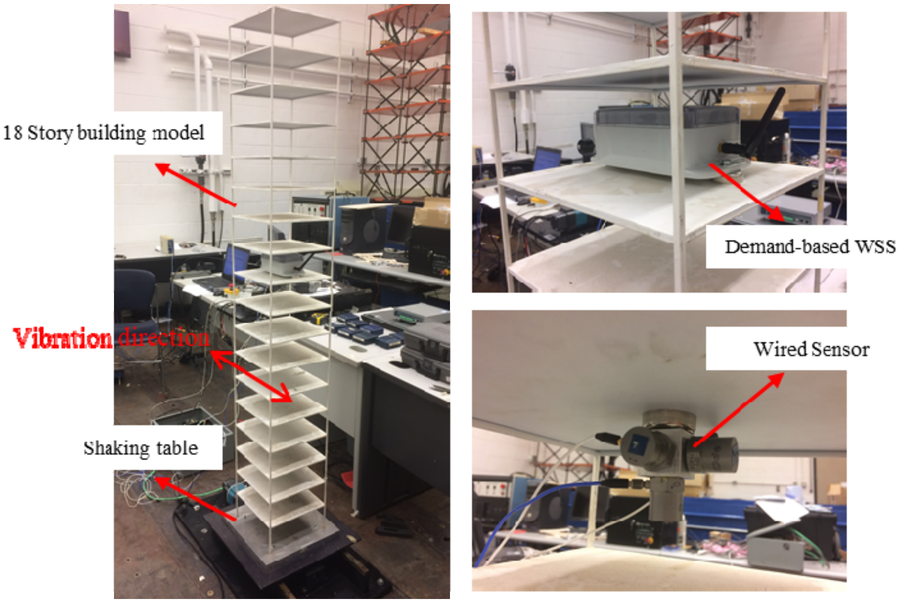
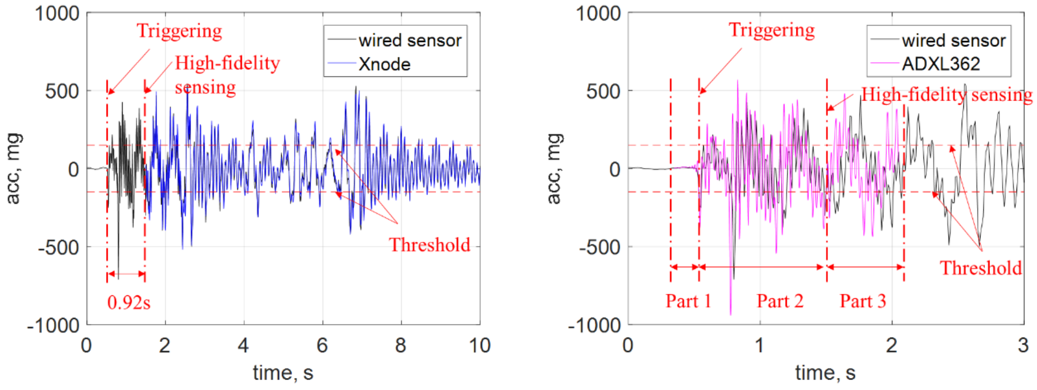


Figure 3.5 Experiment setup for data fusion test of a *Demand-based WSS.*

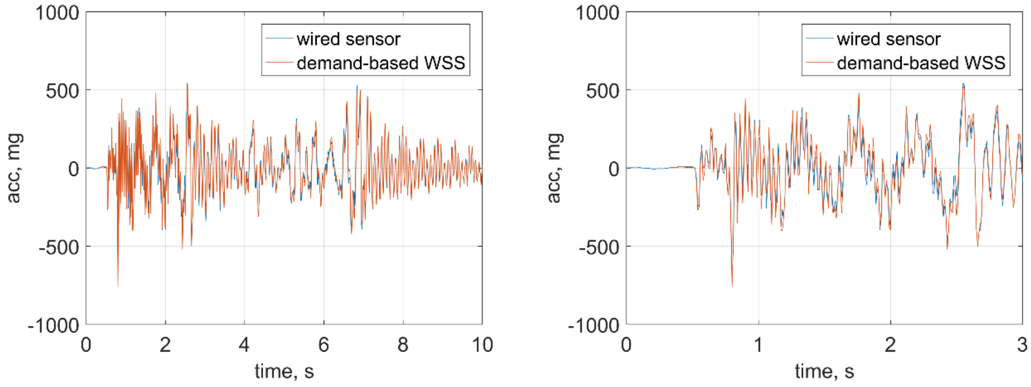
To make a direct comparison in the time domain, the Xnode data and wired sensor data were sent through an 8-pole elliptic low-pass filter with a cutoff frequency of 50 Hz, as displayed in Figure 3.6. The direction of acceleration measurement was the same with the vibration direction specified in Figure 3.5. The vibration exceeded the threshold at 0.7 s, and the event-based switch turned on the Xnode. The Xnode required 0.92 s for initialization. Fortunately, as shown in Figure 3.6b, acceleration data stored in the FIFO buffer of the ADXL362 was recorded during this period. Specifically, the ADXL362 data can be divided into three parts: (i) Part 1 is the pre-triggered data which is around 0.28 s in length; (ii) Part 2 is the data that cover the time where the Xnode is initializing; (iii) Part 3 is where the ADXL362 data overlaps with the Xnode data. The length of the data in Part 3 is approximately 0.6 s. As shown in Figure 3.6b, the data obtained from the ADXL362 does not match well with the reference data from the wired sensors, because the sampling frequency of the ADXL362 was slightly smaller than 100 Hz, which illustrates the first challenge mentioned in the Section 3.2. In addition, the time offset between the Xnode data (blue line in Figure 3.6a) and the ADXL362 data (red line in Figure 3.6b) must be accurately estimated to fuse these two data streams, which illustrates the second challenge of data fusion.

The data fusion strategy discussed in the previous section is applied to the test data. Figure 3.7 shows a comparison of time history data between sensor data from wired sensors and the fused data from the *Demand-based WSS*. The excellent agreement demonstrates the ability of the proposed strategy to seamlessly capture the structural response subjected to a sudden event.



(a) (b)

Figure 3.6 Time history data comparison: (a) Xnode measurement, (b) ADXL362 data buffer.



(a) (b)

Figure 3.7 Results of data fusion: (a) time history data, (b) zoomed view (ADXL362 data).

3.3.2 Earthquake Monitoring

As a typical sudden event, an earthquake is transient and unpredictable, and the consequences can be catastrophic. Continuous efforts are required to develop cost-effective earthquake monitoring systems to mitigate the effect of earthquakes. *Demand-based WSSs* have a significant potential to enable earthquake detection and rapid damage assessment of civil infrastructure, which was validated through a lab test in this subsection.

The test setup was the same with that in Section 3.3.1, as shown in Figure 3.5. The structure model was mounted on a uniaxial shaking table. This shaking table can simulate earthquakes in one horizontal direction, driving a 15 kg mass at 2.5 g with a maximum stroke of ±7.5 cm. The El Centro earthquake excitation was generated by the shaking table to represent a sudden event. The detection threshold for the event-based switch in the *Demand-based WSS* was configured as follows: the onset of event was detected when the acceleration was above 80 mg over 0.02 s, and the end of event was detected when the acceleration was below 40 mg over 5 s. Other configuration parameters are the same with the test in Section 3.1, such as sampling frequencies and filter parameters.

A segment of 90 s recorded time history is shown in Figure 3.8a. As can be seen in the zoomed view of the time history data (Figure 3.8b-c), the ground motion started at 10.7 s, but the vibration in the beginning was very small. From 11.79 s to 11.80 s, two consecutive acceleration samples obtained by the trigger accelerometer exceeded 80 mg. As a result, the event-based switch turned on the *Demand-based WSS* immediately and the WSS started high-fidelity measurement. The acceleration became smaller than 40 mg after 54.50 s. Approximately 5 s later, the event-based switch stopped the high-fidelity sensing. Furthermore, Figure 3.8d shows the power spectral density (PSD) in the frequency domain. The excellent agreement between the results of wired sensors and the *Demand-based WSS* in the both time and frequency domain demonstrates the ability of the proposed WSS to detect the earthquake and capture the accurate structural response during earthquakes in a timely and efficient manner.

3.3.3 Impact Detection

Another typical sudden event is impact detection, especially for bridges. Over 500 bridges in the railway and highway systems failed between 1989 and 2000, based on one report from the Federal Highway Administration in 2001. Lateral impact from the trucks/ships is considered as one of the most frequent causes of those bridge failures. Therefore, the development of monitoring systems is essential to detect the impact events for early warning and obtain the structural response for subsequent condition assessment, such as the identification of impact location and the estimation of structural damages. The result of impact monitoring provides the basis for the engineers to make informed decisions (e.g. bridge closure for repairing). *Demand-based WSSs* can help to enable impact detection and rapid damage assessment of civil infrastructure, which is validated through a lab test.

The testbed structure is the same with that in Section 3.3.1. In the lab test of impact detection, a hammer is used to excite the impact force on one of the floors. In an impact event, the amplitude of structural response can be very high. Accordingly, the measurement range of the trigger accelerometer is configured to be the maximum value, 8g. Also, structures oscillate rapidly at first several seconds of impact events. In this scenario, if the trigger accelerometer is still configured to measure at 100Hz, the possibility that two consecutive samples exceed the threshold is relatively small. Therefore, to detect the onset of the transient impact response, the threshold criteria for impact detection is when acceleration remained above 80mg only for 1 data sample which corresponds to 0.01s. Though it may introduce false-positive scenarios, the possibility of false-negative cases will be reduced. The end of event was detected when acceleration remained below 20mg for 5s. In addition, the sampling frequency of wired sensors is set to be 4096Hz to measure the structural transient response precisely. All the other configuration parameters are kept the same with the test in Section 3.3.2.

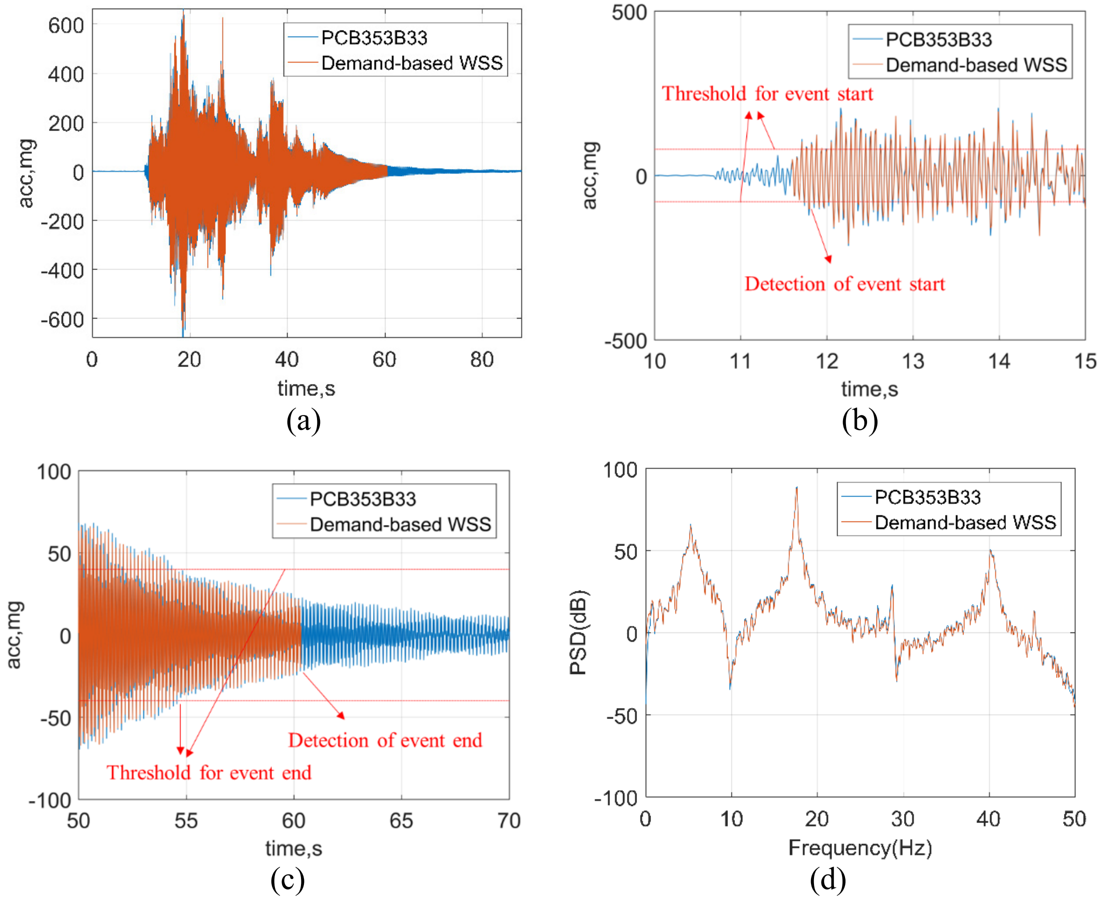
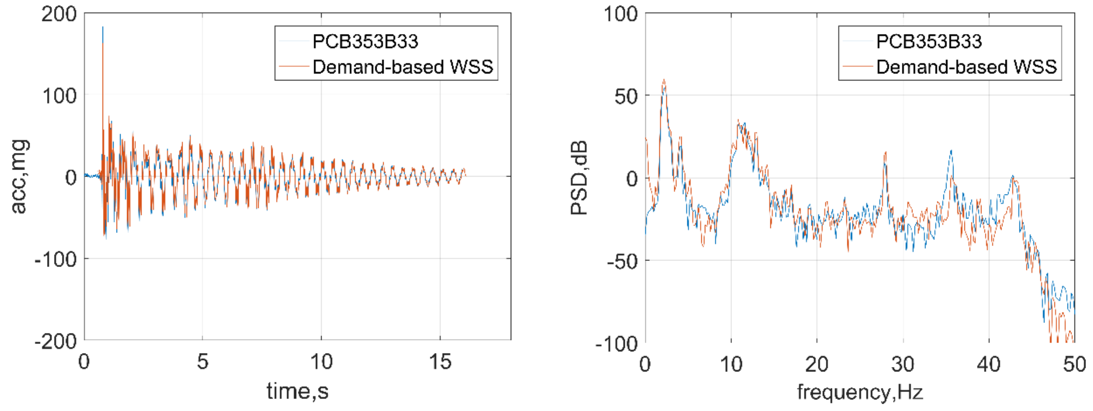


Figure 3.8 Test results of earthquake monitoring: (a) time history data, (b) zoomed view of time history data for event start, (c) zoomed view of time history data for event end, (d) power spectral density (PSD) data.

A segment of 16s recorded time history is obtained. To make a direct comparison in the time domain, the data from both wired sensors and the *Demand-based WSS* was sent through an 8-pole elliptic low-pass filter with a cutoff frequency of 50 Hz and is displayed in Figure 3.9a. The impact occurs at 0.79s, and the vibration exceeds the threshold of 80mg. Accordingly, the *Demand-based WSS* is immediately turned on to start high-fidelity measurement. In addition, at 11.12s, the acceleration is smaller than 20mg, and after around 5s, the event-based switch stops the high-fidelity sensing. Figure 3.9b shows the PSD data in the frequency domain. The good agreement between the results of wired sensors and the *Demand-based WSS* in both time and frequency domain demonstrates the ability of the proposed WSS to detect the impact event and capture the accurate structural response during the impact event in a timely and efficient manner.



(a) (b)

Figure 3.9 Test results of impact detection: (a) time history data, (b) PSD data.

3.3.4 Evaluation and Discussion

In the lab tests described in previous sections, the attractive performance of the *Demand-based WSS* was successfully validated to detect sudden events and provide high-quality sensing data for SHM analysis.

Power consumption tests showed that the proposed WSS has a current draw of only 365 μA when no sudden event occurred, but the power consumption of the original Xnode sensor platform is approximately 170 mA during sensing. Considering that sudden events are rare and short-duration, most of the time the *Demand-based WSS* deployed on a structure is in low-power measurement mode. Therefore, if using a 3.7 V DC, 10,000 mAh, rechargeable lithium polymer battery, employing the proposed WSS can extend the lifetime of always-on monitoring from three days to over three years using a single lithium battery. This feature helps to successfully detect the occurrence of sudden events with minimal power budget in long-term monitoring. In addition, the current draw in each operation associated with duration for a *Demand-based WSS* was determined experimentally and shown in Figure 3.10, in which the majority component of power consumption is sensing.

The data obtained from the *Demand-based WSS* is high-quality, matching well with the data from wired piezoelectric accelerometers. In particular, high-fidelity sensing enables 24-bit sensing resolution and over 1 kHz sampling rates. This feature helps to conduct structural condition assessments accurately under sudden events.

The test results show that, when an event occurs, a seamless transition from the low-power sensing to high-fidelity measurement is carried out, without losing any data about the event.

These three features demonstrate that the proposed WSS satisfies the demands of sudden-event monitoring.

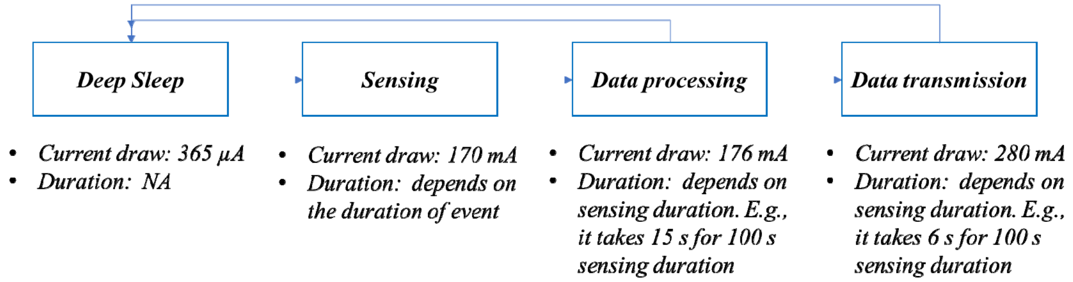

Deep Sleep
Sensing
Data processing
Data transmission
Current draw: 365 µA
Duration: NA
Current draw: 170 mA
Duration: depends on the duration of event
Current draw: 176 mA
Duration: depends on sensing duration. E.g., it takes 15 s for 100 s sensing duration
Current draw: 280 mA
Duration: depends on sensing duration. E.g., it takes 6 s for 100 s sensing duration

Figure 3.10 Event-triggered sensing regarding current draw and duration for each operation.

3.4 Train-crossing Event Monitoring

To further validate system performance, a field test was conducted on a steel railroad bridge north of Champaign, Illinois. Having vibration data while in-service trains traverse the bridge is useful to assess the bridge condition (Moreu et al., 2017). Train-crossing events have similar features to sudden natural events, e.g., unpredictability due to uncertain train schedules, but occur more frequently and therefore provide a convenient test platform. A *Demand-based WSS* was deployed on the bottom side of a bridge girder. Simultaneously, wired sensors, model PCB353B33, were selected as reference sensors and deployed close to the WSS (see Figure 3.11). A detection threshold was configured to be the same as the test in Section 3.3. To avoid signal saturation, the measurement range of the trigger accelerometer was set to the maximum value of 8 g. At 10:52:06 a.m. on 7 May 2019, an Amtrak passenger train passed by the bridge.

Figure 3.12a-d shows the raw acceleration data of the bridge in vertical direction. The train came to the bridge at 121 s and left at 128 s. The event had a short duration of 7 s, and it was successfully detected by the *Demand-based WSS*. Figure 3.12e shows the PSD data. Some slight discrepancies between the data from two sensors are possibly due to the different locations of the sensors. In sum, good agreement can be observed between the wired sensor and the *Demand-based WSS* both in the time and frequency domain, demonstrating that the new WSS can capture the sudden event and obtain high-fidelity measurement in real applications.

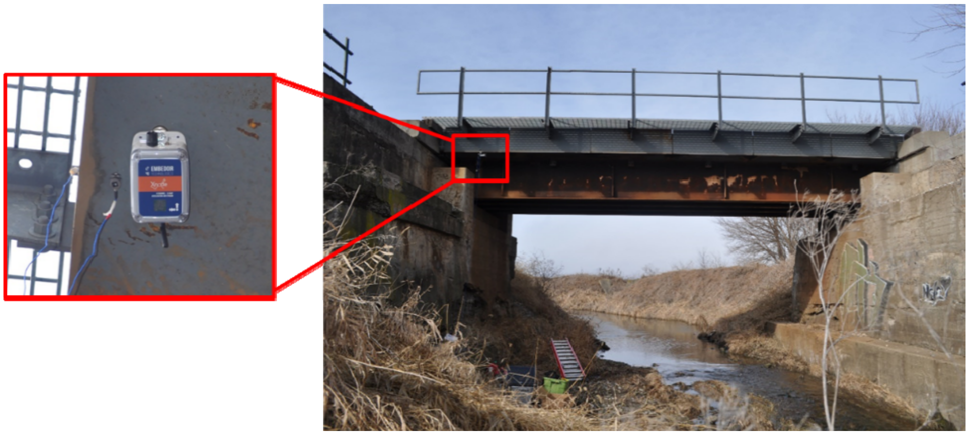


Figure 3.11 Field application of the *Demand-based WSS.*

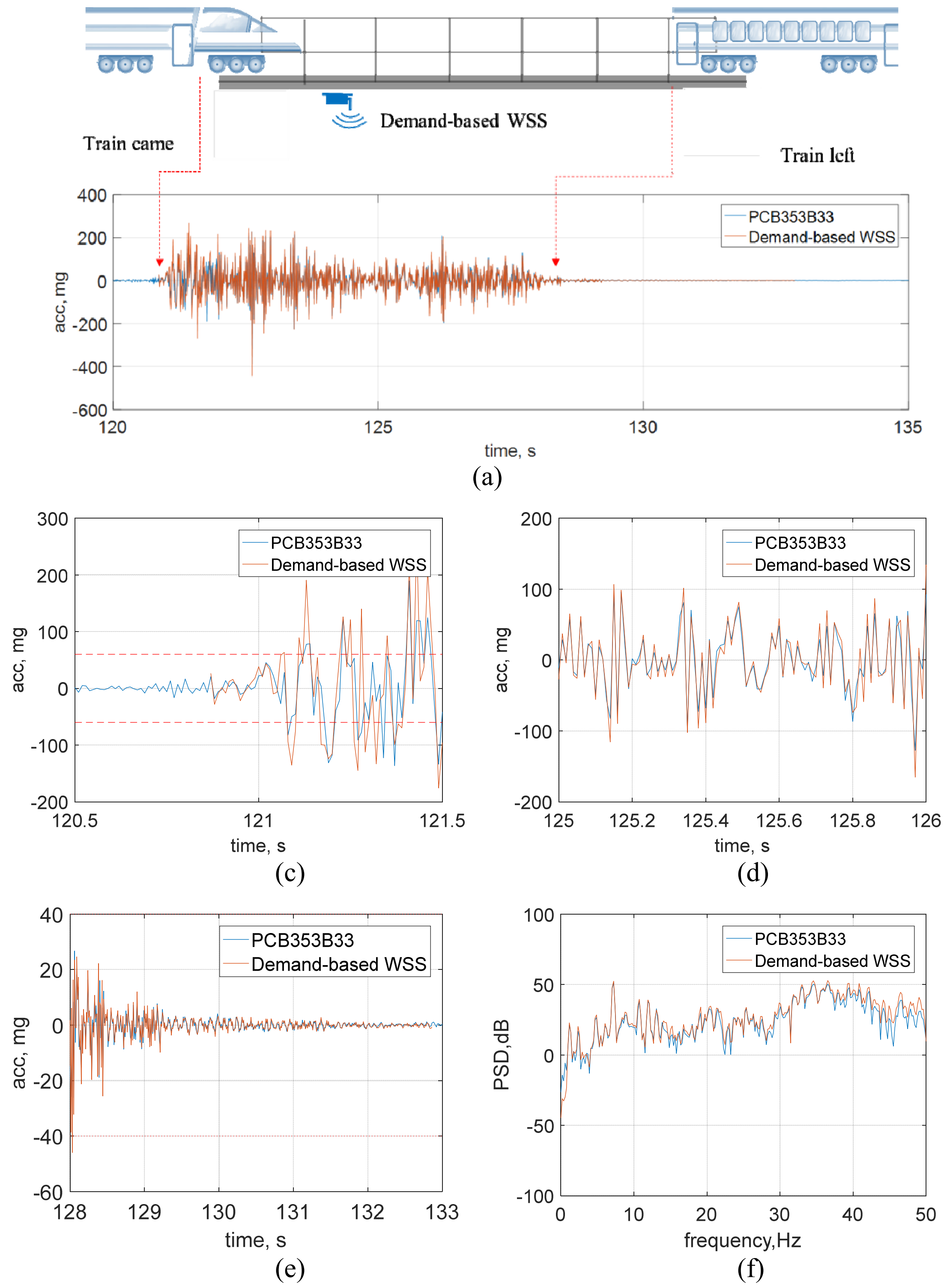


Figure 3.12 Test results on a railroad bridge: (a) time history data, (b) zoomed view of event starts (ADXL362 data), (c) zoomed view of event data (Xnode data), (d) zoomed view of event ends (Xnode data), (e) PSD data.

3.5 Summary

This section presented the design of *Demand-based WSS* prototypes to meet the application requirements of sudden-event monitoring. The proposed WSS mainly consists of a unique programmable event-based switch and a powerful high-fidelity WSS platform. In particular, the event-based switch is built on a trigger accelerometer which allows the new WSS to measure the structural response in ultralow-power in long term, so as not to miss sudden events. In addition, the software of event-triggered sensing and data fusion is implemented. The performance of the proposed WSS is evaluated through lab tests and a field application on a railroad bridge. The test results show that the proposed WSS can continuously monitor structural response with minimal power budget, and hence detect the occurrence of the sudden event with the smallest delay. Besides detecting sudden events, the proposed WSS have the excellent features of high sampling rates and sensing resolution, which finally helps to provide high-quality data in sudden events for rapid condition assessment of civil infrastructure. Moreover, the proposed WSS are powerful and versatile not only for sudden events (e.g., earthquakes), but also for autonomous monitoring of other general events (e.g., bridge/highway overloads).

For large-scale structures, one *Demand-based WSS* is not sufficient, and a network of nodes are needed for a meaningful characterization of the structural response. When subjected to a sudden event, each node may be triggered independently to initiate measurement at slightly different times due to varying response levels in the structure. Next chapter will address the challenges encountered for a network of *Demand-based WSSs* under sudden events. For example, one critical issue is to synchronize data from different sensor nodes without introducing delay of event-triggered sensing.

Chapter 4

TIME SYNCHRONIZATION FOR SUDDEN-EVENT MONITORING

This chapter first describes three unique challenges of time synchronization in sudden-event monitoring. To address these challenges and other issues of time synchronization for traditional SHM, the Xnode's clock behavior is first examined, and an efficient two-stage time synchronization strategy for traditional SHM is implemented as a baseline. Event-triggered synchronization in sudden-event monitoring is classified into three categories that call for different strategies: offline synchronization strategies for short- and long-duration monitoring and an online synchronization strategy for rapid data analysis in long-duration monitoring. These strategies are experimentally validated to provide efficient and accurate (<20μs maximum error) synchronized sensing for sudden-event monitoring.

4.1 Time Synch Requirements for Sudden-Event Monitoring

In contrast to periodic monitoring and campaign-type monitoring using WSSNs, sudden-event monitoring requires event-triggered sensing with minimal delay, which poses unique challenges for WSSNs and most traditional priori time synchronization strategies are not applicable therein. For a better illustration, one particular SHM application is described, and three main challenges for synchronized sensing in sudden-event monitoring are discussed.

4.1.1 Railway Bridge Monitoring

A WSSN was installed on the single-span steel railway bridge to monitor bridge vibrations under train-crossing events (see Section 3.4 for bridge details). The sensor network consists of 4 sensor nodes and 1 gateway node, and they were deployed for over one month, beginning in July 6th, 2018 (Figure 4.1). Particularly, the train-crossing events can be considered as sudden events. Therefore, *Demand-based WSSs* developed in Chapter 3 were employed, which were able to capture the occurrence of train-crossing events and provide high-fidelity data for bridge condition assessment in an efficient manner. An average of 15 train-crossing events per day were captured and uploaded to the cloud for data analysis, e.g., bridge displacement estimation. However, a time synchronization strategy applicable for sudden-event monitoring has not been developed and implemented for the deployment due to several specific challenges, described in the next subsection.

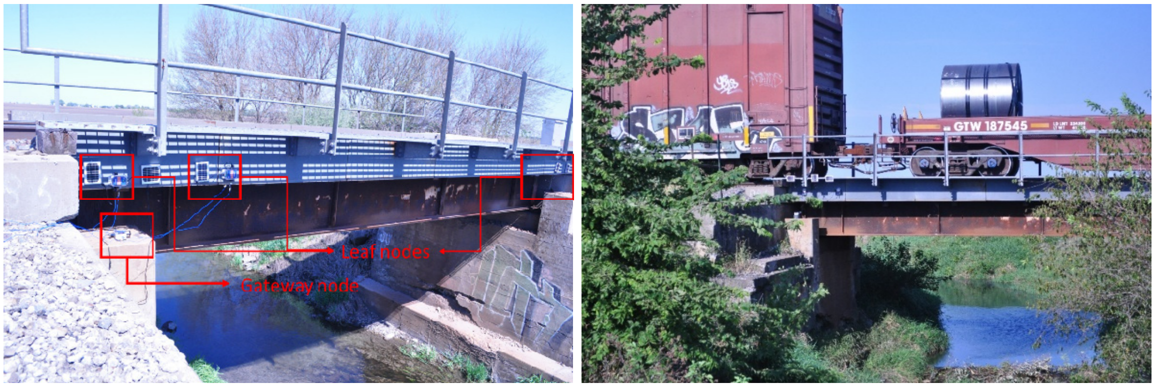


Figure 4.1 Railway bridge monitoring using *Demand-based WSS.*

4.1.2 Time Sync Challenges of Sudden-Event Monitoring Using WSS

This section discusses the unique challenges of time synchronization encountered in sudden-event monitoring, including unpredictability of sudden events, uncertainty of triggering time, and data loss of the initial transient response. In addition, sudden-event monitoring shares several challenges with traditional SHM, including stringent requirement of data synchronization precision and temperature variation during sensing, the details of which can be found in the paper (Li et al., 2016) and are not detailed herein.

(1) Unpredictability of sudden events

Traditional objectives of SHM applications are to assess the structural conditions either under ambient vibrations or subjected to user-controlled forces. Accordingly, wireless sensors are designed to be proactive, allowing users to spend sufficient time to configure sensors and schedule measurement. Conveniently, the gateway node is controlled by users or pre-programmed to coordinate sensor nodes not only for specifying sensing parameters but also for synchronizing local clocks throughout the network. In contrast, SHM for sudden events requires sensors to be reactive, initiating measurement immediately in response to the detection of events the timing of which is generally unpredictable. Accordingly, traditional time synchronization strategies designed for proactive sensor network is no longer practical in sudden-event monitoring, because it must be initiated by users or schedules and generally coordinated by the gateway node.

In the long-term deployment for railway bridge monitoring, all the sensor nodes were equipped with event-triggered technology to capture train-crossing events. After detecting events and obtaining measurement data, sensor nodes stored the data locally. In addition, the gateway node is scheduled to wake up every 5 minutes, checking if there are data sets available in sensor nodes to be collected. If traditional schedule-based time synchronization is employed, the long waiting time for the gateway node to wake up prior to time synchronization will introduce significant variations and uncertainties in clock drifts, resulting in inaccurate time synchronization.

(2) Uncertainty of triggering time

In traditional campaign-type monitoring, the gateway node first wakes up all the sensor nodes and then requests them to start sensing at approximately the same time. Because of random variation in task processing using event-driven operating system in most traditional WSSs, the start-up sensing time in different sensor nodes has a variation on the order of milliseconds, resulting in large data synchronization errors. Nagayama &

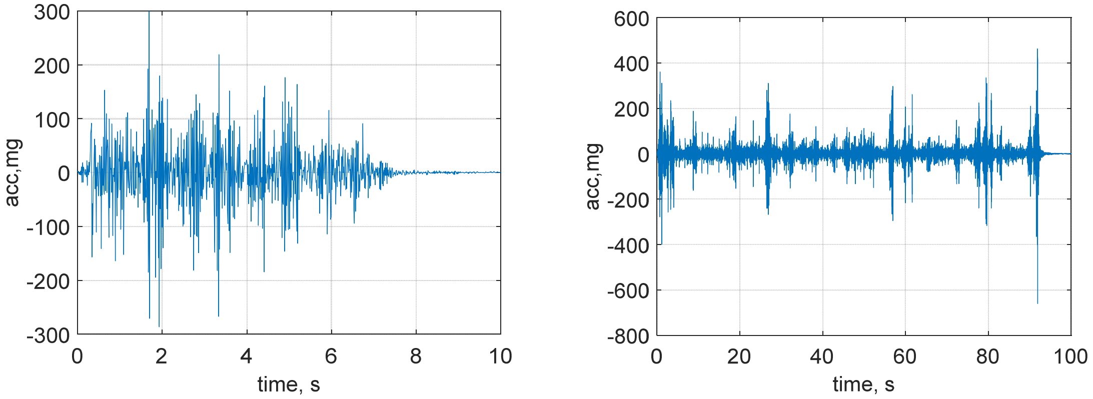
Spencer (2007) proposed an efficient resampling-based strategy to successfully address the issue in long-term periodic monitoring and campaign-type monitoring. However, the issue is more challenging in sudden-event monitoring leveraging event-trigged sensing. In particular, due to varying vibration levels in the structure at the onset of a sudden event, there is significant uncertainty in the triggering time for each sensor node. In an extreme scenario, some nodes may not be triggered to initiate sensing, if the vibration level is lower than the detection threshold. As a result, the difference in start-up sensing time among sensor nodes will become much larger, even on the order of seconds. Without the participation of the gateway node, coordination of sensor nodes with large uncertainties in start-up sensing time for time synchronization is challenging.

In the railway bridge deployment, sensor nodes were installed at different sides of the bridge. When a train crossed the bridge, all the nodes woke up, but were triggered at different time, depending on the train direction. In the meantime, wired sensor were also deployed close to wireless sensors, serving as reference data. The results show that, though two wireless nodes had a short distance of less than 1m, their start-up sensing time had a difference of 0.05s when an Amtrak passenger train moved cross the bridge. The triggering delay can be more significant for bridges with longer spans, which poses challenges for time synchronization implementation.

(3) Risk of data loss in initial transient response

Recent developments in event-triggered sensing provide cost-effective wireless solutions to capture sudden events and enable rapid condition assessment. Even if triggered immediately, one of the most important issues for WSS herein is that critical data between the occurrence of events and the time that data begins to be collected will be lost, mainly due to response latency in essential processes (e.g. cold boot). To this end, the *Demand-based WSS* provides a universal solution by fusing trigger accelerometer data and high-fidelity sensor data to produce a complete representation of the acceleration record (Fu et al., 2018). However, the risk of data loss will rise again if the traditional time synchronization is employed, in which certain amount of time is spent to exchange clock information prior to sensing. For example, 30s in the Jindo Bridge monitoring deployment (Jang et al., 2010) is allocated to perform time synchronization before sensing starts, and thus, a delay is introduced, resulting in missing of a significant portion of sudden events. Therefore, achieving synchronized sensing without data loss remains challenging.

In the case of railway bridge monitoring, the *Demand-based WSS* started sensing if the vibration were above 80mg for 0.02s and stopped sensing if the vibration were below 40mg for 5s, which guaranteed the capture of entire transient structural response under train-crossing events. Two types of train-crossing events were captured, including Amtrak passenger trains with a duration of around 8s, and freight trains with a duration of 1minute to 5minutes, as shown in Figure 4.2. Any extra operations including a priori time synchronization added before sensing may result in the miss of entire Amtrak train-crossing events and significant portion of CN train-crossing events.



(a) (b)

Figure 4.2 Train-crossing events monitoring: (a) Amtrak passenger train, (b) CN freight train.

4.2 Time Sync Using Next-generation Wireless SHM Systems

As discussed in Chapter 3, the next-generation wireless smart sensor platform, Xnode, is leveraged. In this section, the Xnode is employed to develop and implement efficient time synchronization strategies.

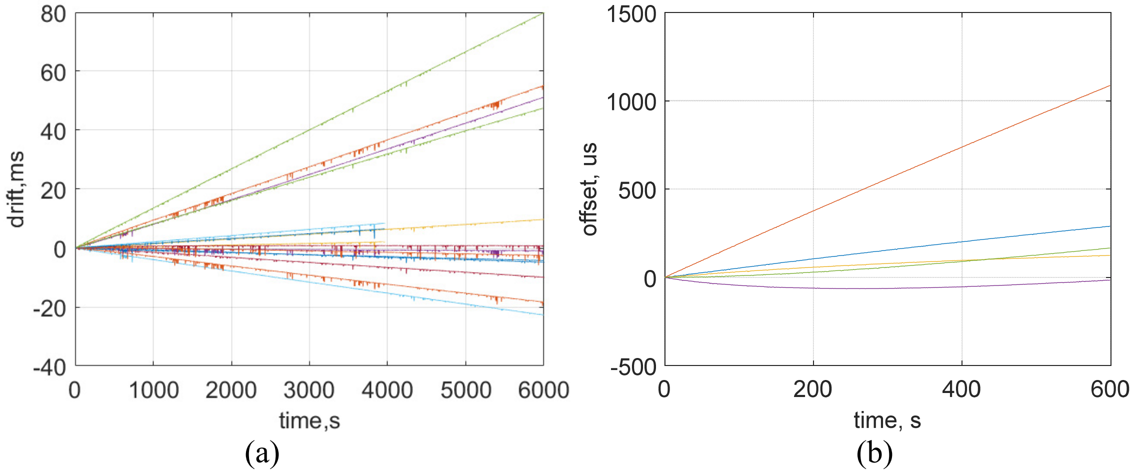
4.2.1 Estimation of Time Synchronization Error Using Xnodes

In SHM using wireless smart sensors, the clock crystal of the microprocessor usually serves as the reference for time stamping in end-user applications, whilst the precision of the sampling rate is generally governed by the clock crystal of analog-to-digital converter (ADC). In a sensing application, users specify parameters including sampling rate and sensing start-up time on the gateway node. Upon receiving the sensing parameters and initiating their sensor drivers, sensor nodes start to collect samples at the user-specified time based on the processor clock crystal and at the user-specified sampling rate based on the ADC clock crystal. In Xnodes, the first one is on the processor board, referred as the local clock, and the second one is on the sensor board. Therefore, the clock behavior of the two crystals must be examined to better understand the issues in synchronized sensing using Xnodes. This section describes the lab tests carried out to this end.

(1) Clock drift estimation

To examine the clock crystal on the processor board, the FTSP protocol is implemented on Xnodes, in which the gateway node broadcasts a series of packets to several sensor nodes. In the gateway node, when a packet is being transmitted, a time stamp, t_{send}, is recorded immediately before the transmission. Instead of appending t_{send} in the current packet, it is carried in the next packet, serving as reference time. This is because the time between appending time stamps and transmitting packets is found to be more than 1 millisecond on the Xnode, which cannot be neglected. When the sensor nodes receive the packet, a time stamp, $t_{receive}$, recorded from their local clocks is appended to the packet immediately upon its reception. The pair of clock information (t_{send}, $t_{receive}$) is then transmitted from each sensor node back to the gateway node. The value of $t_{receive}$ - t_{send} consists of the time (T1) spent for extracting/decoding the packet in the receiver, and the offset (T2) between local clocks and the reference time. T1 is considered as a constant value, and hence the trend of $t_{receive}$ - t_{send} is governed by T2.

This approach is first utilized to estimate clock drift between local clocks and reference time in Xnodes without considering temperature effects. In the experiment, the gateway node broadcasted time-stamped packets to 15 sensor nodes at 2-second intervals over 100 minutes. In the process, sensor nodes did not perform any other operations (e.g. sensing), and the on-board temperature is assumed to be constant over the period. The collected offset values, $t_{receive}$ - t_{send}, from 15 sensor nodes are plotted in Figure 4.3a. As can be seen, the clock drifts of the clock crystals on the processor boards are largely linear, indicating that the clock rates are stable over long time; but there are slight differences among the processor boards. The maximum clock drift difference is approximately 13μs per second, which is around 4 times smaller than that of Imote2s reported in the paper (Nagayama & Spencer, 2007). In addition, temperature effect on the clock drift was examined. Generally, when temperature changes, the resonant frequencies of clock crystals will also change, resulting in nonlinear clock drift (Li et al., 2016). To assess the clock crystal behavior under temperature variation, the same approach was utilized to record clock offsets, whilst the sensor nodes performed sensing operations. Particularly, the sensing operations caused heat accumulation on the processor and ADC chips, changing the on-board temperature. In the test, the gateway node broadcasted time-stamped packets at 1-second intervals over 10 minutes. As shown in Figure 4.3b, meaningful nonlinearity of clock drifts is observed in three sensor nodes. In sum, the clock offset of the clock crystal on the processor board of Xnodes grows as the measurement time increases, which is also affected by temperature variation. The linear/nonlinear clock drifts should be efficiently compensated, otherwise they will negatively affect subsequent data analysis. The detailed discussion of the clock drift effect can be found in the paper (Li et al., 2016).



(a) (b)

Figure 4.3 Clock drift measurement of the crystal on the processor boards: (a) without temperature effect, (b) with temperature effect.

Due to random variation in processing time and potential conflicts between various operations in wireless smart sensors, variable delay occurs when the gateway node records the transmission time of packets (t_{send}) and the sensor node records the reception time of packets ($t_{receive}$). A similar phenomenon can also be found in Imote2s, where large variation with scattered spikes was observed in time synchronization using time-stamping packets (Nagayama & Spencer, 2017). This error becomes larger and more frequent when time-stamping operation is conducted during sensing, due to scheduling conflicts with sensing or other operations. Accordingly, another test was conducted using the same approach to check the variance of time stamping error utilizing the clock crystal in the Xnode processor board. In particular, the gateway node broadcasted time-stamped packets to 5 sensor nodes every 5 milliseconds over a period of 5 seconds, a process which was repeated for 3 times. The collected offsets, $t_{receive}$ - t_{send}, of all the repeated tests were recorded by the gateway node and are plotted in Figure 4.4a. In the figure, the variation in time stamping can be observed, as well as several scattered large spikes, which demonstrates that the time stamping error cannot be neglected. Furthermore, the obtained offsets over each interval of 100ms were subtracted by their mean value; the corresponding time stamping errors were then obtained and are plotted in Figure 4.4b. It can be seen that the time stamping error has a normal distribution, and the observed maximum error is up to 36 μs. As a result, clock synchronization relying on a single time-stamped packet is not reliable, and time-stamping operation must be performed carefully to reduce the observed error.

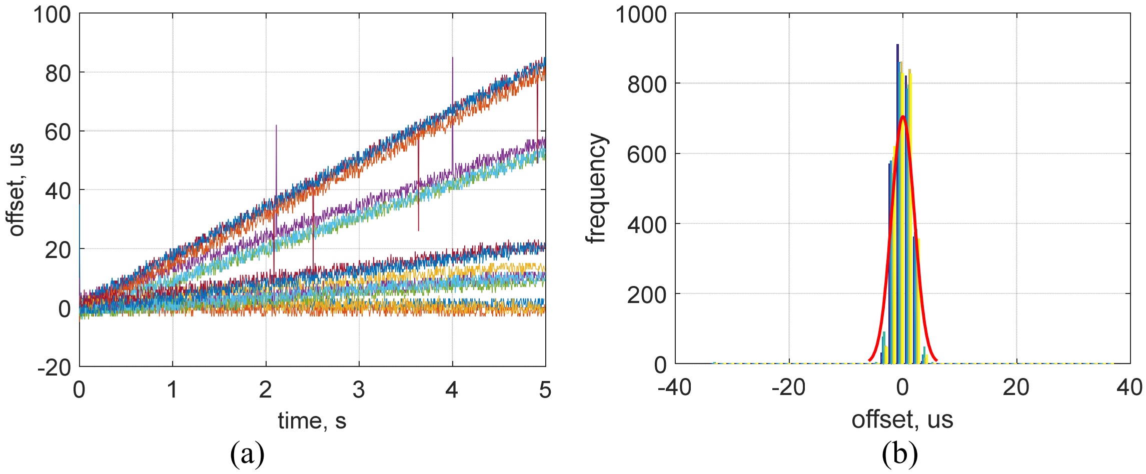
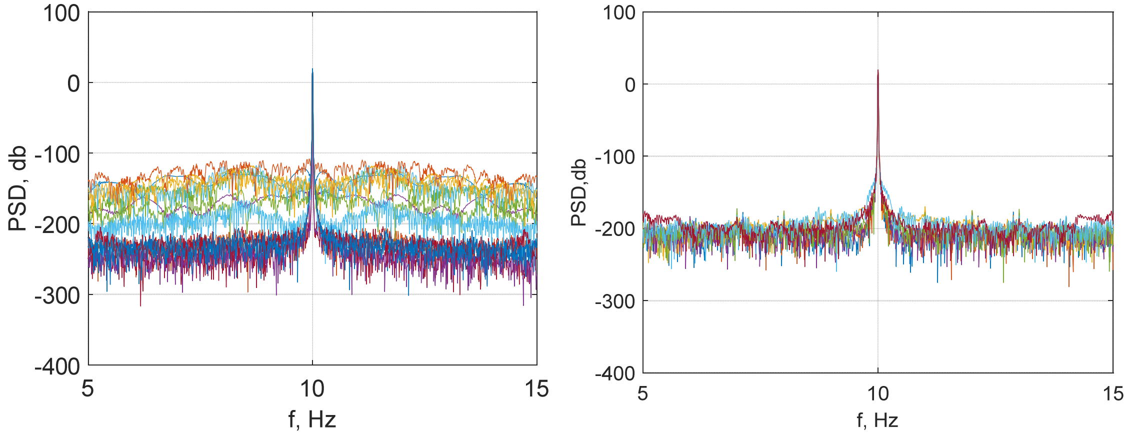


Figure 4.4 Uncertainties of time stamping error: (a) time stamping error over short time, (b) time stamping error distribution.

(2) Sampling rate variation

A similar approach using FTSP is not feasible to examine the clock crystal on the sensor board, because the clock crystal is wired directly to the ADC and reading it directly from the processor is not possible. Considering that the clock crystal on the sensor board determines the precision of the sampling rate, we estimated the clock error by examining the measured data of generated sine wave signals. Particularly, multiple sensor nodes were configured with the *RemoteSensing* application (Fu et al., 2016), which enables distributed data acquisition of external signals. These nodes were then connected to a dynamic signal analyzer, VibPilot, (m+p international). The VibPilot is utilized to generate a signal of sine wave with a sampling rate of f Hz continuously. The sine wave signal generated by the VibPilot is considered high-precision and stable. If the clock crystal on the sensor board ticks at exactly user-specified frequencies, the measured data samples have a peak at exactly f Hz in frequency domain; otherwise, the observed peak will be located at $f_s / f_r \cdot f$ Hz, in which f_s is user-specified sampling rate and f_r is the actual tick rate.

This approach is first used to examine the sampling rate variation in multiple sensor nodes under room temperature. A total of 15 sensor boards were tested to measure a 10Hz sine wave over a period of 2 minutes. Subsequently, power spectral density (PSD) of the obtained data samples was calculated and is plotted in Figure 4.5a. The sampling rate is specified as 100Hz, and the obtained peaks in all PSD curves are located at exactly 10Hz. The resolution of frequency in PSD calculation herein is 0.0083Hz. Accordingly, it can be claimed that the variation of sampling rate is negligible, which should be less than 0.1 percent, if it exists. In addition, the same approach was used to investigate the temperature effect on the sampling rate error. In particular, a sensor node was set to collect 10Hz sine wave signals from the VibPilot at a sampling rate of 100Hz. A heat gun was used to affect the on-board temperature, which was increased from 30° C to 65 ° C at a 5° C intervals. At each temperature level, the node measured the signals over 1 minute, and the PSD of obtained data samples was calculated, as plotted in Figure 4.5b. All PSDs match well with each other in that their peak value is at 10Hz, demonstrating that the temperature effect can be neglected on the clock crystal on the sensor board. In sum, sampling rates are very stable and accurate among different sensor nodes, regardless of the temperature effect. Furthermore, in the current configuration of the Xnode, the stability of the ADC clock crystal is much higher than the processor one.



(a) (b)

Figure 4.5 Sampling rate examination: (a) sampling rate for 15 sensor boards; (b) sampling rate of one node between 30°C~65°C.

Another concern regarding the clock crystal behavior is the sampling rate fluctuation over time. During sensing operation, the Xnode collected data samples from the ADC at a nominal rate of 1000Hz and then decimated the data to the user-specified sampling rate. To examine the sampling rate fluctuation, the time stamp was recorded when the node obtained a data sample from the ADC, utilizing the processor clock crystal. Subsequently, the intervals of consecutive data samples were obtained. In the test, the sampling intervals over 60 seconds are plotted in Figure 4.6. As can be seen, the sampling interval is very stable. The slight variation around the nominal value of 1000 μs is observed in the recorded sampling intervals, falling into the range of -4μs to 4μs. It is due to the time stamping error using the processor clock crystal and can be eliminated by taking the average of several consecutive data points.

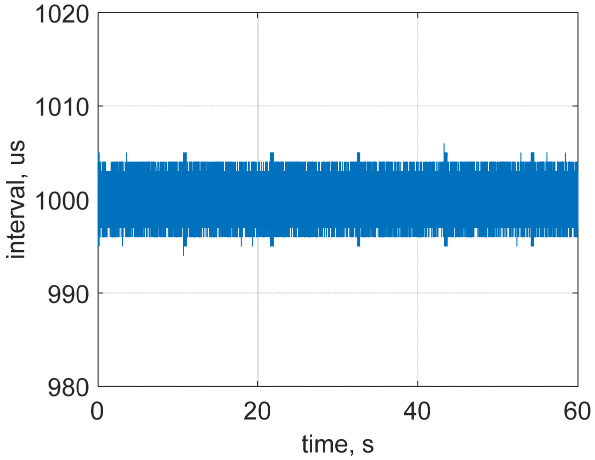


Figure 4.6 Sampling interval over time.

4.2.2 Two-stage Time Synchronization Strategy

As summarized in the papers (Nagayama & Spencer, 2007; Li et al., 2016), the challenges to achieve synchronized sensing using WSSNs exist in the following aspects:

(1) Due to random variation of processing time in WSS, the start-up sensing time is generally different from the user-specified time, and can vary among sensor nodes;

(2) Due to generally low-precision clock crystals in inexpensive embedded nodes, the actual sampling rate is often different from the nominal rate, with a large variation among nodes;

(3) Due to jitter and low-precision clock crystals, frequency stability can be low, resulting in fluctuation of sampling rate over time in the same sensor node;

(4) Due to temperature change during sensing, nonlinear clock drift can occur, resulting in inaccurate time stamping of sensor data and degradation of subsequent data analysis;

(5) As shown in Figure 4, due to uncertainties in processing time and potential scheduling conflicts among operations, time-stamping error cannot be neglected, which is more significant if time stamping is performed during sensing.

Based on the test results summarized in Section 4.2.1, challenges of (2) and (3) can be neglected using the next-generation WSS; hence, the main challenges for time synchronization herein are the uncertainty in start-up sensing time, nonlinear clock drift, and time-stamping error. To address the remaining challenges, a two-stage efficient time synchronization strategy is developed and implemented on Xnodes, serving as a baseline strategy. It contains both clock synchronization and data synchronization. The proposed idea is similar to the strategies in the papers (Nagayama & Spencer, 2007; Li et al., 2016), but it only has two steps for clock synchronization, before and after sensing, respectively, significantly simplifying the implementation of time synchronization for WSS. It is actually an improved version of the second implementation of time synchronization strategy proposed by Li et al. (2016) but considering the time-stamping error to achieve higher-precision synchronized sensing. More precisely, at the start of the sensing application, the gateway node sends start-up time of sensing T_{start} and the duration of sensing time; then end time of sensing is obtained T_{end}. As shown in Figure 4.7a, before sensing starts, the first round of point synchronization is performed, in which the gateway node broadcasts 10 packets at 5-millisecond intervals. The transmission time of packet *i*, $t_{gb}(i)$, is recorded right before the interrupt function of the radio is triggered and is stored in packet *i+1*. In sensor nodes, once packet *i* is received, the reception time, $t_{lb}(i)$, is recorded. After packet *i+1* is received. the corresponding offset is calculated,

$$\Delta t_j(i) = t_{lbj}(i) - t_{gbj}(i), \quad i \in [1,9] \tag{4.1}$$

where *j* is the round of point synchronization. After at least 5 offset values is obtained, the median of these offset values, ΔT_j, and corresponding local clock, t_j, are considered as the clock information pair for point synchronization. Subsequently, all sensor nodes start measurement roughly at the same time and record their local clock time when they collect the first data sample (t_{start}) and the last data sample (t_{end}), respectively. After sensing is completed, the second round of point synchronization is performed, and another clock information pair is obtained. The estimated local clock drift (*k*) and offset (*b*) are obtained based on two rounds of point synchronization,

$$k = \frac{\Delta T_2 - \Delta T_1}{t_2 - t_1}, \quad b = \Delta T_2 \tag{4.2}$$

which are unitized to correct the time stamps in sensor nodes,

$$t' = t_{lb} + b + k(t_{lb} - t_2) \tag{4.3}$$

In particular, the offset of start-up sensing time is estimated as follows,

$$dt = \frac{1}{2}\left(T_{start} + T_{stop} - t'_{start} - t'_{stop}\right) \tag{4.4}$$

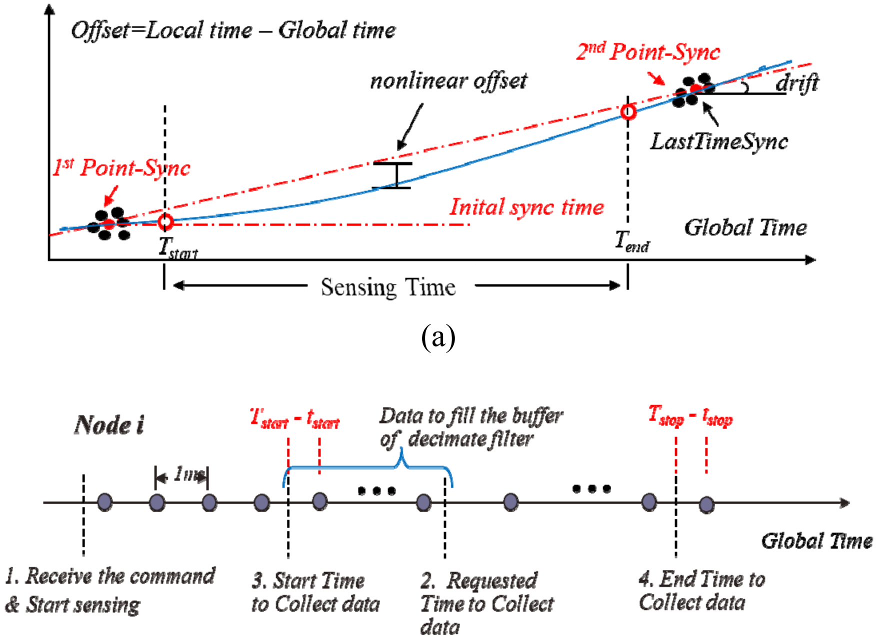
where t'_{start} and t'_{stop} are the corrected start time and stop time for the measurement. After clock synchronization, a resampling-based approach (Nagayama & Spencer, 2016) is used to achieve data synchronization (Figure 4.7b). The time synchronization strategy requires only two rounds of point synchronization, which is more efficient than the time synchronization strategy developed by Nagayama and Spencer [15] and the first implementation of time synchronization strategy developed by Li et al. (2016). Furthermore, it determines the time-stamping error carefully, resulting in higher accuracy than the second implementation of time synchronization strategy developed by Li et al. (2016).

A lab test was carried out to evaluate the performance of the proposed time synchronization strategy (Figure 4.8). Three Xnodes were programmed as sensor nodes which performed *RemoteSensing* application, whilst another Xnode served as the gateway node. The VibPilot was employed to generate a band-limited white noise (BLWN) with a frequency bandwidth of 20Hz, which was measured by sensor nodes. In the test, the gateway node first sent the command to sensor nodes with a request of start-up sensing time, sampling rate (100Hz), and measurement time (1minute, 10 minutes, and 30 minutes). The sensor nodes performed the two-stage time synchronization during sensing operations, and finally transmitted data back to the gateway node. To quantitatively evaluate the synchronization error, the equation in the paper (Li et al.,

2016) is applied herein, in which the obtained indicator, TS_{error}, can evaluate the error in μs, as follows,

$$TS_{error,ij} = \frac{\theta_{ij}}{2\pi} \times 10^6 \tag{4.5}$$

where θ_{ij} is the phase angle slope of the cross power spectral density (CPSD) within 0 to 20 Hz between two signals, *i* and *j*. If the nodes were perfectly synchronized, the measured signals from different sensor nodes must be exactly the same, and the phase angle between the signals must be zero. The tests of different sampling durations were repeated for 3-5 times, and the averaged results are listed in Table 4.1. As can be seen, the precision of the proposed time synchronization using Xnodes is approximately 10 μs. The minimized time synchronization error is attributed to both the high-precision crystals of the Xnode and the proposed efficient time synchronization strategy.


Offset=Local time – Global time
2nd Point-Sync
nonlinear offset
drift
LastTimeSync
1st Point-Sync
Inital sync time
T_{start}
T_{end}
Global Time
Sensing Time
Node i
T'_{start} - t_{start}
Data to fill the buffer of decimate filter
T_{stop} - t_{stop}
1ms
Global Time
1. Receive the command & Start sensing
3. Start Time to Collect data
2. Requested Time to Collect data
4. End Time to Collect data

(a)

(b)

Figure 4.7 two-point two-stage time synchronization: (a) clock synchronization, (b) data synchronization.

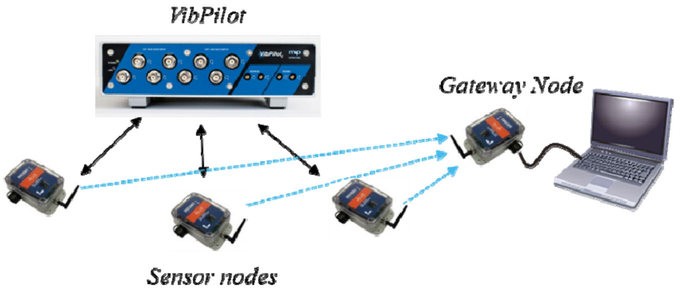


Figure 4.8 Test setup for time synchronization accuracy evaluation.

Table 4.1 Time synchronization result.

Sensing duration	Pairwise synchronization error		
	Pair 1	Pair 2	Pair 3
1min	6.89	4.58	10.92
10min	6.80	3.57	8.95
30min	9.74	11.90	12.82

4.3 Time Sync Strategies for Sudden-Event Monitoring

Besides the challenges of time synchronization for SHM summarized in Section 4.2, time synchronization for sudden-event monitoring poses three additional unique challenges, as described in Section 4.1. If leveraging the Xnodes equipped with high-precision clock crystals, the remaining challenges of time synchronization for sudden-event monitoring can be summarized by the following five aspects: (1) unpredictability of events onset, (2) uncertainty in triggering/start-up sensing time, (3) risk of data loss in the initial transient response, (4) nonlinear clock drift, and (5) time-stamping error. To address the challenges, several efficient time synchronization strategies are proposed in this section based on the two-stage baseline time synchronization strategy, summarized as follows,

- A post-event time synchronization is developed for sudden-event monitoring. Particularly, it employs reference node election for relative clock synchronization and postpones the time synchronization operation until event-triggered sensing is completed, without introducing any sensing delay. The post-event time synchronization meets the needs for most scenarios (e.g., earthquakes) where sudden events are short-duration.
- The post-event time synchronization does not perform well for long-duration events (e.g., downbursts) in which nonlinear clock drift is not negligible. Accordingly, a variant version, called pre-post-event time synchronization, is developed. Specifically, two rounds of point synchronization are conducted before and after event-triggered sensing. The strategy compensates for the nonlinear clock drift during extended sensing period, but initial structural response is lost, which is not as important in long-duration monitoring.
- The previous time synchronization strategies are both offline in nature, which may work poorly for some applications, e.g., real-time damage assessment during a natural disaster. Accordingly, a piecewise real-time time synchronization strategy is proposed, in which time synchronization is conducted periodically during event-triggered sensing.

The selection of the strategies is based on the types of sudden events and objectives of monitoring. The more details are illustrated in following subsections.

4.3.1 Offline Strategy: Post-Event Time Synchronization for Short-duration Sudden Events

Most sudden events have a short duration, ranging from several seconds to several minutes, for events such as earthquakes, bridge impacts, and explosions. For example, the main shock of the El Centro Earthquake lasts only 30 seconds. In these events, the risk of data loss in the initial transient response is critical, but the need for real-time condition assessment is not as strong. To meet the requirements, an offline time synchronization strategy is proposed, in which time synchronization is carried out immediately after sensing is completed, without introducing any delay for event-triggered sensing.

As illustrated in Figure 4.9, when an event occurs, demand-based sensor nodes initiate sensing immediately and start-up sensing time (t_{start}) is recorded using the local clock. Once they detect the end of the event, the nodes stop sensing accordingly and recorded the local clock as t_{stop}. Particularly, because of varying vibration levels in the structure at the onset/end of the event, t_{start} and t_{stop} are different among sensor nodes. After sensing is completed, each sensor node will participate in energy-aware reference node election. In the process, the node which completes sensing first will broadcast a run-in-election message continuously to declare that it is a reference node. The message contains the node's information including its node ID, voltage level and start-up sensing time. For other nodes, after sensing is finished, they will first listen for messages. If no message is received, they will broadcast their message to compete for reference election; otherwise, they will wait until the reference node election is finished. It is possible that two or more nodes broadcast their run-in-election message at approximately the same time. In this situation, those nodes will examine the information received from the other node's message. A node with the higher voltage level will win the election, because the reference node will consume more energy in the following stages. When the two nodes have the same voltage level, the node ID will be used as the tie breaker, with the lower ID elected as the reference node. This process will continue for 30 seconds to compensate for uncertainty in the triggering time, such that the nodes that initiate and stop sensing late can also participate in the election. Afterwards, the reference node will broadcast time-stamped packets based on its local clock for 15 seconds at a 1-second interval. In each point synchronization, a number of packets are sent to reduce the time-stamping error. The rest of nodes will perform relative clock synchronization by employing linear regression. The obtained clock drift (k) and offset (b) are obtained as follows,

$$k = \frac{n\sum \Delta T_i t_i - \sum \Delta T_i \sum t_i}{n\sum t_i^2 - (\sum t_i)^2}, \quad b = \frac{\sum \Delta T_i - k\sum t_i}{n}, \quad n \in [1, n] \tag{4.6}$$

where n is set to be 15 in the experiments. Then, the corrected t_{start} and t_{stop} are calculated using Eq.(4.3); the offset of start-up sensing time is also obtained using Eq.(4.4). Though the clock synchronization is performed post-event, the nonlinear clock drift can generally be neglected here, as the duration of the entire process is so short that nonlinear drift has little impact. Then, data synchronization is performed using the resampling-based approach. Finally, to enable rapid condition assessment, the reference node will send wake-up command and then notification message to the gateway node which is duty-cycled to preserve its battery power. Subsequently, the gateway node will send data retrieval request to sensor nodes and perform data analysis for decision making.

The proposed strategy addresses all of the aforementioned challenges. In particular, the entire process is automatic and efficient, with no need for a priori schedule information or user involvement. This addresses the event onset unpredictability challenge. The reference node election and the subsequent relative clock synchronization compensate for the rest challenges, including uncertainty in the triggering/start-up sensing time, risk of data loss in the initial transient response, and time-stamping error.

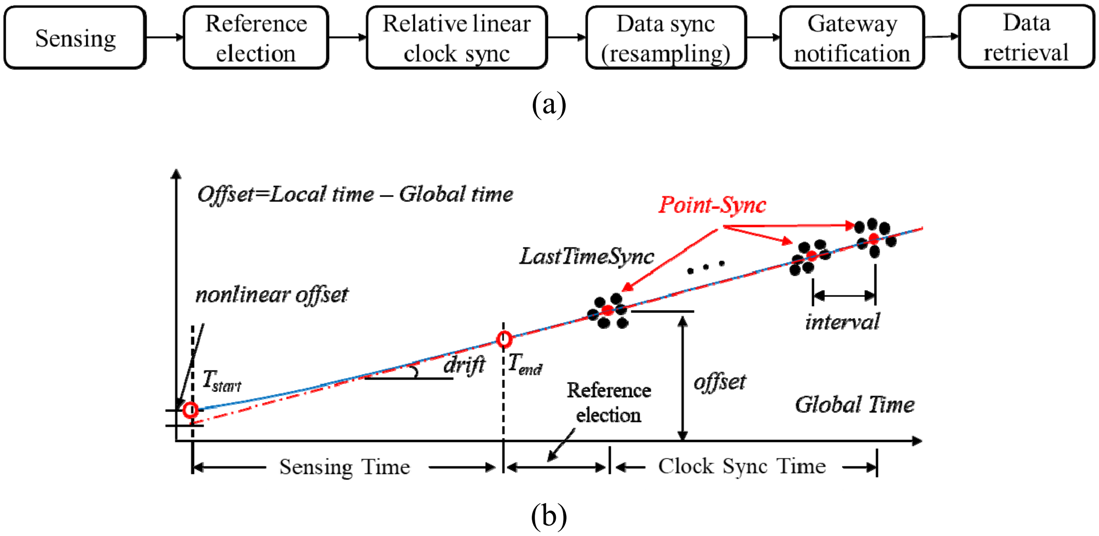


Figure 4.9 Time synchronization for short-duration sudden-event monitoring: (a) flowchart, (b) clock synchronization.

4.3.2 Offline Strategy: Pre-post-event Time Synchronization for Long-duration Sudden Events

Some sudden events have a long duration, several minutes up to hours, such as downbursts, hurricanes, and blizzards. For example, a downburst recorded in the paper (Huang & Chen, 2009) lasted half an hour, with a speed of above 15m/s over 15 minutes. In these long-duration sudden events, data loss in the initial measurement is not as important, but the effect of nonlinear clock drift becomes more significant. Accordingly, another offline strategy is proposed to address the challenges of time synchronization in this situation. Note that for the events that have a duration of many hours are not considered in this paper, because the technologies of sudden-event monitoring are not required in these scenarios.

When an event occurs, a reference election is carried out first (Figure 4.10). The selected reference node performs first-phase point synchronization to synchronize clocks among sensor nodes. Then the nodes start sensing at approximately the same time. After the event ends, the reference node performs second-phase point synchronization. The subsequent time synchronization operation is the same with the two-stage time synchronization described in Section 4.2, but the difference is that the clock synchronization herein is relative time synchronization among sensors, without the involvement of the gateway node. Finally, the reference node wakes up the gateway node and notifies the occurrence of sudden events. Data retrieval is conducted afterwards, to collect data from sensor nodes for subsequent data analysis and decision making.

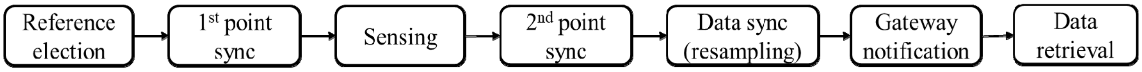


Figure 4.10 Time synchronization for long-duration sudden-event monitoring.

4.3.3 Online Strategy: Real-time Time Synchronization for Long-duration Sudden Events

One of the critical objectives for sudden-event monitoring is to enable rapid condition assessment. In some cases, online data analysis is required, especially for sudden-event monitoring with a duration of over 15 minutes. If offline condition assessment is implemented, informed decisions cannot be made in an efficient manner until the event ends and the data transmitted afterwards. As a result, the previous offline time synchronization strategy may not be applicable in this scenario. Therefore, another strategy using piecewise real-time time synchronization is developed for online data analysis.

The main obstacle to achieve the real-time time synchronization is the scheduling conflict between time synchronization operation and sensing process. In particular, traditional wireless sensors employ event-driven operating systems such as TinyOS in which tasks are executed in a First In, First Out (FIFO) manner. Adding a time synchronization task will not perform well, affecting the timing accuracy of both time synchronization and sensing. To avoid the potential scheduling conflicts, researchers prefer to employ pre-sensing time synchronization or post-sensing time synchronization (Nagayama & Spencer, 2007; Li et al., 2016). In contrast, the proposed strategy leverages the preemptive multitasking framework implemented on the Xnode (Spencer, et al., 2016) to enable periodic time synchronization operation without affecting sensing process. More precisely, the sensing task is assigned with a higher priority, whilst another task is created for time synchronization operations with a relatively lower priority, without affecting the accuracy of sensing operation. More detailed discussion regarding preemptive multitasking framework can be found in Section 5.1. Note that the real-time time synchronization strategy proposed here is more meaningful when it is integrated with real-time data acquisition for rapid condition assessment. More details of development of the real-time data acquisition using Xnodes can be found in Chapter 5.

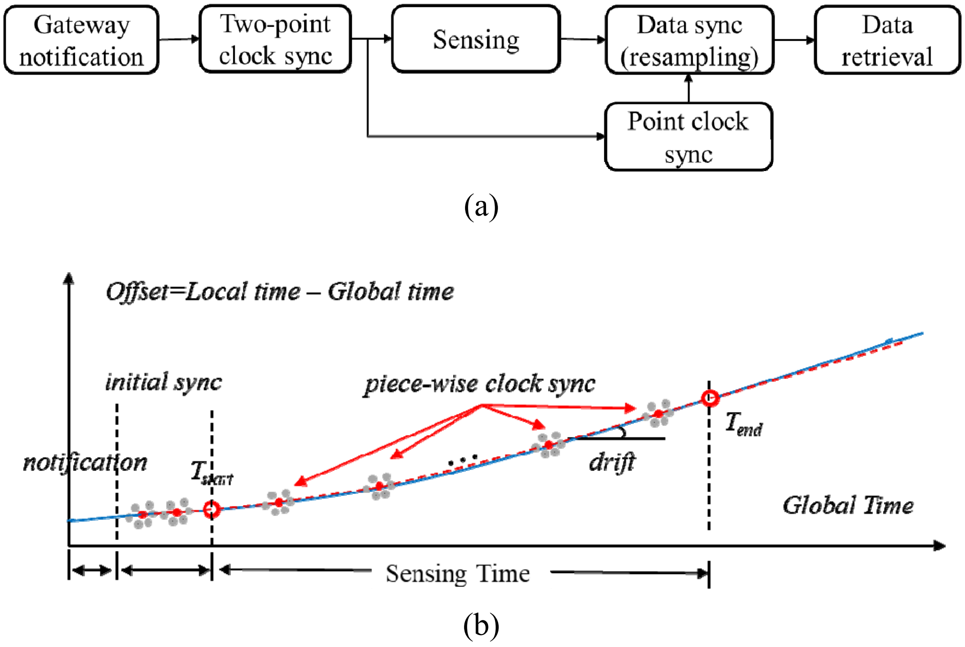
Figure 4.11 illustrates the concept of piecewise real-time time synchronization. After an event is detected, sensor nodes will send messages to notify the gateway node. Upon receiving the message, the gateway node broadcasts time synchronization messages to sensor nodes which perform two rounds of point synchronization to estimate initial clock drift (k_0) and offset (b_0). Afterwards, sensing is carried out; meanwhile, every 30 seconds, the gateway node broadcasts time synchronization messages and the sensor nodes conduct point synchronization to update clock drift (k_i) and offset (b_i).

$$k_i = \frac{\Delta T_i - \Delta T_{i-1}}{t_i - t_{i-1}}, \quad b_i = \Delta T_i \tag{4.7}$$

Where *i* is the index of the round of point synchronization. The updated clock drift and offset are utilized to correct the time stamps between the *i*th round and the (*i+1*)th round point synchronization.

$$t' = t_{lb} + b_i + k_i(t_{lb} - t_i), \quad t_i < t_{lb} < t_{i+1} \tag{4.8}$$

In the meantime, data synchronization is performed every 1s based on the corrected time stamps. For campaign type monitoring, the piecewise real-time time synchronization is carried out in sensor nodes, and resampled data is finally transmitted back to the gateway node. If real-time data acquisition is implemented, data is transmitted during sensing; the real-time time synchronization is performed on the gateway node.



(a)

(b)

Figure 4.11 Piece-wise real-time time synchronization for long-duration sudden-event monitoring: (a) flowchart, (b) clock synchronization.

4.4 Experimental Validation of Time Synchronization for Sudden-Event Monitoring

The performance of the proposed time synchronization strategies is evaluated experimentally through two types of lab tests. The first test has the same setup with that in Figure 4.8 to evaluate the precision of the proposed time synchronization strategies. The second test is conducted on an 18-story building model mounted on a shaking table, to demonstrate the capability of sudden-event monitoring using the proposed time synchronization under earthquakes. This section provides a detailed description and discussion of these tests.

4.4.1 Performance Evaluation of Time Synchronization Accuracy

The test setup is the same with Section 4.2, in which three Xnodes (Node 2,3,4) are set to measure a BLWN signal with 20Hz bandwidth which is generated by the VibPilot, and another Xnode (Node 1) is configured as a gateway node. However, the sensing scenario is sudden-event monitoring, which is different from Section 4.2. In particular, all the nodes are located on the same table, and they do not initiate sensing unless the researcher hits the table with a hammer to mimic real sudden events. The mechanism of event-triggered sensing employed here in the paper (Fu et al., 2018). The effectiveness of the three proposed time synchronization strategies is assessed by considering various sensing durations. Each test was repeated multiple times, and the hammer hit different areas of the table to ensure that each node had a chance to be elected as the reference node. Figure 4.12 shows representative examples of test results for each time synchronization strategy. Figure 4.12a is the time history data using the post-event time synchronization strategy for 1-minute measurement. As can be seen, Node 2 was triggered first to start measurement, and hence elected as a reference node. Other two nodes started measurements approximately 0.05s later; the beginning segment of their time history data is not in line with that of the reference node. After 0.06s, excellent agreement can be observed between the measurement data from three nodes, indicating they are well synchronized. As an index of time synchronization error, the phase angles θ_{ij} of CPSD between the reference node and other two nodes are shown in Figure 4.12b, and are very close to 0, within the range of $-5\times10^{-5} \sim 5\times10^{-5}$. Compared to Node 2, the associated synchronization errors for Node 3 and Node 4, are 2.04μs and 5.62μs, respectively. Similarly, Figure 4.12c-d shows the test result using the pre-post-event time synchronization strategy for 10 minutes measurement, in which Node 2 is the reference node. In the pre-post-event time synchronization strategy, the measurement is conducted after reference election, which starts roughly at the same time for three nodes. The corresponding time synchronization errors for Node 3 and Node 4 are 0.91μs and 13.23μs, respectively. Likewise, Figure 4.12e-f shows the test result using the real-time time synchronization strategy for 25 minutes measurement. Excellent agreement between the measurement data from three sensor nodes. The associated time synchronization errors are all less than 15μs.

The averaged test results of three time synchronization strategies are summarized in Table 4.2. Overall, the precision of the proposed time synchronization strategies is high with time synchronization errors under 20 μs in all cases. The post-event time synchronization strategy achieves high time synchronization precision with an average error of around 10 μs. Though it employs linear extrapolation to estimate the clock drift of event-triggered sensing, the high precision verifies that nonlinear clock drift is negligible in short-duration monitoring. The pre-post-event time synchronization strategy shows consistent good performance for long-duration monitoring of up to 15 minutes, demonstrating its efficacy to compensate nonlinear clock drift. The piecewise real-time time synchronization strategy also shows excellent performance over extended sensing duration. Though linear secant slope is employed here, the high precision indicating that piecewise linear estimation is adequate for long-duration measurement. Theoretically, the real-time time synchronization strategies can support high synchronized sensing of any duration.

Table 4.2 Test results of time synchronization for sudden-event monitoring.

time synchronization strategy	Sensing duration (min)	Pairwise synchronization error (μs)		
		Pair 1	Pair 2	Pair 3
Post-event time synchronization	0.5	5.40	3.70	4.46
	1	6.86	8.56	8.07
	3	6.32	8.87	9.32
Pre-post-event time synchronization	5	6.16	10.56	10.83
	10	7.89	11.52	11.65
	15	5.60	16.35	11.74
Piecewise real-time time synchronization	5	1.08	12.01	11.56
	15	2.64	12.47	13.01
	25	2.14	18.68	17.79

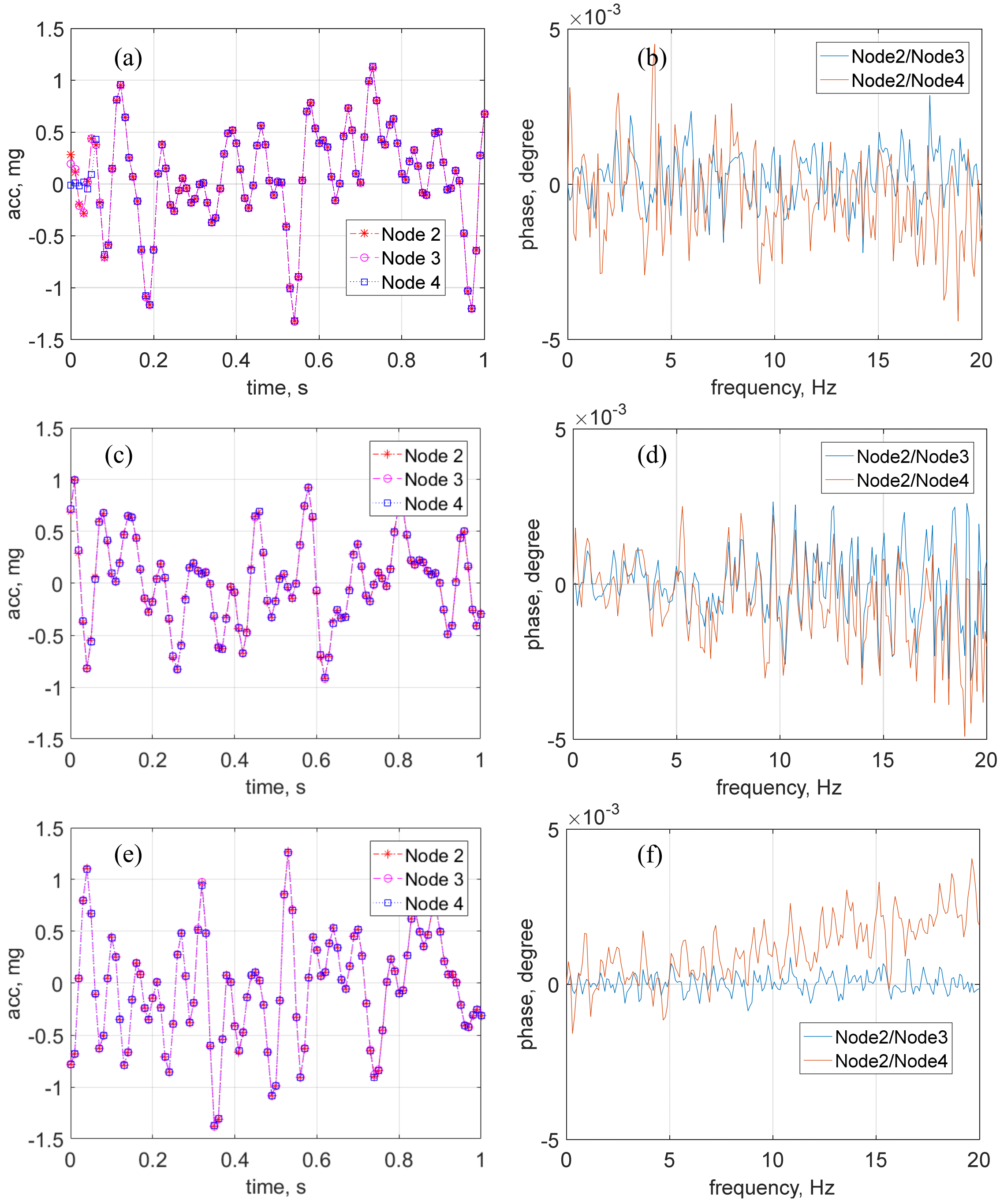


Figure 4.12 Time synchronization for sudden-event monitoring: (a) post-event time synchronization data (1min), (b) post-event time synchronization phase delay (1min), (c) pre-post-event time synchronization data (10min), (d) pre-post-event time synchronization phase delay (10min), (e) piecewise real-time time synchronization data (25min), (d) piecewise real-time time synchronization phase delay (25min).

4.4.2 Application of Time Synchronization Strategies for Sudden-Event Monitoring

To further validate the efficacy of time synchronization strategies in sudden-event monitoring, another lab test was conducted for seismic structural monitoring (Figure 4.13). Specifically, an 18-story building model was mounted on a uniaxial shaking table which could generate ground excitation in one horizontal direction. A total of five *Demand-based WSSs* were installed at floors 4, 8, 12, 15 and 18. Wired piezoelectric accelerometers, model PCB393B12, were installed on the same floors as reference sensors for comparison. The 30s El Centro earthquake excitation was generated by the shaking table to represent a sudden event. Threshold criteria of event detection for *Demand-based WSSs* was established as follows: the onset of the event was detected when the acceleration remained above 80mg for 0.02s, and the end of the event was detected when acceleration remained below 40mg for 5s. The wireless sensors were configured with a sampling rate of 100Hz, and a gateway node was connected with a PC to retrieve data after the events. Considering that the earthquake excitation is short-duration, the post-event time synchronization strategy is implemented. For the wired counterparts, the sampling rate was configured as 128Hz, and the VibPilot was employed to acquire the data continuously from the 5 wired sensors at the same time.

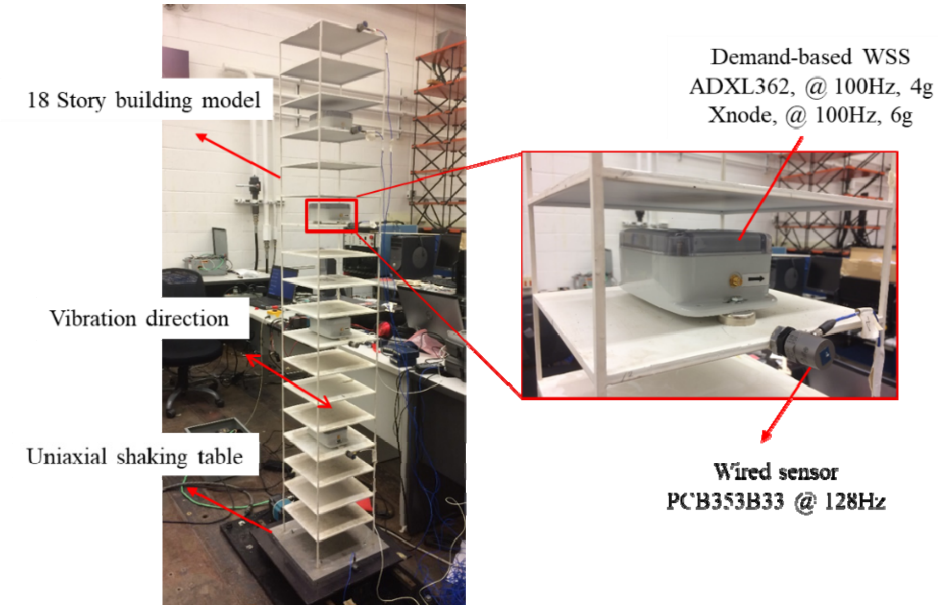
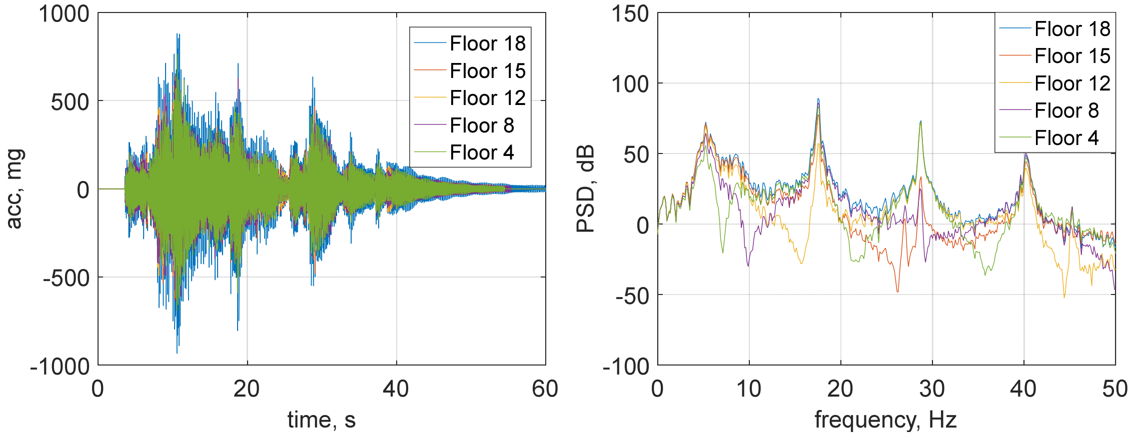


Figure 4.13 Test setup for building monitoring under earthquakes.

The sudden events were captured by *Demand-based WSSs* successfully. Due to varying vibration levels, the difference of triggering time for each node is up to 0.34s for various floors. By utilizing the proposed offline time synchronization strategy, the time history records from multiple wireless sensors were well synchronized. The measurement data and their PSDs are shown in Figure 4.14. For comparison of data from wired and wireless sensors, the acceleration data obtained from wired sensors was down-sampled from 128Hz to 100Hz. The data sets from wired and wireless systems were then synchronized by maximizing their cross-correlation function. The comparison of test results is shown in Figure 4.15, in which measurements from Floor #18, #15, and #12 are plotted regarding time history data and PSD. The excellent agreement between the results of the two monitoring systems in the both time and frequency domain from different floors demonstrate the capability of the proposed time synchronization strategy for synchronized seismic sensing using multiple sensors.

During the test, another wired sensor was installed on the table of the shaker to measure ground excitation. The acceleration responses of the structure measured by two monitoring systems and the ground excitation record were used together to calculate frequency response function (FRF). Modal analysis was then conducted using the peak picking method. The obtained natural frequencies are listed and compared in Table 4.3. In addition, the mode shapes estimated by peak picking from the FRF are plotted in Figure 4.16; the modal amplitude coherence (MAC) of mode shapes obtained from two monitoring system is also analyzed and listed in Table 4.3. The modal analysis results from wired and wireless sensor, especially the mode shapes, match very well, demonstrating that wireless sensor data is synchronized accurately.

The aforementioned test results validate the effectiveness and precision of proposed time synchronization strategies for sudden-event monitoring. In particular, the time synchronization strategies are performed automatically, regardless of when the unexpected sudden event occurs; the time synchronization strategies introduce minimal delay in event-triggered sensing, without sacrificing the high-level precision; they are robust under temperature variation, keeping high precision for an extended sensing period.

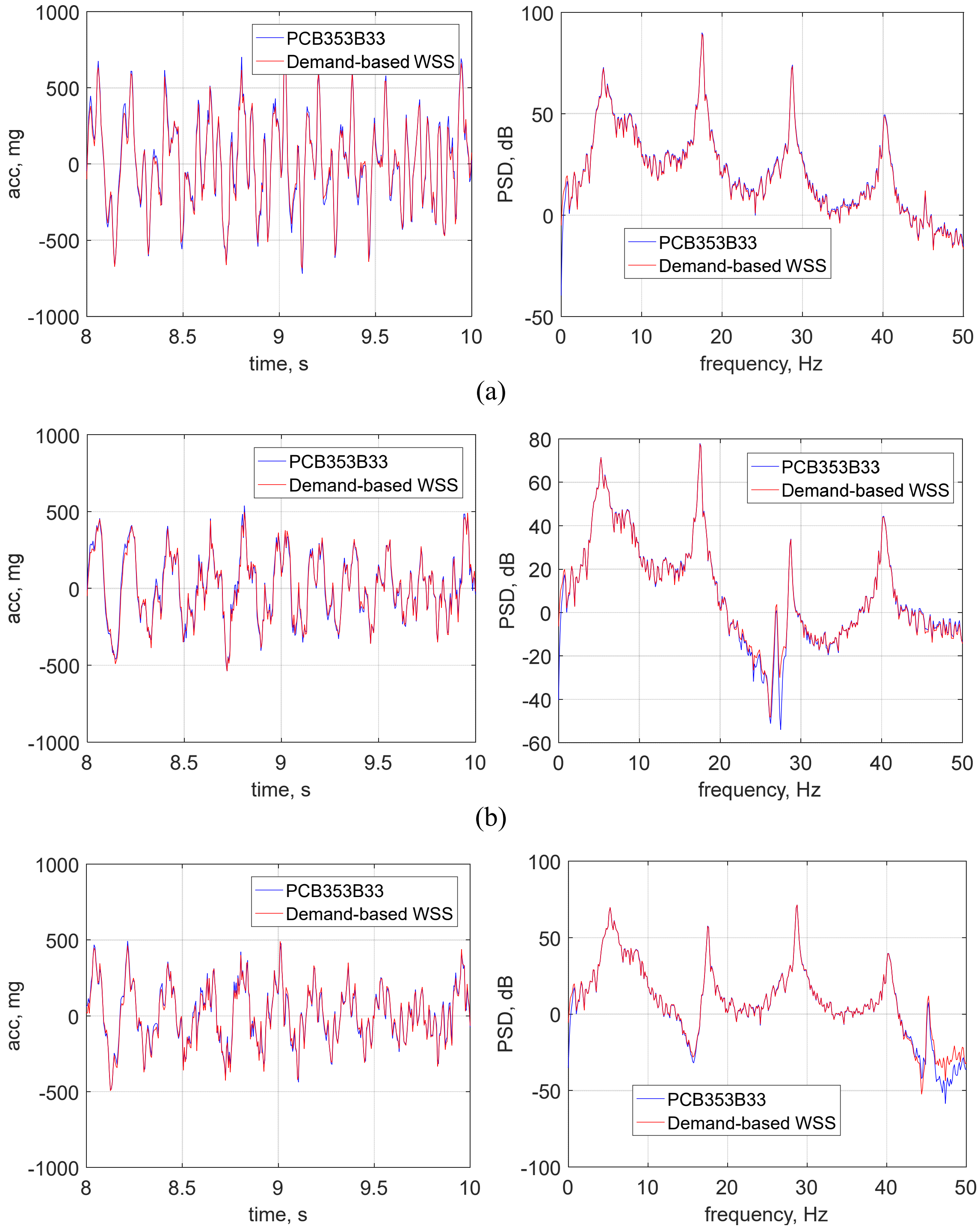


(a) (b)

Figure 4.14 Test results of earthquake monitoring: (a) time history data, (b) PSD data.

Table 4.3 Modal properties comparison.

No. modes	Natural frequency comparison			Mode shape comparison
	WSS	wired sensor	difference	Modal Amplitude Coherence
1st	5.73	5.75	0.35%	0.9998
2nd	17.42	17.38	-0.23%	1.0000
3rd	28.64	28.63	-0.03%	0.9998
4th	40.21	40.25	0.10%	1.0000



(a)

(b)

(c)

Figure 4.15 Test results comparison between WSS and wired sensors: (a) Floor #18, (b) Floor #15, (c) Floor #12.

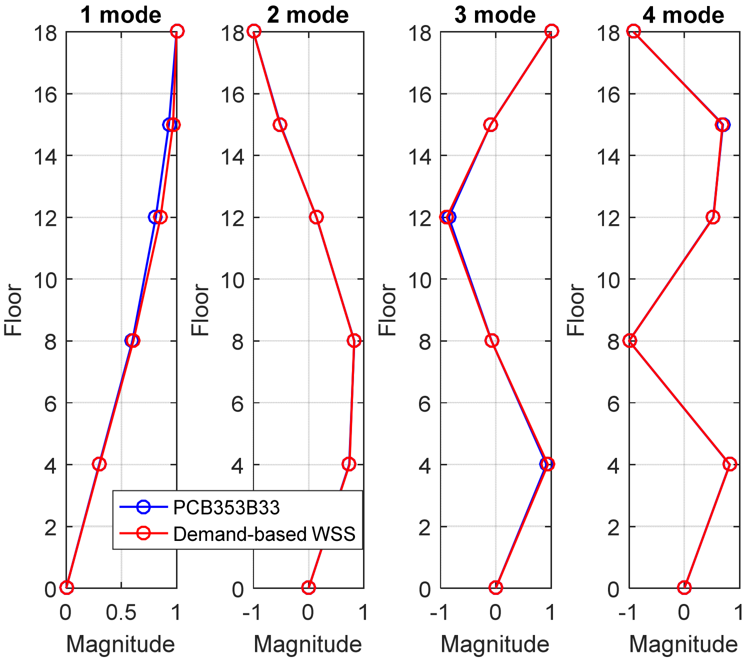


Figure 4.16 Mode shape comparison.

4.5 Summary

In this chapter, the unique challenges of synchronized sensing for sudden-event monitoring using WSSNs are discussed. The challenges include unpredictability of events onset, uncertainty of triggering time, and risk of Data loss in initial transient response, which combined with other challenges of time synchronization in traditional SHM applications make the synchronized sensing extremely difficult for sudden-event monitoring. To facilitate the development of efficient time synchronization strategies, the next-generation WSS, Xnode, is leveraged. The Xnodes' clock behavior was first examined to better understand the issues towards synchronized sensing; an efficient two-stage time synchronization strategy was developed and implemented on the Xnode, achieving high-accuracy time synchronization for traditional SHM applications. Afterwards, three efficient time synchronization strategies for sudden-event monitoring are designed for different use cases: two offline time synchronization strategies for short-duration monitoring and long-duration monitoring, respectively; one real-time time synchronization strategy for rapid data analysis in long-duration monitoring. The performance of proposed time synchronization strategies has been validated through lab tests, showing that the maximum time synchronization error is less than 20μs. In addition, a building model test has been conducted, demonstrating the capability of proposed time synchronization strategies to achieve synchronized sensing for sudden-event monitoring.

Chapter 5

REAL-TIME CONDITION ASSESSMENT FOR SUDDEN-EVENT MONITORING

In this chapter, two critical requirements for rapid damage assessment in sudden-event monitoring are presented. To meet these requirements, a high-throughput real-time data acquisition framework is first developed and implemented in the Xnodes. Its performance is validated in a lab test that, it allows near-real-time sensing of 36 channels across 12 nodes at 100Hz with a throughput of 115.2 kbps and without data loss. In addition, efficient online condition assessment is developed to support early structural damage estimation in real-time, which are validated in numerical simulations. Finally, a MATLAB-based application is designed to realize real-time data processing and visualization for end users.

5.1 Rapid Condition Assessment Requirements of Sudden-Event Monitoring

In contract to campaign-type monitoring or traditional long-term monitoring using WSSNs, sudden-event monitoring requires rapid condition assessment of civil infrastructure to support informed decisions in an efficient manner. Particularly, in some scenarios, real-time condition assessment is required for fast decision making with minimal delay, which poses unique challenges for monitoring systems. For example, in the history, a series of space grid structures failed under blizzards without any warning, resulting in significant economic loss and deaths. Most of these structural failures involve instability problem, which usually initiated from individual member buckling and then developed into progressive collapse in seconds. Accordingly, the traditional data collection and analysis style in WSSNs that have large latency are not applicable. The associated two main challenges which may result in significant latency for rapid condition assessment in sudden-event monitoring are discussed in this section.

(1) Response latency due to data transmission

In WSSNs, data acquisition is generally realized in three common ways: centralized data logging, decentralized data aggregation, and real-time data acquisition, all of which may have significant delay for rapid condition assessment. In the first, measurement data is first stored in sensor nodes locally and then transmitted back to the base station after sensing is completed. The transmission of raw time history data can take significant amount of time before data is analyzed by users, especially for long-duration measurement. In the second one, measurement data is first analyzed and condensed locally before transmitted back to the base station for subsequent analysis. Though the transmission time can be decreased by reducing the size of data, it is still not applicable for real-time condition assessment. In the last approach, data is transmitted back during sensing, making it possible for real-time data analysis in the base station. However, several challenges remained to impose the limitations on the application of real-time data acquisition using WSSN. Specifically, because of potential scheduling conflicts between sensing process and data transmission, the throughput of most real-time data acquisition strategies is very low, which is not applicable for large-scale networks. In addition, before real-time data acquisition, priori time synchronization and network configuration are generally required to coordinate sensor nodes and minimize radio interference, which, however, may take tens of seconds and result in data loss for sudden-event monitoring. In sum, an efficient and reliable data acquisition strategy for real-time condition assessment remains challenging.

(2) Lack of support for real-time data management

Rapid condition assessment in sudden-event monitoring requires two essential features: autonomous operation, and real-time analysis with minimal delay. More precisely, because of the unpredictable nature of sudden event onset, autonomous operation must be ensured, such that user interaction and subjectivity is eliminated in the data analysis. Although autonomous data acquisition has been extensively explored and implemented in full-scale applications, subsequent data analysis is generally performed with user interactions. For example, the widely used output-only system identification approach, NExT-ERA, requires the users to not only specify the minimal model order by checking the dominant singular values in singular value decomposition, but also distinguish the genuine mode from noise modes using stabilization diagram (Nagayama & Spencer, 2007). In practical, even if the genuine modes are obtained, visual examination of corresponding mode shapes is usually needed to examine and identify the modes finally. On the other hand, real-time analysis is important to save maintenance crew enough time to make informed decisions and emergency response upon the onset of sudden events. Though some autonomous strategies have already been developed and implemented in WSSNs, most of them need to wait and analyze a certain amount of measurement data before they can finally produce correct results. A typical example is the predictive model-based diagnostic approach developed by Whelan & Janoyan (2009) to work with a real-time lossless wireless sensor network. In this approach, a baseline state-space model is utilized for forward prediction of time history data without user interaction. The difference between prediction value and real measurement data is then used to make a damage index for in-service structural diagnosis. However, every time the damage index value is updated, a data set of 90-s measurement is required, resulting in a certain delay. In sum, an efficient online structural condition assessment for sudden-event monitoring remains elusive.

In addition, real-time data visualization is another desired feature for rapid condition assessment of the structure under sudden events. Though this technique has been extensively applied in tethered monitoring system, very few efforts have been made for wireless counterparts.

5.2 Real-time Data Acquisition Using Next-generation Wireless Smart Sensors

To address the challenge of response latency due to data transmission in WSSNs, a high-throughput live streaming framework is developed, leveraging the next-generation wireless smart sensor platform, Xnode. More specifically, preemptive multitasking is adopted to address the scheduling conflicts between sensing task and radio transmission in each sensor node, and adaptive Time Division Multiple Access (TDMA) is designed to minimize the radio inference between multiple sensor nodes for lossless data transmission over long-duration measurement.

5.2.1 Preemptive Multitasking Application

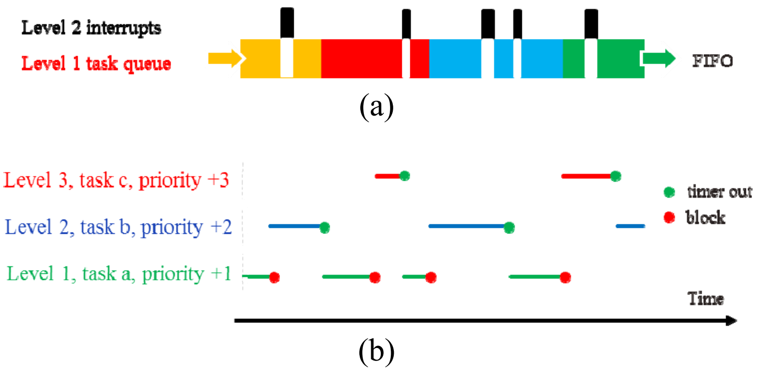
As discussed in Section 4.3.3, event-driven operating systems (e.g., TinyOS) employed in most wireless smart sensors impose the limits for the application of WSSNs. Particularly, lessons learned from experience of the Imote2 deployments reveal two major limitations:

(1) Only two levels of execution exist in TinyOS, tasks and events. Tasks are executed in a First In, First Out (FIFO) manner. Therefore, real-time applications are difficult to realize, because critical tasks may be delayed by execution of preceding tasks. Besides, non-deterministic delays in task execution are inevitable in some cases.

(2) TinyOS uses a split-phase system. Long-running operations must be divided into multiple pieces, and each piece should be carefully scheduled. The programming efforts are significant, especially for real-time applications.

These two main issues have severely limited the development of real-time data acquisition. Particularly, the constraints in OS make it extremely difficult to address the scheduling conflicts between sensing operation and radio transmission, and hence impose the limits to achieve high-throughput real-time data acquisition.

To address these issues, preemptive multitasking is considered in developing the software framework. Using preemptive multitasking, components of a specific application are well-organized into multiple tasks with different priorities. A scheduler is used to manage task sequencing at run time, and different tasks execute or block on the basis of their priorities. Specifically, tasks with lower priorities will be suspended to give way to important tasks and later be resumed after the important tasks are completed or blocked. This scheduling flexibility not only significantly facilitates complex real-time SHM applications, but also simplifies code design and reduces software development complexity. The illustration of preemptive multitasking and FIFO tasking is shown in Figure 5.1. The software framework is developed on the basis of a real-time operating system, FreeRTOS, because it provides strong support for preemptive multitasking and has various advantages, such as open source nature, portable C-language implementation, and high degree of configurability.


Level 2 interrupts
Level 1 task queue
FIFO
Level 3, task c, priority +3
Level 2, task b, priority +2
Level 1, task a, priority +1
timer out
block
Time

(a)

(b)

Figure 5.1 Comparison of two scheduling schemes: (a) FIFO tasking (b) preemptive multitasking.

In real-time data acquisition, sensing and radio transmission must be executed simultaneously using the same microprocessor; therefore, preemptive multitasking strategy is applied to address the scheduling conflicts between them. Figure 5.2 shows the structure of real-time data acquisition in preemptive multitasking framework in each sensor node. Four tasks are defined, including Application Task, Receiving Task, DataSend Task, and Sensing Task. They are assigned with different priorities, based on their latency tolerance. In particular, the Application Task is utilized to initiate the operation of sensor nodes and monitor the behavior of other tasks. If unexpected hanging occurs, the task will reset the node. Large delay is allowed in these operations, so the task priority is the lowest in the application. The Sensing Task manages the data collection process from sensors to measure structural response. It has the highest priority, because any delay of this task will result in loss of measurement data. The DataSend Task, with lower priority, is aimed to send the data which has been processed in the Sensing Task. The transmission time of a single data packet is actually larger than the sampling interval of raw data in the Xnode. Instead of splitting transmission operation carefully among multiple sampling intervals if using TinyOS, the preemptive multitasking can avoid the interruption of the sensing process with reduced programming efforts and more efficient usage of time schedule. In addition, the Receiving Task with lower priority is defined to receive potential message or commands from the base station. In addition, another conflict is involved in low-level drivers for sensing operation and radio transmission. More precisely, the radio driver and sensor driver share the same Serial Peripheral Interface (SPI) bus, when they handle interrupts during real-time data acquisition. To resolve the potential SPI conflicts, a mutex tool in FreeRTOS is utilized to coordinate the sensor driver and radio driver, ensuring that each SPI read/write operation is atomic.

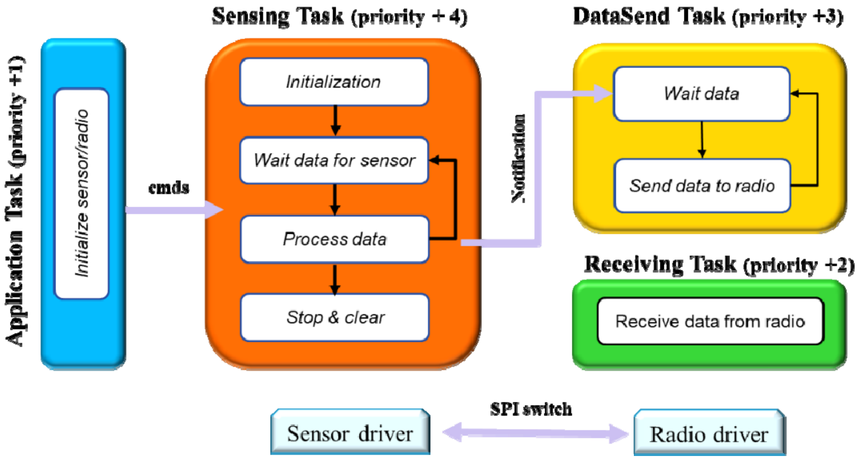


Figure 5.2 Structure of real-time data acquisition in preemptive multitasking.

5.2.2 Adaptive Time-division Multiple Access Protocol

To allow multiple sensor nodes to transmit data back to the base station in real time, the common TDMA protocol is implemented. As illustrated in Figure 5.3, sensor nodes are allocated with different time slots to transmit measurement data, such that only one sensor node can transmit data at a time. As a result, the radio interference is minimized, and high packet reception rate can be achieved. Note that, in the development of the real-time data acquisition framework for Xnodes, the packet for transmission contains 16-bit node ID, 32-bit data index for tracking data loss, 32-bit timestamp for accurate reconstruction of data, and three channels of 32-bit data in float; the packet payload is 22 bytes. Furthermore, Linderman et al. (2013) developed a staggered TDMA scheme, which can achieve higher throughput at the cost of short latency. The data acquisition built on the staggered TDMA is named as near-real-time data acquisition. In this data acquisition, multiple samples are buffered and transmitted at the same time. Though the payload size for each packet is increased, the time for packet transmission and processing only slightly increases. Hence, the throughput is significantly increased. Similar to real-time data acquisition, the packet for transmission contains the node ID, data index, timestamp, and three channels of N-buffered 32-bit data in float; the packet payload of 3-buffered samples is 46 bytes. Most SHM applications are less sensitive to real-time constraints, allowing a certain delay for response collection and analysis, and hence, near-real-time data acquisition is applicable herein.

Before implementation of TDMA scheme, the time for three main operations must be examined, as described in Linderman et al. (2013): (1) processing data in the sensor node, which includes reading data from sensor driver, resampling data at a user-specified rate, and time stamping; (2) sending data from the sensor node to the base station, which includes the preparation of packet for sending, radio channel assessment, and packet transmission in radio driver; (3) processing data in the base station, which includes radio packet handling, extracting data from the packet, and associated data post-processing. The performance of real-time data acquisition highly depends on the time consumed for all the three operations. Particularly, the combination of first two values determines the required minimal value for each time slot while constructing TDMA. A timing analysis was carried out to estimate the time required for each of the three operations. The test results are summarized in Table 5.1 and compared with the results using Imote2 from the paper (Linderman, et al., 2013). As can be seen in the table, the time for sending data and processing in the base station is much smaller than the that of Imote2s; they also have negligible variations, which presenting the potential to achieve high-throughput data acquisition. In addition, though the number of samples for transmission increases from 1 to 3, the total time for data acquisition is only increased by less than 2 times, demonstrating that near-real-time data acquisition can achieve much higher throughput than that of real-time data acquisition.

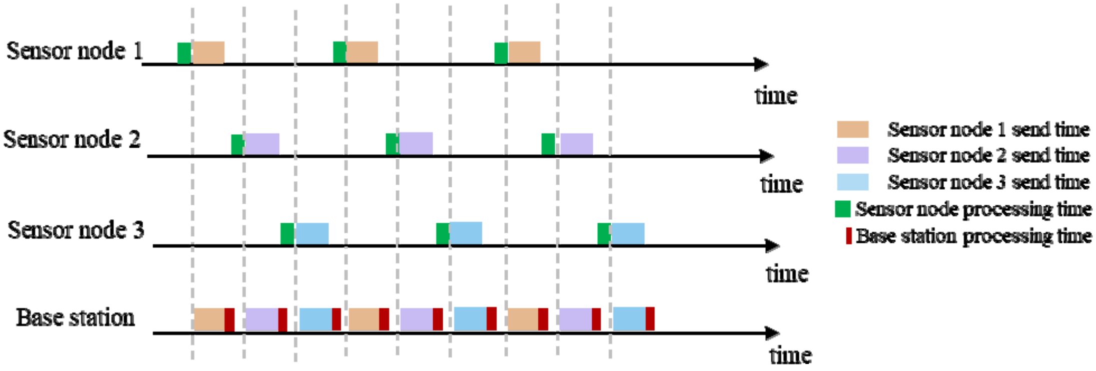


Figure 5.3 Time-division multiple access scheme and time for main operations.

Table 5.1 Timing analysis for three operations in WSSN (ms).

Mechanism	Time	Process in sensor node		Send time		Process in base station		Total time	
		Xnode	iMote2	Xnode	iMote2	Xnode	iMote2	Xnode	iMote2
Real-time	97th	0.60[2]	0.50	1.30	6.55	0.08	1.5	1.98	8.55
	Mean	0.60	0.37	1.30	4.17	0.08	1.27	1.98	5.81
	Std	0.0009	0.09	0.0014	1.48	0.0008	0.20	0.0031	1.77
Near-real-time[3]	97th	0.60[2]	0.40	2.18	7.7	0.16	2.3	2.94	10.4
	Mean	0.60	/	2.18	/	0.16	/	2.94	/
	Std	0.0009	/	0.0015	/	0.0009	/	0.0033	/

[1] the Imote2 data is obtained from the paper (Lindermen et al., 2013).

[2] it depends on sampling frequency, the data listed here is for 100Hz measurement

[3] the test is based on 3-sample-buffer approach as a representative example.

For real-time data acquisition, the adaptive TDMA scheme is designed as shown in Figure 5.4, on the basis of timing analysis results. More precisely, on the base station, each sampling interval is divided into several time slots, each of which is set for a data sample to be transmitted from a sensor node and then processed in the base station. Additional spare time is reserved in case the data transmission has a random delay. On each sensor node, a timer with high priority is created to schedule the sending process periodically. In most cases (Scenario 1 in Figure 5.4), when the timer fires, the data processing is already completed and ready to be sent, the sensor node transmits the data sample as scheduled. In some cases, the data processing is completed later than the scheduled time due to the random variation of the processing time. To compensate the processing time variation, a short allowable time slot is allocated and make the TDMA adaptive. If the data processing is finished within the specific time slot (Scenario 2 in Figure 5.4), the sample will be postponed and still be able to be sent without introducing radio interference to other sensor nodes. If the delay of data processing time is larger than the allowable time slot (Scenario 3 in Figure 5.4), the corresponding data transmit will be cancelled directly.

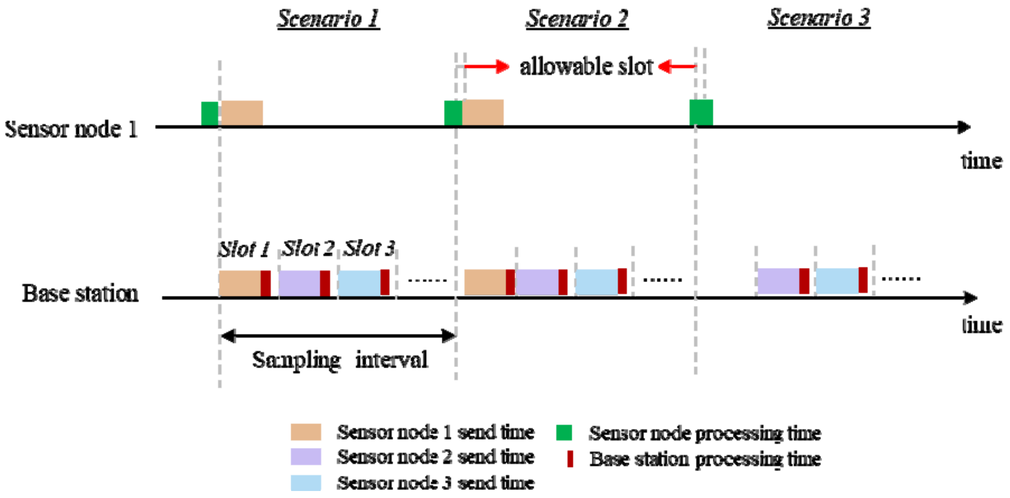


Figure 5.4 Adaptive TDMA scheme for real-time data acquisition.

For near-real-time data acquisition, the adaptive staggered TDMA scheme is designed as shown in Figure 5.5. The concept of TDMA is very similar to that designed for real-time data acquisition. Assume each packet has a buffer of N samples for better illustration. On the base station, time for N sampling intervals is divided into several time slots, each of which is set for a packet containing N data samples to be transmitted from a sensor node and then processed in the base station. In near-real-time scenario, the response latency is not strictly defined, allowing the delay of data transmission over several sampling intervals. Therefore, the adaptive strategy described in Figure 5.4 is not essential any more. However, because of transmitting larger packet, the transmission time of a single packet from each node increases. Considering the random variation of processing time during live streaming, the transmission time between multiple nodes is more likely to overlap; it may result in packet loss, which is a more critical concern herein, because the loss of a packet corresponds to the loss of multiple data samples. The likelihood of collision becomes more serious over long-duration measurement. More precisely, a one-shot clock synchronization is only viable for several minutes, after which the TDMA schedule will be disrupted by clock drift. Therefore, a new adaptive strategy is designed for near-real-time data acquisition to minimize potential packet collision over long-duration measurement. More specifically, the data transmission is fully independent of sensing process and is scheduled by a timer from FreeRTOS. The local clocks which control the timer, are not only synchronized prior to sensing, but also resynchronized periodically (e.g., every 60s) to keep the staggered TDMA stable. During clock resynchronization, the base station broadcasts multiple timestamping beacons to sensor nodes which stop data transmission but still continue sensing process. As a result, the clock drift and offset are updated in each sensor node, and the TDMA schedule is refreshed accordingly. Generally, to accurately estimate the linear clock drift, clock synchronization prior to sensing takes much time (e.g., 30-s in the Imote2s), which is not practical for sudden-event monitoring. In contrast, the time for initial clock synchronization can be reduced to 1-2s, because the proposed strategy does not require high-precision clock synchronization prior to sensing.

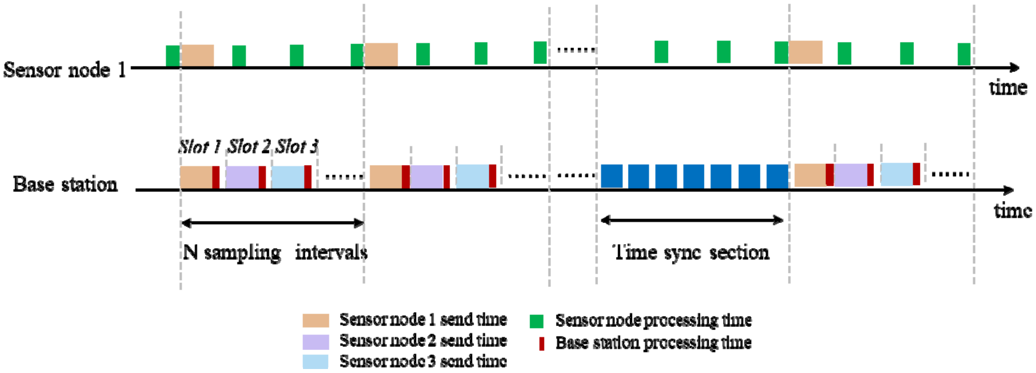


Figure 5.5 Adaptive staggered TDMA scheme for near-real-time data acquisition.

5.2.3 Application Flowchart

To realize a robust real-time data acquisition within a network of wireless smart sensors, a complete application framework is required to synchronize and coordinate multiple leaf nodes. The basic idea is similar to the paper (Linderman et al., 2013). Figure 5.6 shows the flowchart of the entire application. At the beginning, the first command is sent from the base station to all the designated sensor nodes to initialize the application. This command contains user-specified sensing parameters, including sampling frequency, sensor channels, measurement time, and time to start sensing. Then, global timestamps are broadcasted to synchronize the local clocks on sensor nodes. Afterwards, the base station records the number of responsive sensor nodes, and then specifies TDMA schedule for each responsive node. For example, the first responsive node in the user-specified node list will be assigned to take the first time slot for data transmission in each sampling interval. Another command is sent, containing the schedule information, to all the responsive sensor nodes to start both sensing and scheduling of data transmission. Finally, sensor nodes collect data samples and transmit data in real-time until the sensing time expires. For near-real-time data acquisition, global timestamps are broadcasted periodically, and sensor nodes performed resynchronization periodically. The two commands are transmitted on the basis of reliable communication to ensure their successful delivery, whilst the sensor data transmission is carried out using unreliable communication to reduce the workload of radio by eliminating acknowledgements and resends. In this unreliable communication, the potential data loss is negligible, because the radio interference is considered as the main source of data loss and it is already minimized by the TDMA scheme.

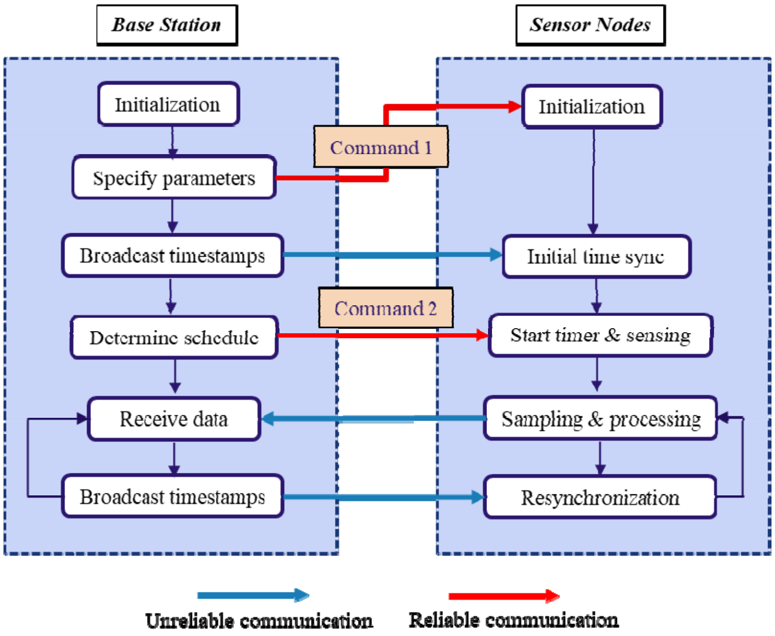


Figure 5.6 Real-time/near-real-time data acquisition application flowchart.

5.2.4 Application Performance

To evaluate the proposed framework of real-time/near-real-time data acquisition, multiple repeated lab tests were conducted involving several sensor nodes and one base station, as shown in Figure 5.7. To mitigate the radio interference, the radio power level is reduced, and the sensor nodes are placed at a certain distance. The framework performance is summarized in Table 5.2. In particular, using a single sensor node, the proposed framework can achieve real-time data acquisition at a sampling rate of up to 500Hz, which is more than 4 times over the TinyOS-based application using an Imote2. The corresponding maximum throughput is also increased by over 6 times. If the network size is increased to four, the maximum sampling rate and throughput decreases. In addition, packet reception rate (PRR) offers an indicator of data loss during acquisition. Note that the packet reception listed in the table corresponds to data sample loss. The range of PRR for 5 repeated tests is recorded; the observed minimal data loss demonstrates the excellent performance of the proposed framework. Likewise, the performance of near-real-time data acquisition framework is examined using the same test setup, in which 6-sample buffered staggered TDMA is tested. As can be seen, by buffering samples, the throughput is increased significantly, which is approximately 4 times larger than that using Imote2s. Furthermore, with the aid of adaptive strategies, the framework is kept stable over long-duration measurement, achieving 100% PRR for over 10 minutes.

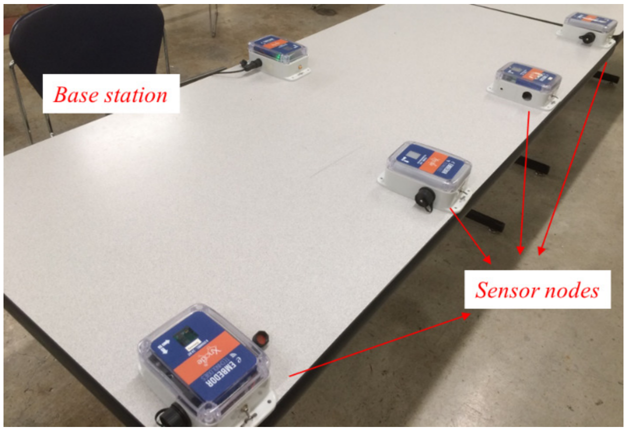


Figure 5.7 Test set-up for real-time/near-real-time data acquisition.

Table 5.2 Performance of real-time/near-real-time data acquisition framework.

Mechanism	Network size	Max sampling rate (Hz)	Max data throughput (kbps)	Packet reception rate
Real-time	1	500	48	100%
	2	200	38.4	97.2%~97.7%
	4	100	38.4	97.4%~98.2%
Near-real-time	6	200	115.2	100%
	12	100	115.2	100%

5.3 Online Condition Assessment

To address the challenge of lack of support for real-time data management, two online condition assessment strategies are developed, including interstory drift estimation and sudden damage detection. Both strategies suffice for the rapid condition assessment, in terms of two critical features: autonomous operation and real-time analysis with minimal delay. More specifically, the first strategy estimates the interstory drifts of buildings under earthquakes or strong winds in real-time and provides early story condition assessment autonomously. The second strategy support data analysis and identify the location and onset time of structural damage in real-time.

5.3.1 Interstory Drift Estimation Using Acceleration Record

This section describes the use of a FIR filter based on a minimization problem to estimate interstory drifts of buildings from accelerations measurements at different locations. First, reference-free displacement estimation is described, which is developed on the basis of a FIR filter. Then, the strategy of intertory drift estimation is proposed utilizing the FIR filter that performs double integration and high-pass filter simultaneously. Finally, numerical validation is carried out to demonstrate the performance of the proposed strategy. Note that, this study is only focused on dynamic displacements, because double integration of acceleration cannot obtain residual displacements which can be potentially large in significant nonlinear responses.

(1) Reference-free displacement estimation

The following minimization problem with Tikhonov regularization was proposed by Gomez et al. (2018) recently to estimate displacements from measured accelerations

$$\Pi(u)=\frac{1}{2}\int_T\left[\frac{d^{n-2}}{dt^{n-2}}\left(\frac{d^2u}{dt^2}-\bar{a}\right)\right]^2dt+\frac{1}{2}\beta^n\int_T u^2dt \tag{5.1}$$

where T is the time window $t_1<t<t_2$, u is the estimated displacement, $\bar{a}$ is the measured acceleration, $n \geq 2$ is the order of the function, and $\beta>0$ is the Tikhonov regularization factor.

Applying variational calculus, the following ordinary differential equation is obtained

$$\frac{d^{2n}u}{dt^{2n}}+(-\beta)^n u=\frac{d^{2n-2}\bar{a}}{dt^{2n-2}},\quad t_1<t<t_2 \tag{5.2}$$

Boundary conditions for this equation are prescribed displacements and its derivative at times t_1, t_2. Although, they have an important impact close to the ends of the interval, they are not readily available. Moreover, the influence of the boundary conditions is smaller towards the center of the time window; considering windows centered at different time points minimize the boundary condition effects for each point. The frequency response of the previous equation and the accuracy function are given by

$$H_{u\bar{a}}(\omega)=-\frac{\omega^{2n-2}}{\omega^{2n}+\beta^n} \tag{5.3}$$

$$H^{\text{accu}}(\omega)=\frac{\omega^{2n}}{\omega^{2n}+\beta^n} \tag{5.4}$$

where the regularization factor β can depend on the target frequency by means of the following expression

$$\beta = \sqrt[n]{\frac{1-\alpha_T}{\alpha_T}}(2\pi f_T)^2 \tag{5.5}$$

The target accuracy should take values between 0 and 1 not inclusive.

Due to the instability of the filter in the continuous-time representation, a discrete FIR filter, with generalized linear phase and type I, is considered. A vector of coefficients **c** of length 2k+1 represents the FIR filter. The relation between the measured acceleration and the estimated displacement is

$$u(t) = (\Delta t)^2 \sum_{p=-k}^{k} c_{k+1+p}\bar{a}(t + p\Delta t) \tag{5.6}$$

where the coefficients of filter are given by

$$c_{p+k+1} = -\frac{f_s}{2\pi^2}\int_0^{f_s/2} \frac{f^{2n-2}}{f^{2n} + \lambda^{2n} f_T^{2n}} \cos(2\pi p f \Delta t)\,df \tag{5.7}$$

and the number of points of the filter is a function of the normalized window length N_w, the target frequency, and the sampling frequency by the following equation

$$k = N_w \frac{f_s}{2f_T} \tag{5.8}$$

To minimize Gibbs' phenomenon, the time-window should be such that the impulse response function in the time-domain ends at zero.

(2) Interstory drift estimation using FIR filter

The FIR filter described in previous section is utilized to estimate interstory drift. As illustrated in Figure 5.8, time-synchronized acceleration digital records with the same sampling frequency at two consecutive floors are used to obtain relative acceleration, Δa, which is processed through the FIR filter designed for the structure, and finally interstory drift estimation is obtained.

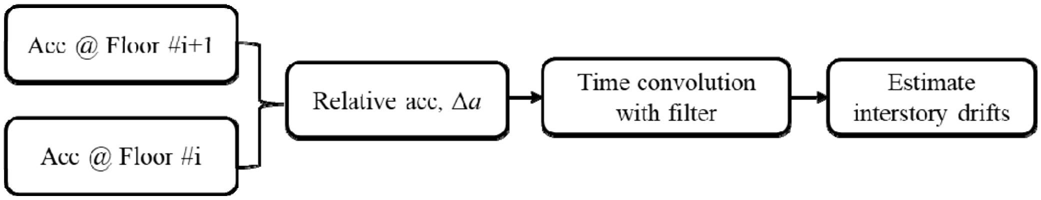


Figure 5.8 Flowchart of early estimation of story conditions.

More detailed mathematical discussion is presented herein. A vector of coefficients **c** of length $2k + 1$ represents the FIR filter. The relation between the measured acceleration and the estimated displacement at the two locations are

$$u_i(t) = (\Delta t)^2 \sum_{p=-k}^{k} c_{k+1+p}\bar{a}_i(t + p\Delta t) \tag{5.9}$$

$$u_{i+1}(t) = (\Delta t)^2 \sum_{p=-k}^{k} c_{k+1+p}\bar{a}_{i+1}(t + p\Delta t) \tag{5.10}$$

where the coefficients of filter are given by

$$c_{p+k+1} = -\frac{f_s}{2\pi^2}\int_0^{f_s/2} \frac{f^{2n-2}}{f^{2n} + \lambda^{2n} f_T{}^{2n}} \cos\left(2\pi p f \Delta t\right) df \tag{5.11}$$

and the number of points of the filter is given by Eq. (5.8).

Note that the vector of coefficients is the same for both locations because they are part of the same structure and the records have the same sampling frequency. Consequently, the interstory drift is given by the difference of the displacements of the two floors

$$\theta_i\left(t\right) = u_{i+1}\left(t\right) - u_i\left(t\right) = \left(\Delta t\right)^2 \sum_{p=-k}^{k} c_{k+1+p}\left[\bar{a}_{i+1}\left(t + p\Delta t\right) - \bar{a}_i\left(t + p\Delta t\right)\right] \tag{5.12}$$

$$\theta_i\left(t\right) = \left(\Delta t\right)^2 \sum_{p=-k}^{k} c_{k+1+p} \Delta\bar{a}_i\left(t + p\Delta t\right) \tag{5.13}$$

where $\Delta\bar{a}_i$ is the relative acceleration of one floor to the next floor,

$$\Delta\bar{a}_i = \bar{a}_{i+1} - \bar{a}_i \tag{5.14}$$

Once the coefficients of the FIR filter are applied, the proposed approach to obtain the drift requires only the multiplication of two vectors to estimate the interstory drift in each time point. The filter is not causal and it requires almost half of the window in the future to estimate the current point, which means that the estimation is obtained half the time window length after each point. It would typically lead to a lag in the estimation of less than a second.

Interstory drift in buildings has been widely recognized as an important quantity assess structural performance and damage under seismic events or strong winds. This strategy can measure dynamic interstory drifts of buildings from measured acceleration. For large enough events, the response will contain large residual deformations that cannot be recovered, however, the dynamic interstory drift can still be useful for rapid assessment of tall buildings. As referred to the paper (Çelebi, 2013), if the interstory drift ratio is larger than 0.2%, the associated story is considered to be beyond elastic range; if it is larger than 0.8%, the story is considered to be severely damaged.

(3) Numerical validation

A benchmark for a 9-story linear shear building subjected to ground motions (Xu et al, 2017) is considered to test the accuracy of the method. The mass of the first floor is 505 Ton, the masses of the second to eight floors are 495 Ton, and the mass of the roof is 535 Ton. The stiffnesses for the first to the last floor are 600, 578, 544, 502, 453, 397, 332, 256, and 162 MN/m, respectively. The damping ratio is assumed to be 2% for each mode. Two different ground motions are considered: El Centro (EC) earthquake record and an artificial earthquake using non-stationary Kanai-Tajimi (NSKT) model with the following properties: $\omega_{\mathrm{g}} = 12$ rad/s, $\zeta_{\mathrm{g}} = 0.3$, $S_0 = 0.02$ m^2/s^3, and $e\left(t\right) = 4\left(e^{-0.1t} - e^{-0.2t}\right)$.

The excitation is simulated using Simulink with a sampling rate of 1000 Hz, then the records are decimated to a sampling rate of 100 Hz. The interstory drifts are measured at all stories as a reference for comparison. The absolute floor accelerations are measured at each floor and at the base, and the acceleration measurements are polluted with a zero-mean Gaussian noise with standard deviation equal to 5% of the maximum RMS acceleration. The proposed method is applied to estimate interstory drift from measured acceleration with the following parameters: $f_s = 100$ Hz, $n = 4$, $f_T = 0.8$ Hz, $\alpha_T = 0.99$, and $N_w = 5.223$. As a comparison, the measured total interstory drift is considered; in this case, interstory drifts have small pseudo-static components and there is no need to extract the dynamic component.

Figure 5.9 and 5.10 show the comparison of the interstory drift for floors 1 and 6 and for ground motions EC and NSKT, respectively. As these figures show, the estimated and the exact values agree well for all time steps.

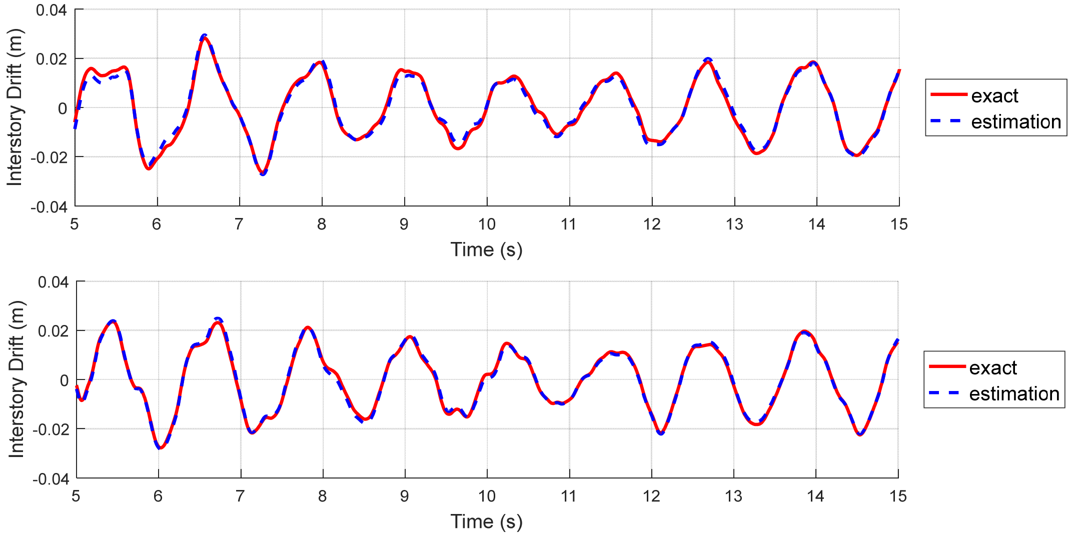


Figure 5.9 Interstory drift comparison with EC ground motion for (a) first story, (b) sixth story.

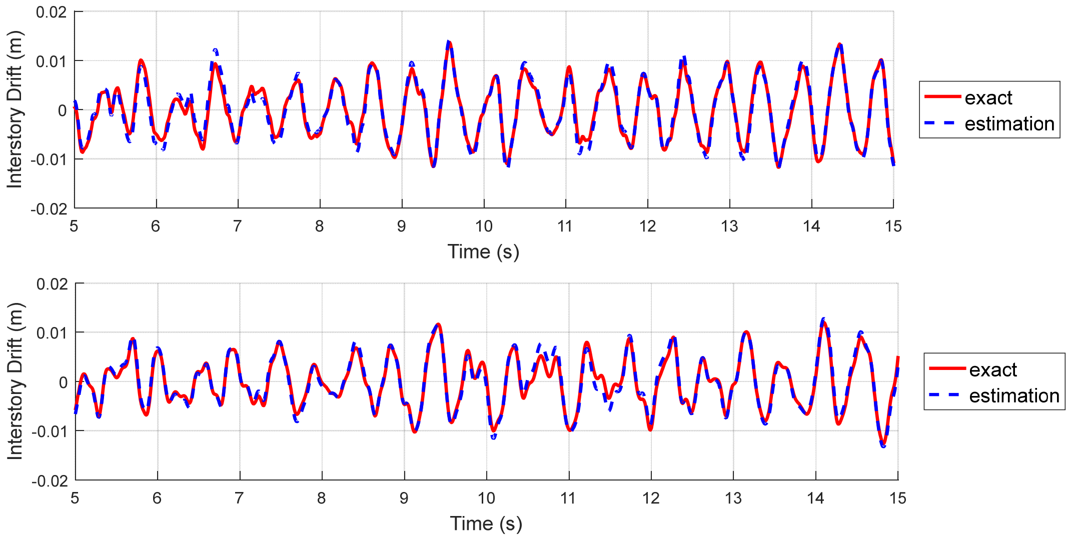


Figure 5.10 Interstory drift comparison with NSKT ground motion for (a) first story, (b) sixth story.

Additionally, to assess the accuracy of the method two types of errors are computed. Amplitude error is defined by the difference between the maximum estimated response and the maximum exact response divided by the maximum exact response. RMS error is defined by the RMS of the difference between the estimated response and the exact response divided by the maximum exact response. Table 5.3 shows the amplitude and RMS errors for both excitations. As these values show, the errors are relatively small even though large Gaussian noise was used to pollute the acceleration records. Estimations for NSKT ground motion yields larger errors than the estimation for EC, because relatively large response occurs in one floor only and the records are polluted using noise with amplitude in terms of the RMS of the maximum response. In real tests, the errors in the measurement are expected to be smaller for typical vibration sensors.

Table 5.3 Errors in the proposed method for both excitations.

Story	El Centro		Non-stationary Kanai-Tajimi	
	Amplitude Error (%)	RMS Error (%)	Amplitude Error (%)	RMS Error (%)
1	4.89	3.42	6.97	6.65
2	2.03	3.75	7.85	8.41
3	4.77	4.30	19.08	8.33
4	4.73	3.47	7.97	10.69
5	1.43	4.29	4.46	8.48
6	1.17	3.31	5.39	6.57
7	1.02	2.83	3.13	5.60
8	5.38	2.89	0.05	4.10
9	0.79	2.87	1.13	3.23

5.3.2 Sudden Damage Detection Using Acceleration Record

This subsection describes the use of wavelet transform (WT) and independent component analysis (ICA) to identify the location and time of the damage occurrence in real-time. First, WT-ICA algorithm (Yang & Nagarajaiah, 2014) is introduced, and associated challenges for online condition assessment are also discussed. Then, the online strategy of sudden damage detection in real-time is proposed, which is an upgraded version of the WT-ICA algorithm. Finally, numerical validation is carried out to demonstrate the performance of the proposed strategy.

(1) WT-ICA algorithm for damage instant detection

In discrete wavelet transform, a signal *f*(t) can be represented through multi-resolution analysis,

$$w_{m,n} = \int_{-\infty}^{\infty} f(t)\psi^{*}_{m,n}(t)dt \tag{5.16}$$

$$v_{m,n} = \int_{-\infty}^{\infty} f(t)\phi^{*}_{m,n}(t)dt \tag{5.17}$$

where $w_{m,n}$ and $v_{m,n}$ are detail and approximation coefficients, respectively. $\psi_{m,n}(t)$ and $\phi_{m,n}(t)$ are the wavelet kernel and scaling function at the scale level of *m*, respectively, which can be expressed as,

$$\psi_{m,n}(t) = \frac{1}{2^m}\psi(\frac{t}{2^m} - n) \tag{5.18}$$

$$\phi_{m,n}(t) = \frac{1}{2^m}\phi(\frac{t}{2^m} - n) \tag{5.19}$$

where *n* is the translation parameter. In turn, the $w_{m,n}$ and $v_{m,n}$ can be used to reconstruct the signal *f*(t). Assuming that the signal *f*(t) can be decomposed into M levels, and the signal can be reconstructed as follows,

$$f(t) = V_M(t) + \sum_{m=1}^{M} W_m(t) \tag{5.20}$$

where $V_M(t)$ is approximation at level M; and $W_m(t)$ is detail at level *m*. They can be expressed as,

$$V_M(t) = \sum_n v_{M,n}\phi_{M,n}(t) \tag{5.21}$$

$$W_m(t) = \sum_n w_{m,n}\psi_{m,n}(t) \tag{5.22}$$

Chen et al. (2014) found that, if using high-rate measurement, signal discontinuity can be observed, when the structural stiffness is decreased suddenly. Though this phenomenon is not identifiable at low-rate measurement, the discontinuity can still be revealed as pulse-like features at a specific detail, $W_m(t)$, at level *m*. Accordingly, structural damage can be captured using discrete wavelet analysis. However, the characteristics of signal discontinuity are very similar to that of signal noises to some extents. Therefore, the pulse-like features can be easily destroyed by a certain level of noises. This challenge is later successfully addressed by using WT-ICA with appropriate assumptions (Yang & Nagarajaiah, 2014), as discussed below.

The wavelet-domain responses can be considered as mixtures containing a pulse-like signal and other responses. The ICA, which is widely used as blind source separation, can be applied to extract the pulse-like signal from the mixtures. More precisely, the reconstructed wavelet detail at certain level which contains the most information of damage can be represented as,

$$\boldsymbol{W} = \boldsymbol{A}\,\boldsymbol{S} \tag{5.23}$$

where $\boldsymbol{W} = [W_1(t), W_2(t), \ldots W_n(t)]^T$, *n* is the number of sensors, ***A*** is the mixing matrix, and the wavelet detail at the certain level from *x*th sensor can be expressed as,

$$W_x(t) = \sum_{i=1}^{n} a_i S_i(t) \tag{5.24}$$

where $S_i(t)$ is the *i*th source signal in the mixture, and a_i is corresponding mixing weight. Based on the classical central limit theorem, ICA searches for proper de-mixing matrix to make the recovered independent source signals $R_i(t)$ as non-gaussian as possible,

$$\boldsymbol{R} = \boldsymbol{B}\,\boldsymbol{W} = \boldsymbol{B}\,\boldsymbol{A}\,\boldsymbol{S} \tag{5.25}$$

where $\boldsymbol{R} = [R_1(t), R_2(t), \ldots R_n(t)]^T$, and ***B*** is the de-mixing matrix which is an estimation of $\boldsymbol{A}^{-1}$. To find the best de-mixing matrix, entropy-based negentropy is applied as a measure of non-gaussianity, finding independent source signals with sparse representation. Finally, the recovered signal with pulse-like features is considered as an "interesting" source signal, $R_d(t)$, which contains the damage instant information.

Furthermore, the column-wise vector for $R_d(t)$, named as source distribution vector (SDV), contains the spatial information of damage location. More precisely, the element, a_{id}, in is the mixing weight of the interesting source signal $R_d(t)$ in the *ith*

sensor measurement data, and the largest value of a_{id} indicates the potential location of the damage.

(2) Sudden damage detection in real-time

The WT-ICA algorithm is very suitable for online condition assessment for sudden-event monitoring. It is not only effective and light-weight to be implemented on WSSN, but also has the significant potential to detect the damage instant in real-time. However, the WT-ICA algorithm is not autonomous, and user subjectivity is involved in two critical steps: (1) determine a certain WT scale which is observed to contain most damage information; (2) find the "interesting" source which contains an outstanding spike to identify damage.

To address the challenges listed above, an efficient strategy is developed, as illustrated in Figure 5.11. Every one second, the collected data samples at each sensor node are decomposed and reconstructed at level *k* in which starts from K_{min} in practice. The wavelet-domain responses are then processed by Fast ICA algorithm, and several independent components (IC) are obtained. To examine if it is the interesting source that contains the most damage information, an index $I(m,k)$ is calculated as,

$$I(m,k)=\max IM_{k,m}^{2}(t) \tag{5.26}$$

$$IM_{k,m}(t)=\frac{1}{5}\sum_{s=1}^{5} IC_{k,m}(t+s) \tag{5.27}$$

where $IM_{k,m}(t)$ is the mean value of the m^{th} *IC* at level *k*, obtained by passing $IC_{k,m}$ through a 5-point moving-average filter. If $I(m,k)$ is larger than a user-specified threshold, the associated $IC_{k,m}$ is considered as the suspected "interesting" source. If $I(m,k)$ for all the *IC*s are below the threshold, a next-ground of WT-ICA is performed at *k+1* level. The process continues until a suspected "interesting" source is found. Finally, within all the suspected "interesting" sources at the same wavelet level *k*, the one with maximum $I(m,k)$ is selected as the "interesting" source, *IS*. In addition, the time when maximum $I(m,k)$achieves in the *IS* corresponds to that of the damage instant. But if no suspected "interesting" sources are found even if k reaches the user-defined maximum value (K_{max}), the searching process stops, concluding that no damage occurs. In this way, the "interesting" source can be selected autonomously, and sudden damage can be detected in real-time.

Based on the experience from the spike detection in the paper (Fu et al., 2019), the threshold for spike-like interesting source is recommended to be 1000. In addition, the range of WT levels depends on the choice of wavelet kernel. Followed the study in the paper (Yang & Nagarajaiah, 2014), *k* is selected from 3 to 6, using "db10" wavelet kernel.

(3) Numerical simulation

A numerical simulation is performed to validate the performance of the proposed online strategy. Particularly, the numerical model is built on the basis of a 6-story building model at the Smart Structures Technology Laboratory, University of Illinois at Urbana-Champaign (UIUC), as shown in Figure 5.12. To get its mass and stiffness matrix, modal analysis of the 6-story building model was conducted using the peak picking method. The estimated natural frequencies of all 6 modes are, 1.406, 4.482, 7.119, 9.58, 11.4, 12.74 Hz, respectively. These natural frequencies and corresponding mode shapes are then used to obtain modal mass matrix and modal stiffness matrix, both of which are diagonal matrixes. Assume that all the floors have the same mass, which is 2.8kg and obtained by using a digital scale. Then, by leveraging modal stiffness matrix, the stiffnesses of all the floors are then estimated accordingly, which are 114.22, 114.22, 100.46, 111.46, 127.79, 136.91 kN/m, respectively. In addition, the damping ratio is assumed to be 2% for each mode. Kobe earthquake is considered as the ground motion, which has approximately 140-s duration. The excitation is simulated using Simulink. To mimic the Xnode measurement, accelerations from all the floors are first recorded at 1000Hz and then decimated to a sampling rate of 100Hz. The signal-to-noise ratio is set to be 50dB. Damage is modeled by an abrupt reduction of stiffness at a certain floor.

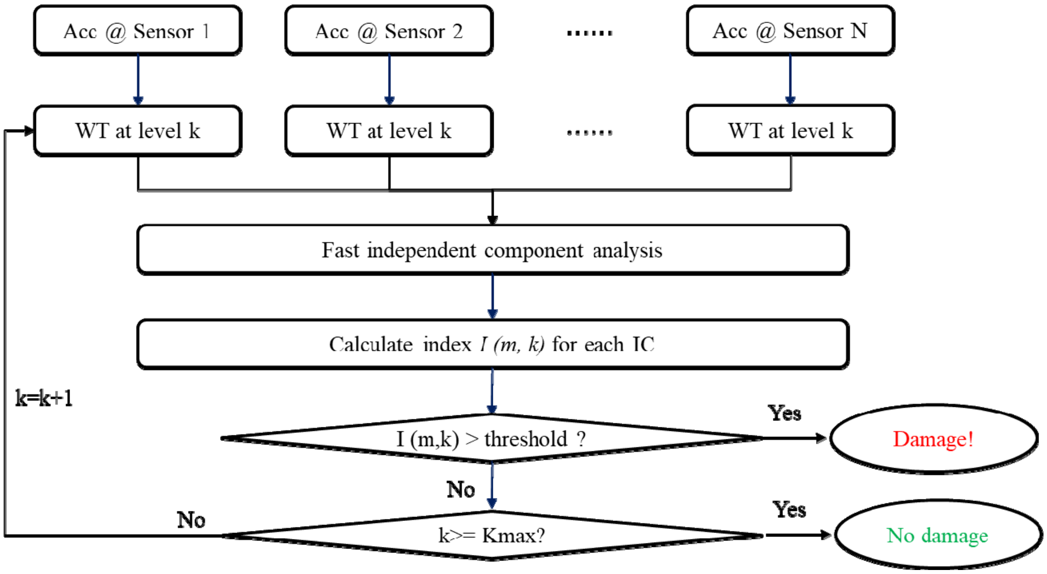
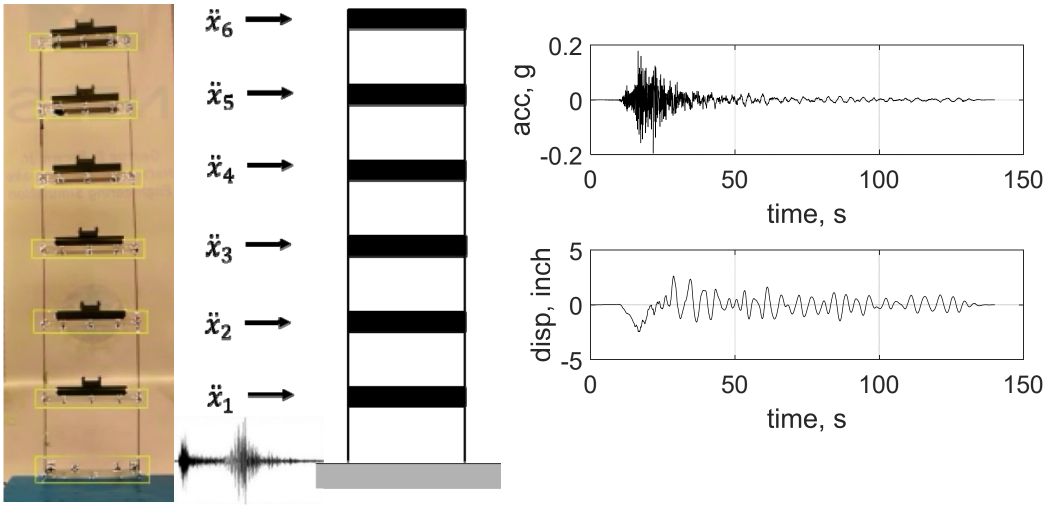


Figure 5.11 Flowchart of online sudden damage detection.



(a) (b) (c)

Figure 5.12 Test model and seismic event: (a) lab model, (b) numerical model, (c) Kobe earthquake.

Various cases are considered, as listed in Table 5.4. In particular, Case 1 and Case 2 examine the algorithm performance for different damage floors. Case 1 and Case 3 examine the algorithm performance for different damage severities. Case 1 and Case 4 examine the algorithm performance for different damage instants. Note that the main shock of the Kobe earthquake record is within first 50s, and the possibility of damage onset before 50s is high. Therefore, the damage instants are only considered for within that time range.

Table 5.4 Damage cases for performance examination of sudden damage detection.

Case	Description
1	Floor #3 has 30% stiffness reduction at 20th second
2	Floor #5 has 30% stiffness reduction at 20th second
3	Floor #3 has 10% stiffness reduction at 20th second
4	Floor #3 has 30% stiffness reduction at 30th second

To better illustrate the idea of autonomous operation of the proposed strategy, results of sudden damage detection in Case 1 are presented and discussed in detail. During simulation, the accumulative data samples from 6 floors responses are processed by db10 wavelet at a 1-s interval, following the flowchart in Figure 5.11. As an example, the intermediate results at 50 seconds are presented in Figure 5.13. The interesting source is identified successfully, when the data is processed by WT at level 3. More precisely, as shown in Figure 5.14, the second IC has the largest $I(m,k)$ which is more than 1000; the maximum value of IC2 index is achieved at 20s, which is exactly when the sudden damage occurs. In addition, the associated SDV curve for IC2 presents the potential damage location at Floor #3.

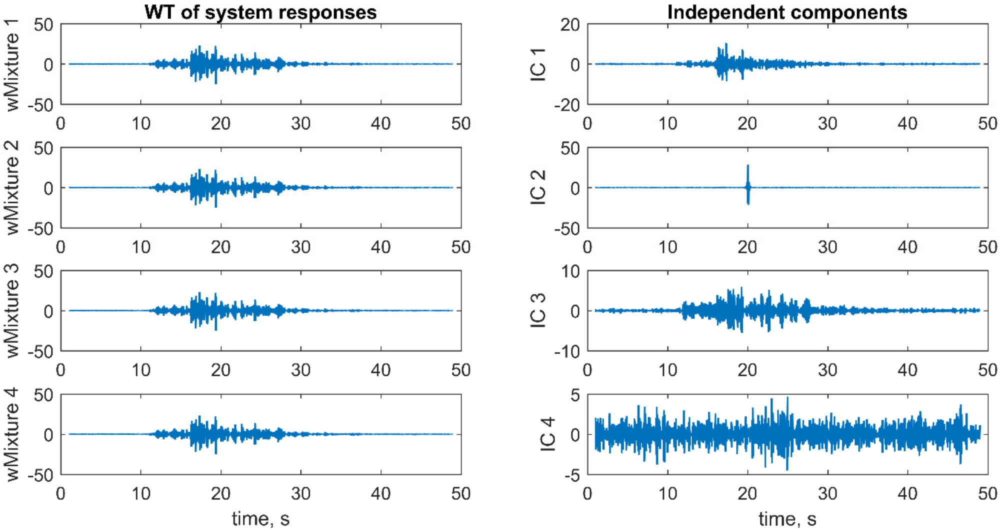


Figure 5.13 online sudden damage detection results at 50s.

The overall result for case 1 is shown in Figure 5.15a. The proposed strategy is able to capture the abrupt damage instant at 20s in real-time and successfully identify the damage location at Floor #3. Figure 5.15b-d show the results for other cases, demonstrating that the proposed strategy is efficient and effective at different damage floors, at different damage severity of even as small as 10% stiffness reduction, at different damage instants.

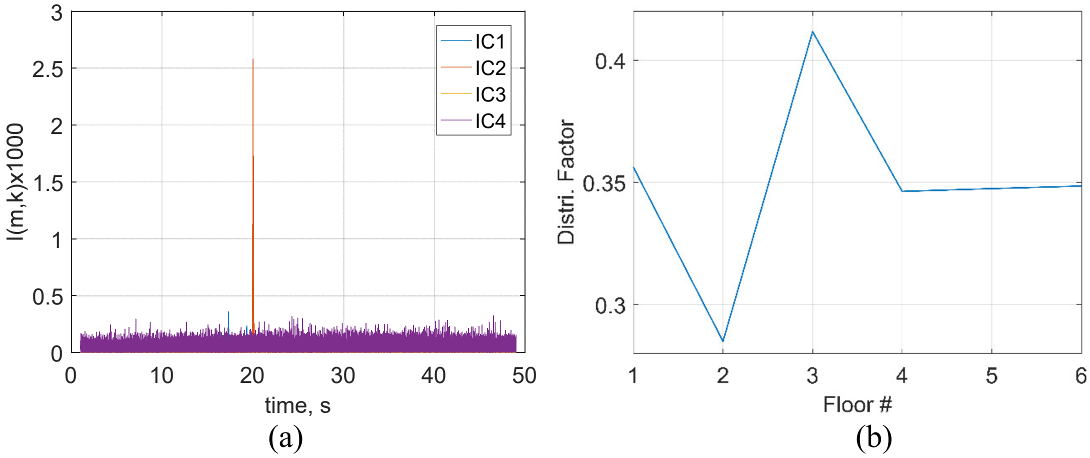


Figure 5.14 index value for ICs and SDV for IC2 at 50s: (a) index value, (2) SDV value.

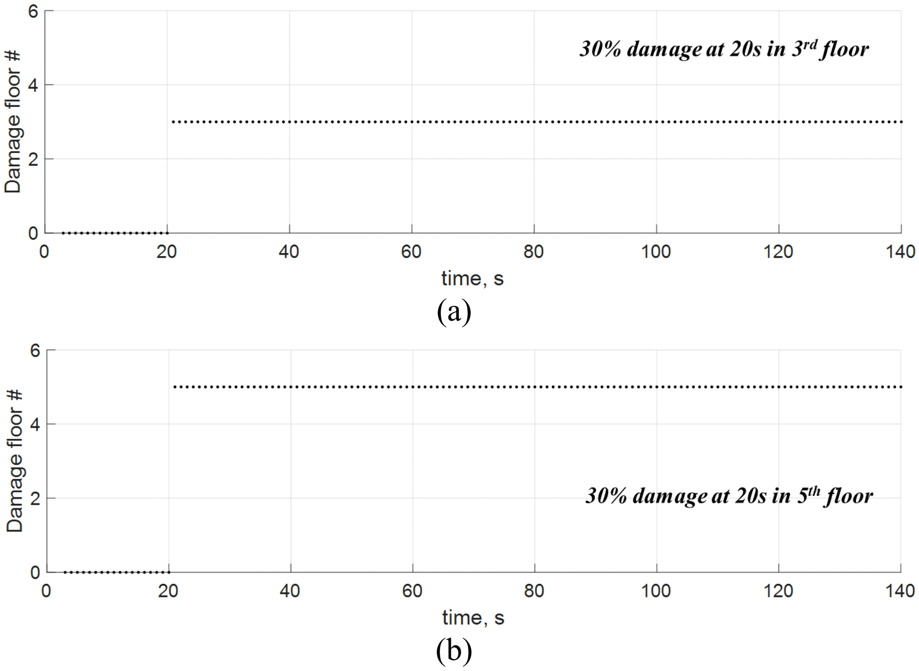
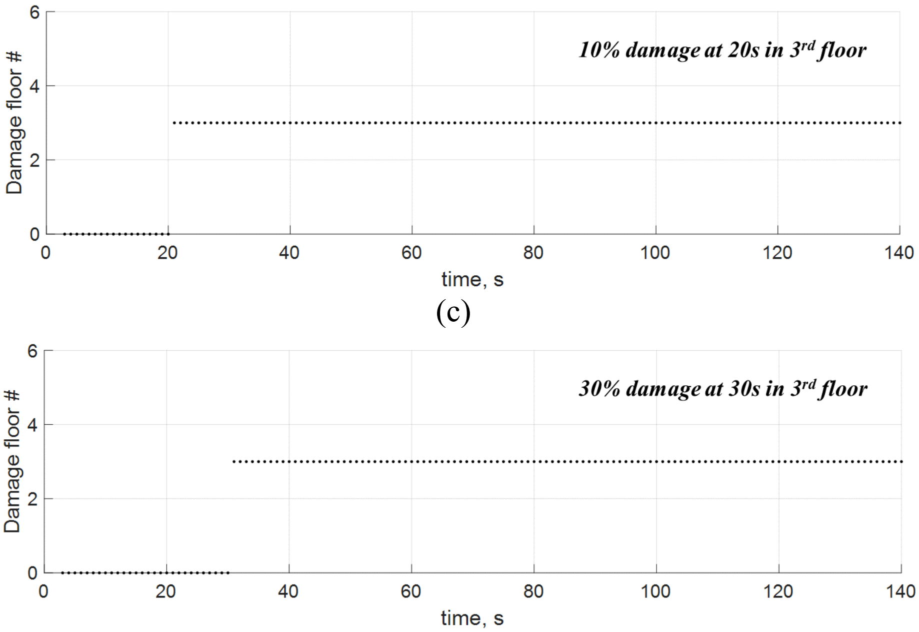


Figure 5.15 Sudden damage detection results: (a) Case 1, (b) Case 2, (c) Case 3, (d) Case 4.



(c)

(d)

Figure 5.15 (cont.)

5.4 MATLAB-based Application

To fully realize rapid condition assessment, a MATLAB-based application is developed, which consists of a graphic user interface (GUI), code performing online condition assessments, and associated data/parameters for configuration. In particular, as the core component, the GUI allows end users to interact with the wireless monitoring systems and enable real-time data visualization. As shown in Figure 5.16, the GUI contains several parts as follows,

(1) Control panel

The control panel allows users to select and run commands for WSSNs without the command line interface. It has three main sections: campaign-type monitoring, real-time data acquisition, and data processing. The campaign-type monitoring section is designed for uninstrumented structures. Before or after a sudden event, users can deploy wireless smart sensor network and use the associated commands to conduct timely structural condition assessment. The main command in this section is *RemoteSensing*, which is a fundamental distributed data acquisition application to help the coordination between base station and multiple sensor nodes during the remote sensing application. The real-time data acquisition section has two options, user-triggered and event-triggered, which is mainly designed for instrumented structures. If the sudden event is long-duration, e.g., typhoon, for which the response latency is not critical, real-time data acquisition can be started or stopped via user interference, aiming to obtain real-time condition assessment; But if the sudden event is short-duration, e.g., earthquakes, the real-time data acquisition is triggered by the structural vibration subjected to the event. In addition, the control panel has several postprocessing commands, including time history statistics analysis, power spectral density calculation, and displacement estimation. The measurement data can also be saved as .mat or .csv file for future analysis.

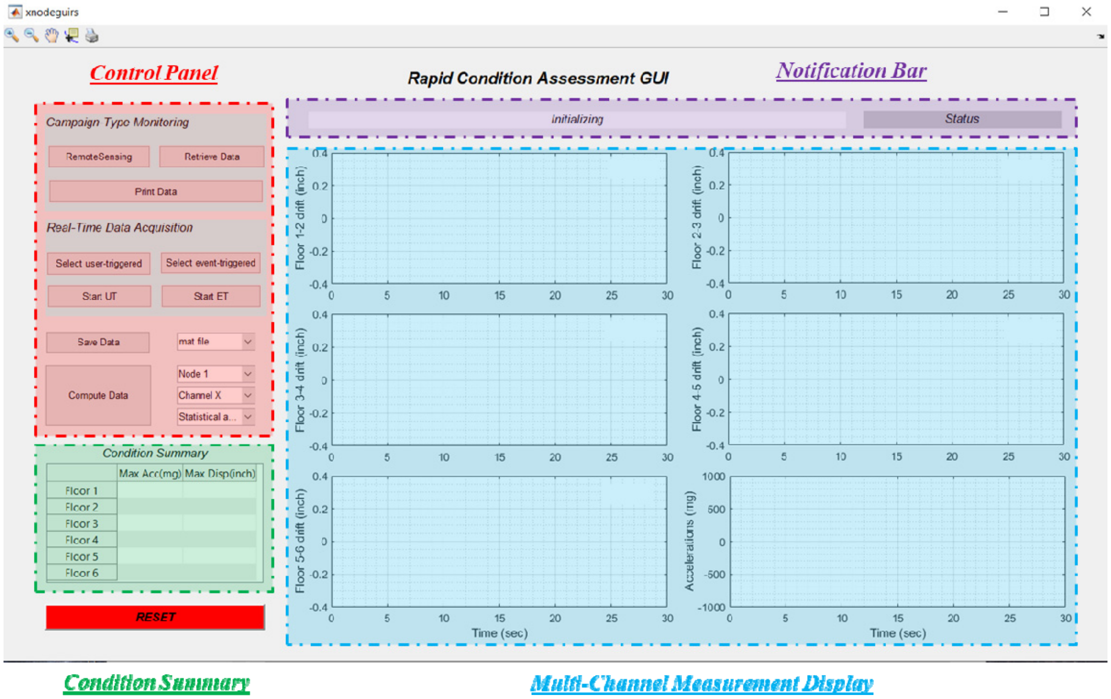


Figure 5.16 Graphic user interface of the efficient data management application.

(2) Notification bar

The notification bar tells the users about the status of the WSSN during data acquisition, including *START*, *WORKING*, and *STOP*. In addition, it shows the detailed commands lines sent from the base station to sensor nodes, such as *sending sensing parameters*, *time synchronization*, and *data request*, etc.

(3) Condition summary

After sudden event stops and data acquisition is completed, the GUI will perform preliminary data analysis and provide a summary of early structural condition estimation, such as the maximum accelerations and displacements for each sensor location in current configuration.

(4) Multi-channel measurement display

Most importantly, the GUI offers multi-channel real-time data visualization. While sensor nodes are sensing, data samples are collected and transmitted continuously in the base station. In the meantime, the GUI plots the measurement history data in real-time. In current configuration, the GUI can support 12 channels data visualization for either acceleration measurement, or displacement estimation. Note that because of the intrinsic filter latency, a delay of approximate 2s exists between acquisition and animation plotting. With the GUI's help, users can see the time history of structural vibration during sudden events and understand the structural behavior in real-time.

5.5 Summary

In this chapter, three main techniques are developed to support real-time condition assessment for sudden-event monitoring. An efficient real-time data acquisition framework is designed to address the challenge of response latency due to data transmission, leveraging next-generation wireless smart sensors. Preemptive multitasking is implemented to address the scheduling conflicts between sensing and radio transmission; adaptive TDMA is developed to minimize radio interference between sensor nodes. The framework performance is excellent, in terms of data throughput and data delivery. Furthermore, two online structural condition assessments are developed to provide strong support of real-time data management. The strategies include interstory drift estimation and sudden damage detection, which can be operated in a fully autonomous and real-time manner. Finally, an efficient data management application is developed to enable real-time data processing and visualization for end users.

Chapter 6

SENSOR FAULT MANAGEMENT FOR RELIABLE DECISION MAKING

This chapter proposes an autonomous strategy for sensor fault management which is practical for large-scale WSSNs. A three-stage approach is developed, which includes detecting, identifying, and recovering common sensor faults. The proposed strategy is validated using vibration data collected from the Jindo Bridge, South Korea. The effectiveness of the integrated sensor fault management strategy is evaluated by conducting a case study on decentralized modal analysis of the Jindo Bridge, using both raw and recovered sensor data. The result of the case study demonstrates the efficacy of the proposed strategy for reliable decision making.

6.1 Sensor Faults of WSN

In many WSSNs, low-cost small-size commercial off-the-shelf components are increasingly adopted, which are vulnerable to harsh environments. As a result, sensor malfunctions are more likely to occur in WSSNs than in their wired counterparts (Lo et al., 2016). There are two categories of sensor malfunctions: sensor failure and sensor faults. The term sensor failure is used when sensor nodes are incapable of sending back sensor data or are unresponsive to user commands. On the other hand, the term sensor fault is used when sensor nodes are able to report the data, but the data is corrupted in some way. Data loss is a typical type of sensor fault (Paradis & Han, 2007; Bao et al., 2013; Yang & Nagarajaiah, 2016), but it is not in the focus of this paper, because data loss is treated as sensor failure in some wireless smart sensor platforms. For example, the reliable communication protocol in the Illinois Structural Health Monitoring Project (ISHMP) Toolsuite (Rice et al., 2010) will discard the entire data record, if data packet loss cannot be recovered. Sensor faults can be examined in two ways: a system-centric view and a data-centric view (Ni et al., 2009). From a system-centric view, sensor faults are analyzed on the basis of the root causes of faulty data, which can be categorized into several types, such as calibration faults, connection failures, and low battery. From a data-centric view, sensor faults are investigated based on the features of the measured sensor data; from this perspective, sensor faults are generally categorized as spikes, drift, excessive noise, and non-linearity (Lo et al., 2015; Dragos et al., 2016). The discussion in this chapter will address sensor faults from a data-centric view.

To better understand the types of sensor faults that might be encountered, consider the monitoring data from the Jindo Bridge. The WSSN deployed on the Jindo Bridge in South Korea had a total of 113 wireless smart sensor nodes (Imote2), as shown in Figure 6.1. Some data sets corrupted with sensor faults were reported in the vibration measurements (Asadollahi & Li, 2017). The corrupted sensor data had one of three types of faults: drift, spike, and bias (Peng et al., 2017). Typical examples of sensor faults in the measurement data are shown in the Figure 6.2. The drift faults are the most common sensor faults, most of which result from temperature effects. The mean value of the sensor data with drift faults is not constant. Spike faults are the next most common and are possibly caused by multiple issues, such as low battery supply (Ni et al., 2009), loose electrical contact (Lo et al., 2011), and sensor saturations. While the first two root causes are demonstrated in the papers, the latter cause is observed in a lab test. Specifically, when acceleration exceeds the measurement range, the signal will be clipped in raw time history data. Then if a low-pass filter is applied to down-sample the signal, a spike will occur where the signal is clipped. Bias faults were relatively rare and are possibly caused by mounting issues. Thus, the Jindo Bridge data possesses the three types of sensor faults that occur most frequently in WSSNs for SHM applications.

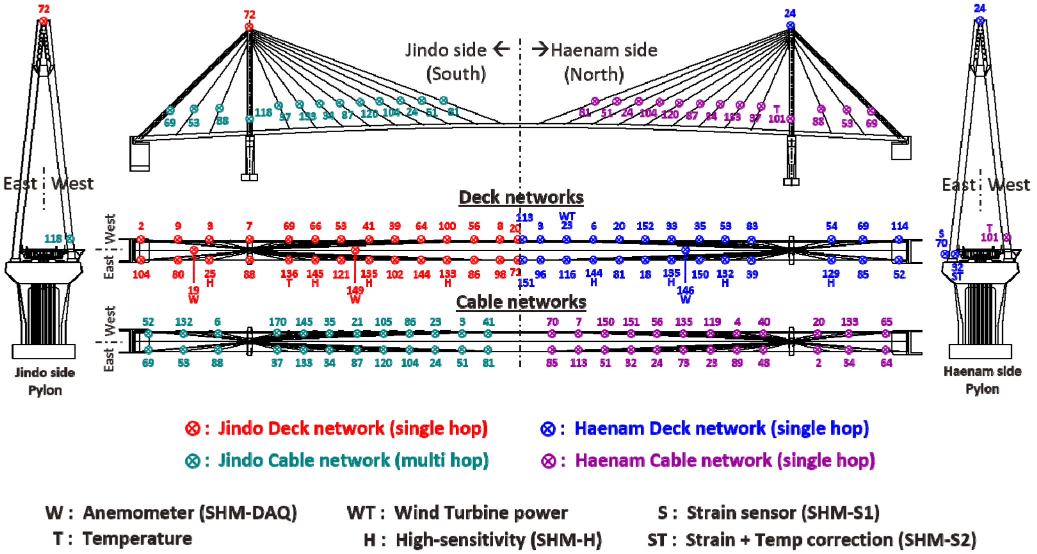
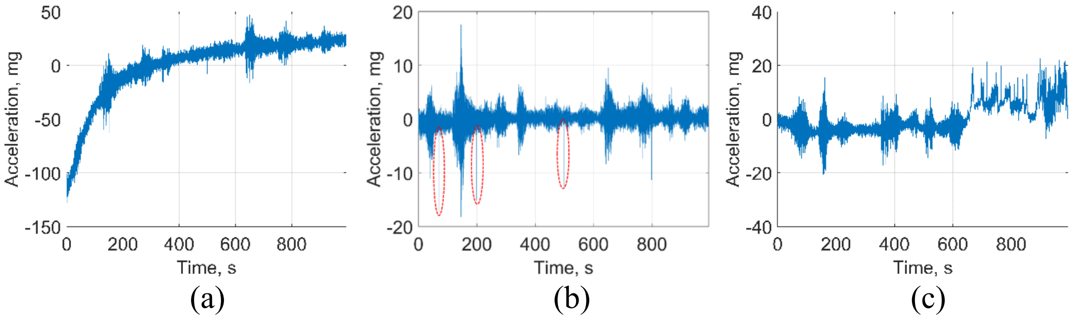


Figure 6.1 the WSSN on the Jindo Bridge (Spencer et al., 2016).



(a) (b) (c)

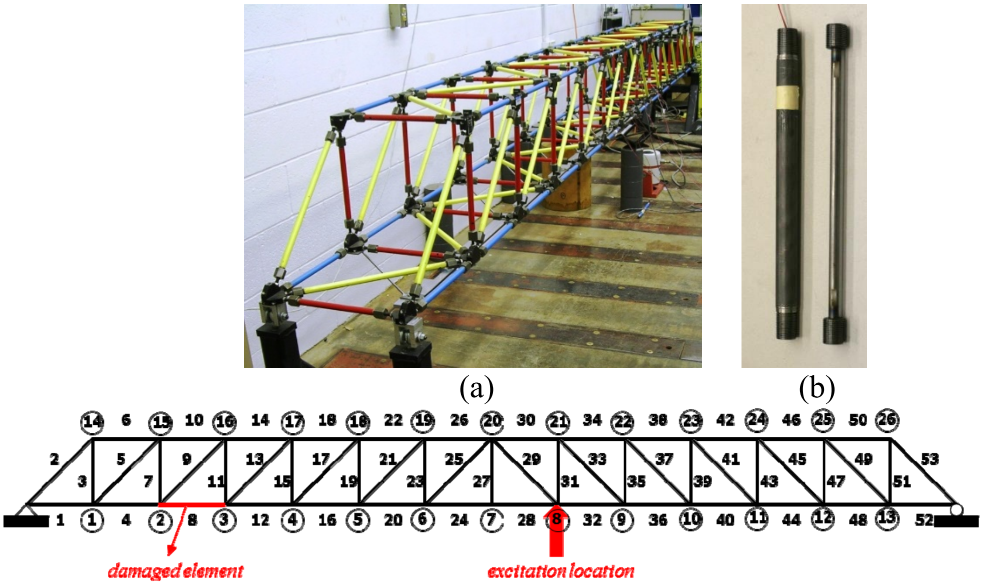
Figure 6.2 Sensor faults in SHM of the Jindo Bridge: (a) drift fault, (b) spike fault, (c) bias fault.

Sensor faults will change the outcomes of SHM, resulting in false condition assessment. The problem can be exacerbated, as more and more computing is done on the sensor nodes. For example, for the sensor nodes deployed on the stay-cables on the Jindo Bridge (Spencer et al., 2016), the raw data was processed independently to estimate the cable force; the estimated cable tension was then transmitted to the base station. Sensor faults in the raw data could significantly affect tension estimation in the respective cables, potentially resulting in incorrect decisions made by engineers. Therefore, sensor faults should be identified and addressed to have confidence in SHM results, especially for WSSNs where significant analysis is done locally on the sensor node.

6.2 Sensor Fault Effect on SHM

To develop effective strategies to mitigate the effect of sensor faults, a better understanding of the impact of sensor faults on SHM results is needed. To this end, a 14-bay truss structure for which extensive vibration data is available will be examined. The data is modified to simulate common sensor fault (e.g., drift, spikes, and bias). As the Power Spectral Density (PSD) is frequently used in various structural health monitoring strategies, the impact of these faults on the PSD will first be examined. Subsequently, the effect of sensor faults on a common damage detection strategy will be investigated, as a typical example of vibration-based SHM. Finally, the analysis of sensor fault effects will be used as a basis for developing multi-stage fault mitigation strategies for WSSNs in SHM applications.

A 5.6m long, three-dimensional truss structure with 14 bays, is attached to two rigid supports simulating pinned-pinned connections (see Figure. 6.3). The truss members are steel hollow circular tubes with an inner diameter of 0.428 in. and an outer diameter of 0.612 in. By aid of specially designed joints, each element can be easily replaced. In particular, one of the truss elements, Element 8, is replaced with an element with one having a 52.7% cross section reduction to simulate damage. A shaker is installed to excite the structure by generating a load of band-limited white noise (BLWN). The BLWN load has a frequency band of up to 200Hz. Sensors are deployed on 26 joints on one side of the truss structure to measure both vertical and horizontal acceleration. Each data set consists of 1000 seconds of data sampled at 256Hz. All the data sets are manually checked to ensure that no sensor faults are present.



(a)

(b)

(c)

Figure 6.3 the 14-bay truss structure (Gao & Spencer, 2008): (a) truss structure, (b) truss elements, (c) sensor locations.

Three common sensor faults are analyzed, including drift, spikes and bias. Faulty data with drift and bias can be represented as the combination of fault-free data and a sensor fault component (Dragos & Smarsly, 2016; Chang et al., 2017). Additionally, faulty data with spikes can be modeled as impulses superimposed on normal sensor measurements (Lo et al., 2011). Therefore, in the analysis in this section, drift, spike, and bias faults were first artificially modeled and then superimposed on the normal measurement. In particular, $x(t)$ represents fault-free data and $z(t)$ represents sensor fault component; the data with faults $y(t)$ is then given by

$$y(\mathrm{t}) = x(\mathrm{t}) + \mathrm{z}(\mathrm{t}) \tag{6.1}$$

where

$$z_{drift}(\mathrm{t}) = \frac{1}{2}(\mathrm{t}^{1/9} - 1) \tag{6.2}$$

$$z_{spike}(\mathrm{t}) = \begin{cases} 1.2 & t = 500 \\ 0 & otherwise \end{cases} \tag{6.3}$$

$$z_{bias}(\mathrm{t}) = \begin{cases} 0 & t < 500 \\ 0.5 & otherwise \end{cases} \tag{6.4}$$

The size of faults is set to make the faulty data close to some of the measurement data that was obtained from the Jindo Bridge deployment. For example, the ratio of the spike size to the Root-Mean-Square (RMS) of the measurement in the lab is close to that observed in one faulty data set of the bridge measurement. Figure 6.4 shows the plots of three common fault components in the time domain.

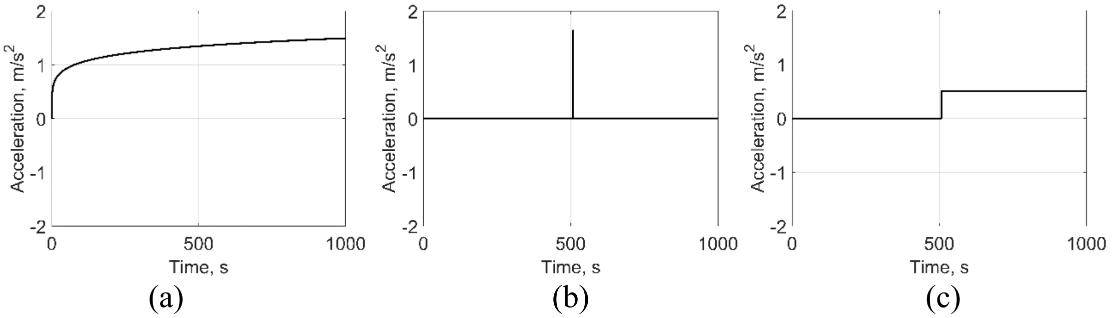


Figure 6.4 Sensor faults: (a) drift, (b) spikes, (c) bias.

6.2.1 Impact of Faults on the PSD

The impact of sensor faults on the PSD is first examined. Sensor data is assumed to be sampled at period of T, and discrete sequences are obtained as follows,

$$y[n] = x[n] + z[n] \tag{6.5}$$

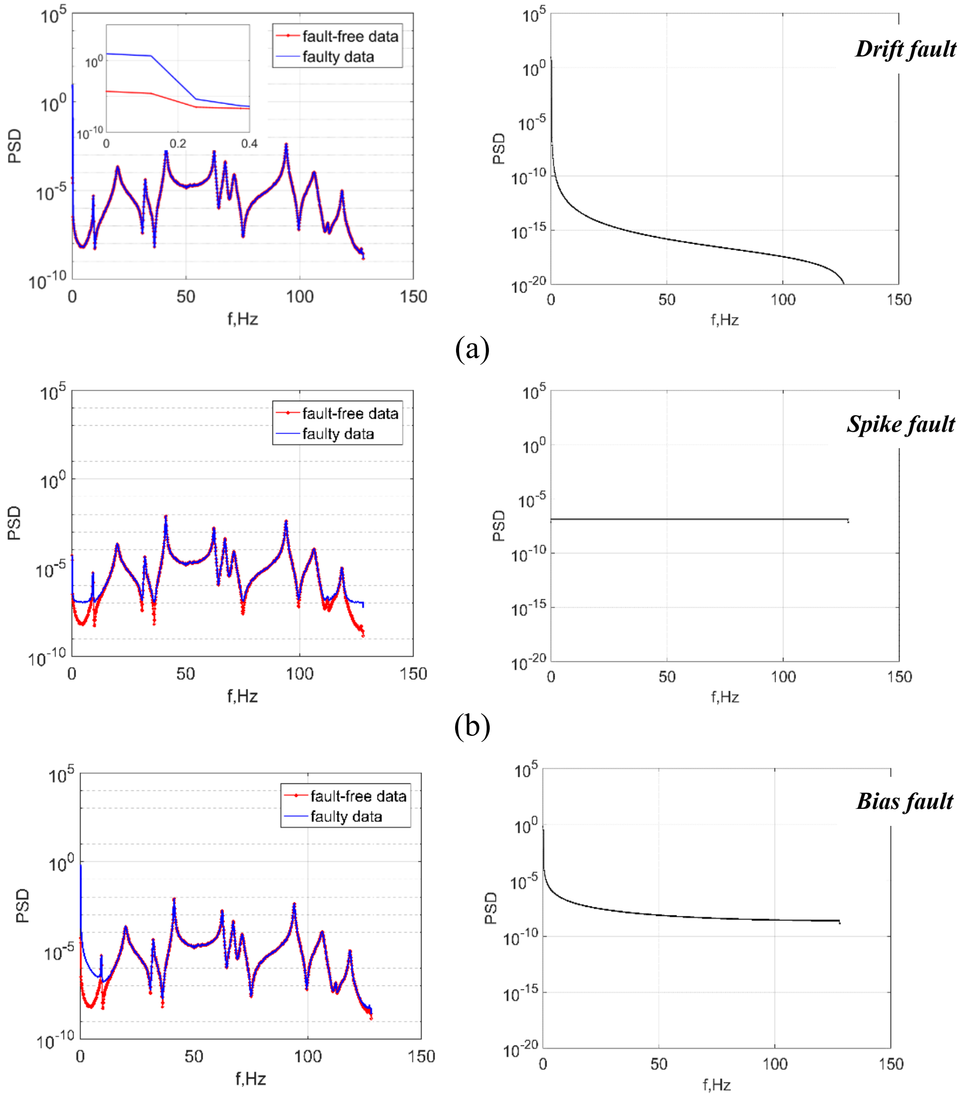
where $x[n] = x(nT)$, $y[n] = y(nT)$ and $z[n] = z(nT)$. Applying the discrete Fourier transform, this relationship can be expressed in the frequency domain as

$$Y_K = \sum_{n=0}^{N-1} y[n] \exp[\frac{-j2\pi kn}{N}] = X_k + Z_k \tag{6.6}$$

Assuming $x[n]$ and $z[n]$ are uncorrelated, the PSDs of the faulty data can be written as,

$$S_{YY} \approx E[Y_K^* Y_K] = E[X_K^* X_K] + E[Z_K^* Z_K] \tag{6.7}$$

Eq. (6.7) reveals that the PSDs of faulty data are equal to the sum of PSDs of fault-free data and PSDs of sensor fault components. To illustrate this point, the three common sensor faults are introduced to the vertical vibration data collected at Node 3 after Element 8 is replaced with one of reduced cross section. As shown in Figure 6.5, plots of PSDs for faulty data, fault-free data, and sensor fault components demonstrate the relationship expressed in Eq. (6.7). More specifically, a drift fault mainly affects the magnitude of PSD of the measurement in the low frequency domain. A spike fault is considered to be equivalent to a certain level of band-limited white noise, which affects the magnitude of PSD of the measurement in whole frequency domain uniformly. A bias fault affects the magnitude of PSD of the measurement in whole frequency domain non-uniformly. To sum up, sensor faults affect the magnitude of PSD of the measurements; the impact of three common faults on the PSD is different from each other.



(a)

(b)

(c)

Figure 6.5 PSD of sensor data with faults: (a) drift, (b) spikes, (c) bias.

6.2.2 Impact of Faults on Detection and Damage Localization

In this section, the impact of data faults on damage detection algorithms is examined. To this end, the decentralized damage locating vector (DLV) strategy (Gao & Spencer, 2008) is considered, because such decentralized approaches employing edge computing are more efficient and practical for deployment on wireless smart sensor networks. In the decentralized DLV, the measurement data of each node in a cluster is processed to obtain corresponding impulse response functions (IRF). Subsequently, the IRFs are fed into the Eigensystem Realization Algorithm (ERA) to estimate the modal properties and then calculate flexibility matrices. The change of the modal flexibility matrices before and after damage are analyzed to obtain the DLVs within local sensor communities. The DLVs are then applied to the structure to analyze the stress of elements; the stress value which is smaller than a threshold indicates a damaged element.

As shown in Figure 6.6, the truss joints are divided into 11 local groups, and sensors are deployed on each group of 6 nodes to conduct the decentralized DLV. To simplify discussion, the investigation of sensor fault effects on damage detection is only conducted on Group 2 sensors. Measurement data of Group 2 sensors are first fed into Eigensystem Realization Algorithm (ERA) routine for modal analysis. The first four modes are identified for estimation of flexibility matrix **F** which will change after damage. Using estimated **F** from the undamaged and damaged truss bridge, the DLVs are calculated. Normalized Accumulated Stress (NAS) is obtained, and small values of NAS indicate possible damaged elements. In this study, the three common sensor faults are introduced to the vertical vibration data collected at Node 3 when measurement data is obtained from the damaged truss bridge. The effect of sensor faults is examined by analyzing flexibility matrix construction error and calculating NAS in the damaged element (Element 8). Flexibility matrix construction error, $F_{\Delta}(k)$, is defined as,

$$F_{\Delta}(k) = \left\| F_d(k) - F_u \right\|_F \tag{6.8}$$

where, F_u is flexibility matrix of undamaged structure using fault-free data; F_d is flexibility matrix of the damaged structure using measurement that may contain faulty data. $\|\quad\|_F$ represents Frobenius norm. k is an index to characterize the degrees of each type of sensor faults, as follows,

$$k = A_f / A_s \tag{6.9}$$

where, A_s is maximum absolute value of the acceleration measurement, and A_f is an index to define sensor fault magnitude, depending on types of sensor faults (as illustrated in Figure 6.7).

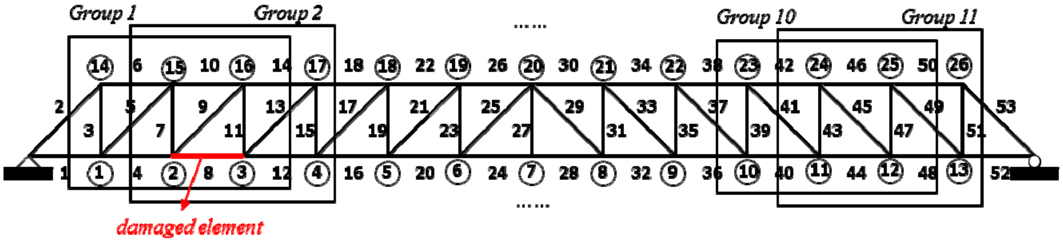


Figure 6.6 Sensor topology on the truss structure (Jang, 2010).

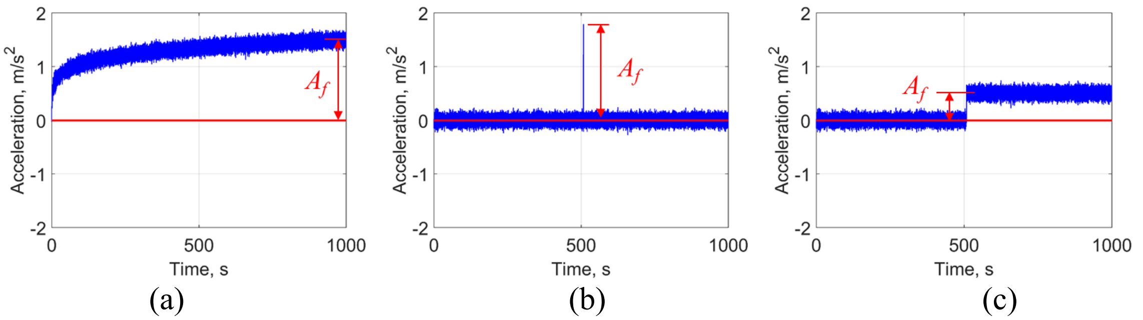
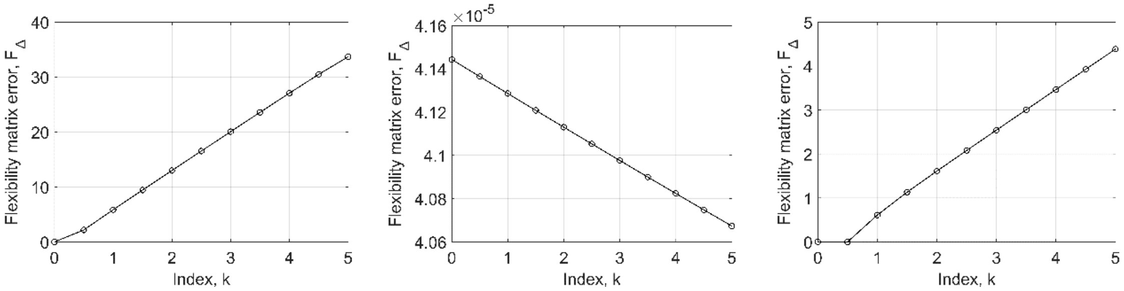


Figure 6.7 Fault magnitude index: (a) drift, (b) spike, (c) bias.

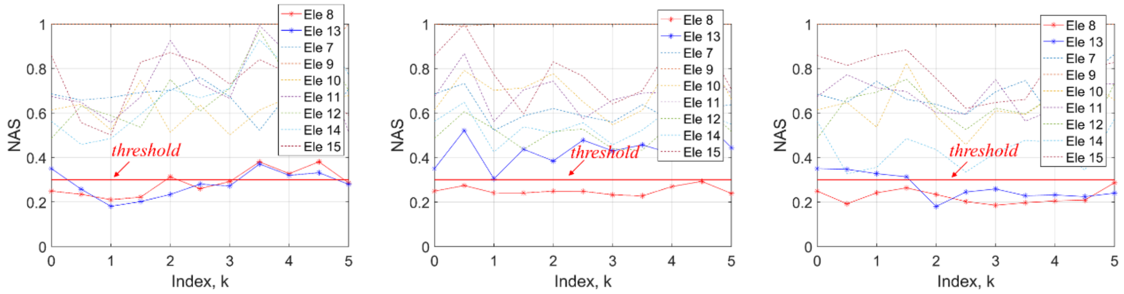
Figure 6.8 shows the result of flexibility matrix construction error resulted from sensor faults. In particular, $k = 0$ indicates that data is fault-free; the corresponding F_Δ is 4.14×10^{-5}. After sensor faults are added, F_Δ changes when the magnitude of faults increases. Different sensor faults have different impacts on the construction of the flexibility matrix. Drift fault is found to have the most impact, while spike fault is negligible. This result is because drift fault affects all the data points, but spike fault only affects one or several points of thousands of data points in data processing. From this analysis, sensor faults introduce error into the construction of the flexibility matrix. This error affects the subsequent analysis, as shown in Figure 6.9. The NAS of each element changes after sensor faults are introduced. Specifically, in Figure 6.9a, when the drift fault is small, the NAS value of Element 8 is smaller than the threshold, indicating that decentralized DLV method is able to identify damage. However, when the drift fault becomes more serious, the NAS value of Element 8 increases and finally becomes larger than the threshold, resulting in false-negative damage detection. In addition, when the index of drift fault is between 0.5 and 3, the NAS value of Element 13 is smaller than the threshold, which is false-positive. In Figure 6.9b, the effect of spike fault on damage detection is negligible. For the scenario of bias fault (Figure 6.9c), when the index of bias fault is larger than 2, the NAS value of Element 13 becomes smaller than the threshold, resulting in false-positive damage detection. The effect of sensor faults is explained as follows: the DLV algorithm recognizes a damage element based on the change of flexibility matrices before and after damage. Besides the damage, the introduction of sensor fault can be considered as another source of difference yielding the change of flexibility matrices. Adding a sensor fault in one measurement data sets reduce the NAS index of some intact members, which results in the false positive outcome. The affected element with a false positive outcome is connected to Node 3 where the sensor faults were introduced in the vertical measurements. The false-negative outcome occurs when the drift fault is relatively large. In this case, sensor faults dominate the change of flexibility matrix, and the distribution of NAS becomes uniform among the elements, such that both the NAS value of Element 8 and Element 13 increase and exceed the threshold. More investigation of this phenomenon will be carried out in the future. In sum, from this data analysis, sensor fault effects are found to degrade the quality of measurement data and affects subsequent decentralized SHM applications.

The understanding regarding the effect of sensor faults will provide a basis for managing sensor faults for WSSN. In particular, the effect of sensor faults on PSD will be utilized to detect sensor faults, which will be discussed in Section 6.3.



(a) (b) (c)

Figure 6.8 Flexibility matrix construction error: (a) drift, (b) spike, (c) bias.

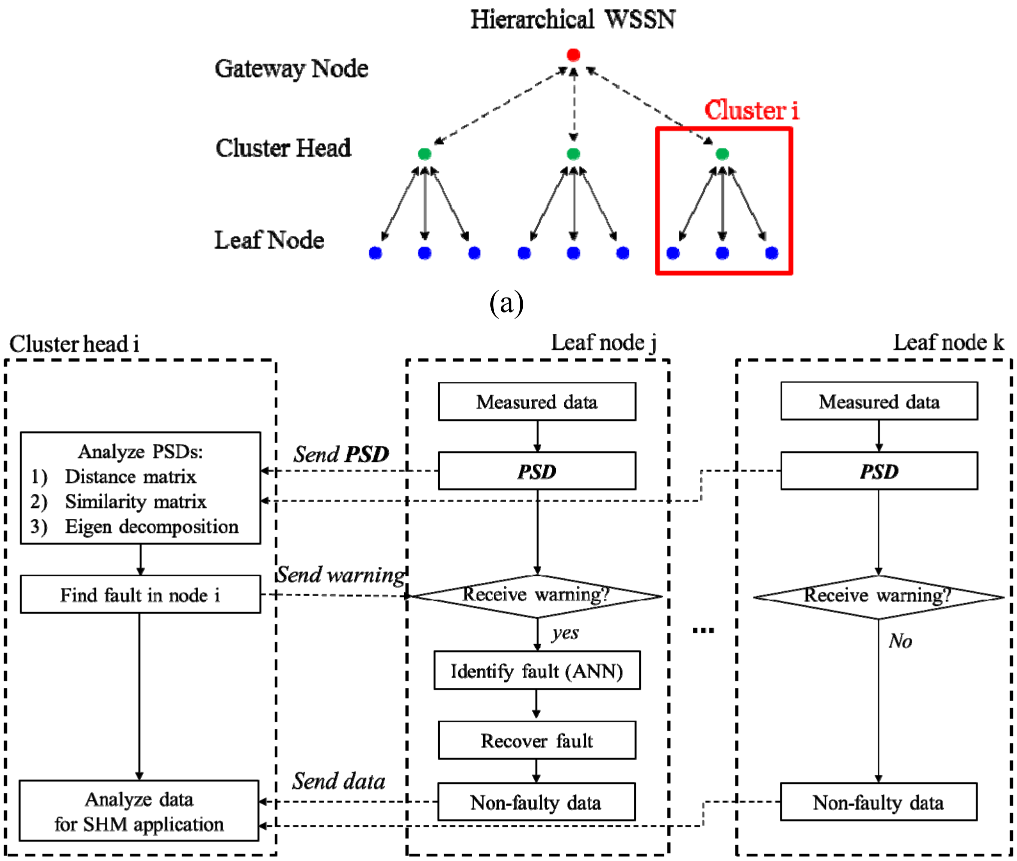


(a) (b) (c)

Figure 6.9 NAS of elements in Group 2: (a) drift, (b) spike, (c) bias.

6.3 Sensor Fault Management Techniques

Sensor fault management is performed using the proposed three-stage strategy: fault detection, fault identification, and fault recovery; a flowchart of the proposed strategy is illustrated in Figure 6.10. This strategy is mainly designed for decentralized hierarchical WSSNs (Sim, 2010), which consist of three layers of sensor nodes, including a gateway, cluster heads, and leaf nodes (see Figure 6.10). More specifically, in each cluster of sensor nodes, the leaf nodes collect vibration measurements, then each leaf node calculates and transmits the PSD of the raw data to the cluster head. Subsequently, each cluster head executes a distributed similarity test on the collected PSDs to identify the leaf nodes with faulty data. When a cluster head finds faulty data, it will send a warning message back to the corresponding leaf nodes. Later, faulty leaf nodes will conduct a fault identification task by using a pre-trained ANN model. When the sensor fault types are determined, the leaf nodes will execute an algorithm to recover the sensor faults and correct raw data. Finally, fault-free sensor data will be transmitted back to the respective cluster head for subsequent distributed data analysis. Each of these steps is described in detail in the sequel.



(a)

(b)

Figure 6.10 Fault management techniques: (a) decentralized hierarchical WSSN, (b) fault management steps in a cluster.

It is worth pointing out that, although this strategy is originally designed for decentralized hierarchical WSSNs, it is also applicable for centralized counterparts by arranging sensor data in different clusters after the gateway collects all the data. In addition, employing the ANN model to carry out both fault detection and classification would be another strategy, which although easier, doesn't work well for this case. The overall accuracy was less than 88% for this strategy based on the preliminary analysis, because the total number of faulty data sets available from the Jindo Bridge are much fewer than non-faulty data to train the ANN model, which makes the pre-trained ANN inefficient to detect the faults.

6.3.1 Fault Detection: Distributed Similarity Test

Assume that the measurements collected within a cluster of nodes are similar to each other. Therefore, as mentioned in Section 6.2.1, sensor faults will drastically affect the magnitude of PSD of the measurements. A distributed similarity test using the PSD of the raw data is proposed to detect faults, and it consists of the following three steps.

(i) Compute the distance matrix **D** of the PSD of vibration measurements collected in a cluster of nodes.

$$\mathbf{D}=\begin{pmatrix} 0 & d_{12} & \dots & d_{1n} \\ d_{21} & 0 & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & \dots & 0 \end{pmatrix} \tag{6.10}$$

where, $d_{ij}=\left\| S_i - S_j \right\|_2$, $S_i = \log P_i$, P_i is the PSD from sensor node *i*, and $\left\| \ \right\|_2$is the L^2-norm of a vector. In this study, S_i is normalized as, $S_i = \frac{\log P_i - \min(\log P_i)}{\max(\log P_i) - \min(\log P_i)}$.

(ii) Obtain the similarity matrix **R**, by using an entry-wise transformation of the matrix **D**:

$$\mathbf{R}=\begin{pmatrix} 1 & r_{12} & \dots & r_{1n} \\ r_{21} & 1 & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & 1 \end{pmatrix} \tag{6.11}$$

where, $r_{ij} = 1 - d_{ij} / \max(d_{ij})$.

(iii) Conduct an eigenvalue decomposition of the matrix **R**. All the entries of the similarity matrix are positive; consequently, the Perron–Frobenius (PF) theorem states that the matrix **R** has a unique largest real eigenvalue, denoted $\lambda_{\max}$, and the corresponding eigenvector, denoted *v*, can be chosen to have strictly positive components v_i. Moreover, the following relations holds:

$$\mathbf{Rv} = \lambda_{\max}\mathbf{v} \tag{6.12}$$

where, $v = [v_1, v_2, \cdots, v_n]$. Ideally, if no fault exists in the measured data, the PSDs from all sensor nodes should be similar; therefore, the entries of the eigenvector *v* should be close to each other, and the expression $v_j / \sum_{i=0}^{n} v_i$ is approximately equal to $1/n$. On the other hand, the components corresponding to faulty data will be smaller than non-faulty data, and the expression $v_j / \sum_{i=0}^{n} v_i$ will be smaller than $1/n$. Therefore, the index Ind_j based on the PF vector is proposed to identify faulty sensors, along with the following criteria: if the index is smaller than a threshold, the corresponding sensor is faulty. The ideal threshold is 1/*n*, which may not applicable in real data sets. Because in real data sets, the PSDs may have some slight differences due to several issues, e.g., measurement error. As a result, some of the non-faulty data may have the index that is smaller than 1/*n*. Based on the practical experience of processing the Jindo data, a threshold of 0.85/*n* is recommended to capture the most of the sensor faults, where *n* is the number of sensor nodes in the cluster.

$$Ind_j = \frac{v_j}{\sum_{i=0}^{n} v_i} \tag{6.13}$$

The proposed test is similar to the method developed by Lu (2010). However, Lu's method is a centralized sensor fault diagnosis, which compares the Fourier amplitudes of peaks indicative of resonant response of all vibration data in a whole sensor network. The proposed approach is implemented in a decentralized fashion, which compares the PSDs of vibration data within a cluster of nodes and significantly improves the accuracy and efficiency of the results.

Additionally, a brief mathematical proof, based on limit state analysis, is given to justify the distributed similarity test. N fault-free sensors and m faulty sensors are given, such that m is small compared to N and the total number of sensors is $N+m$. For this case, the similarity matrix originally defined in Eq. (6.11) can be rearranged and idealized as in Eq. (6.14), such that the entries related to faulty data (i.e., data with drifts, spikes, and bias) sets are isolated from those related to non-faulty data sets.

$$\mathbf{R}=\begin{bmatrix}(1-\varepsilon_1)\mathbf{1}_{N\times N}+\varepsilon_1\mathbf{I}_N & \varepsilon_2\mathbf{1}_{N\times m}\\ \varepsilon_2\mathbf{1}_{m\times N} & \varepsilon_2\mathbf{1}_{m\times m}+(1-\varepsilon_2)\mathbf{I}_m\end{bmatrix} \tag{6.14}$$

where ε_1 is a small positive number corresponding to the measurement error in the fault-free sensors, ε_2 is a small positive number corresponding to the measurement error in the faulty sensors, **1** represents a matrix with entries equal to 1 and dimensions given by the subscripts, and **I** is the identity matrix with the dimension given by the subscript.

Consequently, in the idealized matrix **R**, all the diagonal entries are equal to 1 like in the original matrix, the off-diagonal entries in the first block (upper left) are equal ε_1, the off-diagonal entries in the last block (lower right) are equal ε_2, and the entries of the off-diagonal blocks (lower left and upper right blocks) are equal to ε_2. As discussed in Section 6.2.1, sensor faults (spikes, drifts, biases) will make the PSD of faulty data different from the non-faulty data; therefore, the corresponding entries are relatively small, as reflected in the off-diagonal blocks in matrix **R** in Eq. (6.14). Note that: (i) the arrangement of the matrix **R** does not affect the eigenvalues and merely affects the arrangement of the eigenvectors, and (ii) the assumption of the form of the idealized matrix, in which the errors are of the same order of magnitude, does not affect the conclusions of the limit state analysis.

The similarity matrix has positive entries, which allows the application of the PF theorem. Now, consider the following vector

$$\mathbf{v}=\begin{bmatrix}\mathbf{1}_N\\ \delta\mathbf{1}_m\end{bmatrix} \tag{6.15}$$

where δ is a small positive number to be found later, which means that the entries for the fault-free sensors are equal to 1 and the entries for the faulty sensors are equal to δ. The following relation is satisfied

$$\mathbf{Rv}=\begin{bmatrix}\left(N(1-\varepsilon_1)+\varepsilon_1+m\varepsilon_2\delta\right)\mathbf{1}_N\\ \left(N\varepsilon_2+m\varepsilon_2\delta+(1-\varepsilon_2)\delta\right)\mathbf{1}_m\end{bmatrix}=\lambda\mathbf{v} \tag{6.16}$$

where

$$\lambda=N+(1-N)\varepsilon_1+m\delta\varepsilon_2 \tag{6.17}$$

$$\delta=\frac{N\varepsilon_2+m\varepsilon_2\delta+(1-\varepsilon_2)\delta}{\lambda} \tag{6.18}$$

Note that the previous equation completely defines δ, and it will be used to show it is indeed small and positive. Moreover, note that the vector **v** has positive entries, and therefore according to PF theorem, **v** must be the PF vector and λ must be the PF eigenvalue, which has the largest absolute value. The previous equation can be rewritten as follows

$$m\varepsilon_2\delta^2-\left(\left(N-1\right)\left(\varepsilon_1-1\right)+\left(m-1\right)\varepsilon_2\right)\delta-N\varepsilon_2=0 \tag{6.19}$$

$$a\delta^2+b\delta-c=0 \tag{6.20}$$

where $a=m\varepsilon_2$, $\mathrm{b}=-\left(N-1\right)\left(\varepsilon_1-1\right)-\left(m-1\right)\varepsilon_2$, $\mathrm{c}=N\varepsilon_2$. Then, δ is given by the positive root of the previous quadratic equation

$$\delta=\frac{-b+\sqrt{b^2+4ac}}{2a}=\frac{2c}{b+\sqrt{b^2+4ac}} \tag{6.21}$$

Note that the other root is negative, and this root is indeed positive as assumed previously. If ε_1 and ε_2 are small, then $a=m\varepsilon_2$, $b\approx\left(1-\varepsilon_1\right)N-1$, $c=N\varepsilon_2$, and consequently

$$\delta=\frac{2c}{b+\sqrt{b^2+4ac}}\approx\frac{2c}{b+\sqrt{b^2}}=\frac{c}{b}\approx\frac{\varepsilon_2}{1-\varepsilon_1-1/N},\quad \lambda\approx N\left(1-\varepsilon_1\right) \tag{6.22}$$

In this case, δ is a small positive number as assumed, which means that the entries of the faulty sensors in the PF vector are small compared to the entries of the fault-free sensors. Consequently, the proposed method identifies the faulty sensors.

If ε_1 is small, and ε_2 is close to 1, then $a \approx m$, $b \approx N - m$, $c \approx N$, and

$$\delta\approx\frac{2N}{N-m+N+m}=1,\ \lambda \approx N+m \tag{6.23}$$

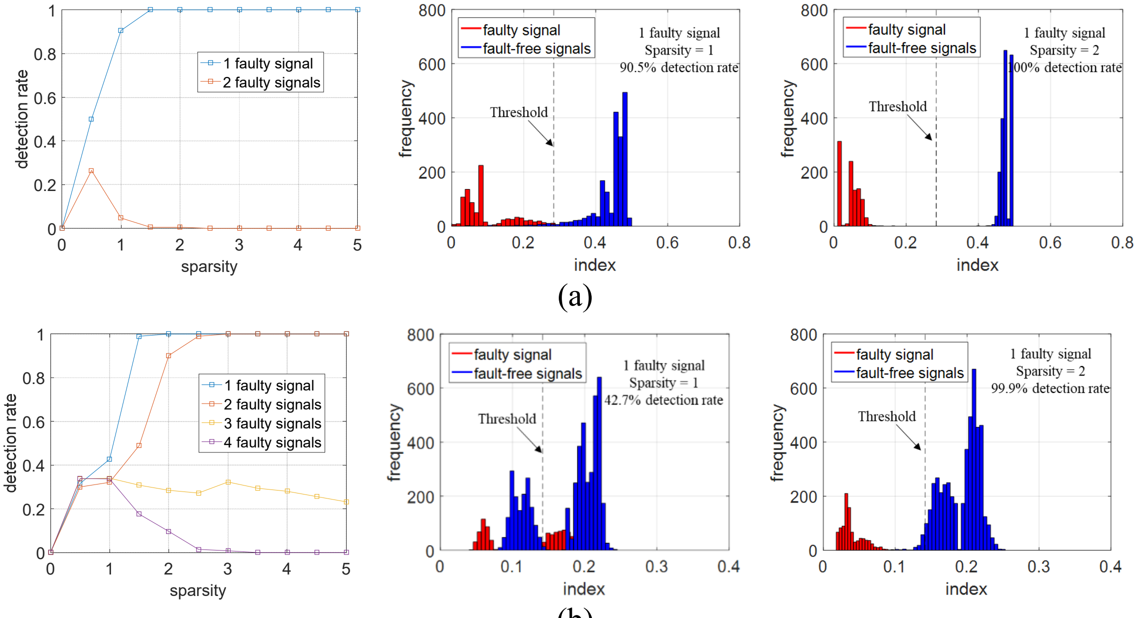
Therefore, in this case the result implies correctly that all sensors have entries equal to 1 in the PF vector, which means that the sensors are in good condition, and consequently, the method is compatible.

It is worthwhile to point out that the sparse faults of sensors in WSSN is the paper's focus, while catastrophic faults of WSSN are relatively easy to detect. In addition, some factors (e.g. spatial distribution and cluster size) may affect the performance of algorithm, but it depends on the objective of fault detection and should be considered on a case-by-case basis. For example, if the effect of sensor fault is much smaller than the effect of spatial distribution, though the fault may not be able to detect, but we may neglect the fault, because sensor faults don't make big difference on subsequent SHM analysis. Also, if the sensor nodes are densely deployed on a simple structure, the cluster size (i.e., the number of sensor nodes) could be large, because the PSDs of adjacent sensors would be very similar to each other.

To assess the algorithm performance, spike faults, as typical examples of sensor faults are examined herein. In particular, the sparsity measure of spike faults in a measurement is defined as,

$$l=\sqrt{\sum k_i^2} \tag{6.24}$$

where k_i is the i^{th} spike magnitude to signal amplitude ratio, defined in Eq.(6.9). This measure considers spike sparsity in terms of both number and magnitude. Particularly, the effect of a spike on the PSD is approximately proportional to the square of spike magnitude, and the square root makes l meaningful. For example, l = 1 equivalently corresponds to one spike of which the spike magnitude is the same with the signal amplitude. Afterwards, the experimental data obtained from Group 2 measurements in Section 6.2 is utilized to better examine the algorithm performance. Specifically, spikes of various numbers with various magnitudes are randomly set and added to the fault-free signals, and 1000 repeated tests are conducted for each degree of sparsity to estimate the detection rate. Clusters of two different sizes are checked extensively, including 3-node cluster and 6-node cluster. 3 is the minimal size of a cluster for the algorithm, because the *v* vector obtained in Eq. (6.12) is not unique for the cluster with a size of less than 3. And 6 is the maximum size for the experiment measurements, because only 6 joints are measured at the same time due to the limited experimental equipment. As shown in Figure 6.11, the algorithm works well as long as less than 50% of signals in a cluster have spike faults. The detection rate increases as the sparsity becomes larger. For example, in the 3-node cluster, the detection rate is 90.5% when sparsity *l* is 1; it increases to 100% when sparsity *l* is 2. As illustrated in the histograms, when the sparsity increases, the overlap area between fault-free signal indexes and faulty signal indexes is reduced, indicating the increase of detection rate and the decrease of detection errors. If the number of faulty signals increases, the detection rate decreases. Furthermore, as shown in Figure 6.11b, the detection rate in the 6-node cluster is lower than that in the 3-node cluster for given values of spike sparsity, because the fault-free signals in a small-size cluster are more “similar” to each other than in a large-size cluster herein.



(a)

(b)

Figure 6.11 The spike sparsity effect on detection rate using proposed algorithm: (a) a 3-node cluster, (b) a 6-node cluster. The sparsity and index values are calculated using Eq. (6.24) and Eq. (6.13), respectively.

In addition, the correlation coefficients of normalized S_i, as indicators of similarity between fault-free signals, were calculated to explore the detection sensitivity of spike faults using the proposed algorithm. In the experiment, correlation coefficients decrease as the cluster expands. Accordingly, clusters of four different sizes in Group 2 measurements were considered: 3-node, 4-node, 5-node, and 6-node cluster. The minimal value of the correlation coefficient was recorded, ranging from 0.9448 in the 6-node cluster to 0.9941 in the 3-node cluster. The detection rate for each cluster was obtained by repeating the test described in the previous paragraph. Afterwards, the relationship between the minimal correlation coefficient and the detection rate was investigated and plotted in Figure 6.12. Approximately, the detection rate is reduced as the minimal correlation coefficient decreases. The change of detection rate over the minimal correlation coefficient is significant when spike sparsity is 1. As the sparsity becomes larger than 1, the detection rate always keeps close to 1. As spike faults can be neglected if their sparsity is less than 1, the result obtained here demonstrates the satisfactory performance of the algorithm.

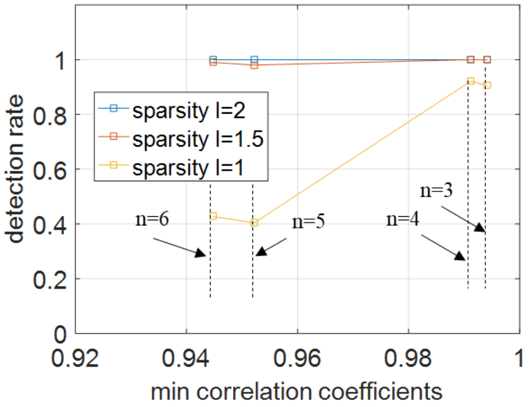


Figure 6.12 Relationship between minimal correlation coefficients of normalized S_i and the detection rate. n is the cluster size.

In sum, the algorithm is applicable for wireless sensor network in most scenarios, reasonably assuming that only a few number of signals have faults. The detection rate of the algorithm is satisfactory for sensor faults that may affect subsequent data analysis. Though more extensive evaluation is required to give a universal criterion for fault detection sensitivity, the exploration herein provides a reference for the usage of the proposed algorithm.

6.3.2 Fault Identification: Artificial Neural Network

An ANN model is applied in each leaf node to identify faulty sensor data into three categories, including data with spikes, data with drift, and data with bias. Note that fault identification doesn't require sensor data from other leaf nodes.

Vibration data collected from the WSSN on the Jindo Bridge was used to train and test the ANN. A total of 8090 data segments were considered with a fixed length of 200s, and the data segments were manually assigned with one of the four labels: fault-free data, data with spikes, data with drift, and data with bias. These data sets served as a learning database to train and test the ANN. Several statistical variables of acceleration data set were calculated and tested as candidates of input to the ANN. For example, for a data segment of 200 s, the mean value of all 5-second windows of data samples were calculated, obtaining a vector *mAvg* containing 40 entries. Analogously, another 9 vectors of statistical variables were obtained, including *mSqr, dMax, sgma, rtDM, Medg, sedg, M2ed, pedg,* and *S2dg.* For each vector of variable, 8 statistical characteristics were calculated: standard deviation, range, max, min, mean, interquartile range, skewness, and kurtosis. In sum, a total of 80 features were generated as the input candidates for the ANN, as shown in Table 6.1.

Table 6.1 Statistic characteristics of the ANN inputs.

No.	Variable	Description
1	*mAvg*	mean value of consecutive 5-seconds windows
2	*mSqr*	square of mean value of consecutive 5-seconds windows
3	*dMax*	maximum difference between consecutive 5-seconds windows
4	*Sgma*	standard deviation of consecutive 5-seconds windows
5	*rtDM*	(max value – mean value)/standard deviation of consecutive 5-seconds windows
6	*Medg*	output of an edge detection filter applied to the numerical derivative of *mSqr*
7	*Sedg*	output of an edge detection filter applied to *sgma*
8	*M2ed*	output of an edge detection filter applied to the numerical derivative of *Medg*
9	*Pedg*	output of a pulse detection filter applied to consecutive 5-seconds windows
10	*S2dg*	output of an edge detection filter applied to the derivative of *sgma*

Note: for each variable, eight statistical characteristics are computed: standard deviation, range, max, min, mean, interquartile range, skewness, and kurtosis.

However, not all features were important. Hence, a linear programming analysis was performed to evaluate the impact of the 80 features on sensor fault identification. Specifically, the following linear program is implemented

$$y_i(\vec{\omega}^T\vec{x}_i+\theta)\geq 0, \quad \forall(\vec{x}_i, y_i)\in D \tag{6.25}$$

where $\vec{x}_i \in R^{80}$ is an 80-dimensional vector which represents the 80 features of ith data segment; $y_i \in \{-1,1\}$ is a scalar representing the respective label of ith data segment; D is the set of all data segments; $(\vec{\omega}, \theta)$ represents a linear discriminant function that separates the data set D. In particular, $\vec{\omega}\in R^{80}$ is an 80×1 vector representing the weights of 80 features in the linear discriminant function, and θ is a threshold value. As a first step in the analysis, the data segments were labeled as, $y_i = 1$, if ith data segment had the drift fault; otherwise, $y_i = -1$. Then, the linear program was executed to separate data with drift from the rest of sensor data. A linear discriminant function and corresponding $\vec{\omega}$ were obtained, and a typical example is in Figure 6.13. The absolute value of weights for 30 features highlighted in a red dash box are relatively large, when compared to other features which are even smaller than 1×10^{-9} . This result reveals that these 30 features are important to identify sensor faults, and the other features can be neglected. Although the figure indicates that the number of features can even decrease to 6 (i.e., mAvg and mSqr considering the rag, max, min statistics), the 30 features are considered for training the ANN model to balance the accuracy and the computation efficiency. Ideally, more features will result in better results for ANN establishment, but it requires more data sets of relatively high quality. However, in real bridge data, the number of data sets is limited, and the data quality is not ideal. As a result, more features may result in overfitting. Therefore, the number of features we selected is in-between for trade-off. Similar analyses were conducted to separate data with spikes from the rest of sensor data, and data with bias from the rest of sensor data. In all these cases, the same conclusion hold that the same 30 features were the most important.

The 30 features described previously were selected as the input of the ANN. On the other hand, four output neurons were established to predict types of data sets, namely, non-faulty data and three types of sensor faults. Between the input layer and the output layer, different topologies of the ANN were tested, which contained various numbers of hidden layers as well as various hidden neurons per layer. Between adjacent layers, the neuron behaviors were defined and tested using different activation functions, such as the sigmoid function and the Rectifier linear units (ReLU) function. The preliminary test of the ANN model shows that the overall accuracy of sensor data classification is unsatisfactory, which is less than 88%. This happens because faulty data sets in Jindo Bridge data are fewer than non-faulty data, which makes the ANN inefficient to detect the faults. The lack of faulty data is indeed one of the challenges using real bridge data for ANN training. Therefore, in the finalized version of ANN method, the non-faulty data sets were excluded, and the ANN is only employed to classify three type of sensor faults (i.e. data with spikes, data with drift, and data with bias). Accordingly, the number of output neurons in the ANN model is reduced to 3.

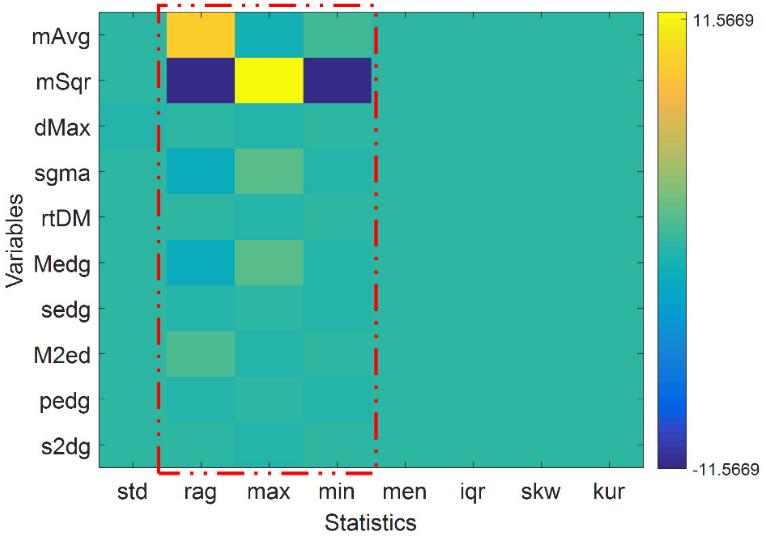


Figure 6.13 Weights plot of the linear discriminant function.

Following the common practice in neural network applications, the data segments were divided into a training and test sets. The training set was used to adjust the weights of the ANN; and after training process was completed, the test set was used to evaluate the accuracy of the ANN in identifying the types of sensor faults. Moreover, to reduce the variability of analysis results, a five-fold cross validation was applied to evaluate the performance of the ANN; i.e., the data segments were divided into five partitions with data of different types randomly sampled to ensure good balance in the train and validation stage. In the first round, partitions one to four were used for training process, and the last subset was used for testing. Analogously, another four rounds of training and testing process were conducted by using different partitions of data segments. Prediction errors of all five rounds were averaged to tell the prediction accuracy of the ANN. In addition, because the total number of faulty data sets was only 1/5 of original data sets, and the distribution of three types of sensor faults is also imbalanced. To address these challenges, we randomly oversample the data sets of faulty data for training the ANN model using Synthetic Minority Over-sampling Technique (SMOTE), not only to increase the total number of data sets for training, but also to balance the distribution of different types of sensor faults.

Based on the five-fold cross validation results, an optimal topology of the ANN was determined, as shown in Figure 6.14. Two hidden layers were defined with ten neurons each, and the sigmoid function was implemented for data processing. Meanwhile, three neurons were defined in the output layer using a softmax function to enable multi-class outputs. The learning curve, shown in Figure 6.15a, converges to a low error rate, which demonstrates that the number of iterations is adequate to train the ANN. The overall accuracy of the trained ANN applied to the test results is 95.2%. Figure 6.15b shows the confusion matrix, where each entry is divided by the number of data segments in each class; the color in each cell represents the number of data segments in the test set: darker colors correspond to larger numbers. Additionally, the diagonal elements in the confusion matrix represent the number of data segments in which the predicted label coincides with the true label; all diagonal values are approximately 95%, which confirms the accuracy of sensor fault identification using the trained ANN. It is worth pointing out that, the number of data segments for bias faults is very small, compared to other sensor faults. Therefore, 100% accuracy of the test for bias fault may not represent the true situation.

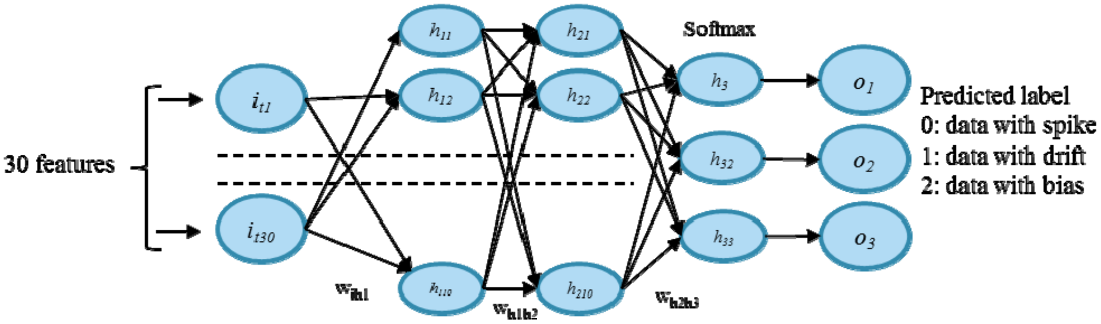


Figure 6.14 Topology of the artificial neural network for fault identification.

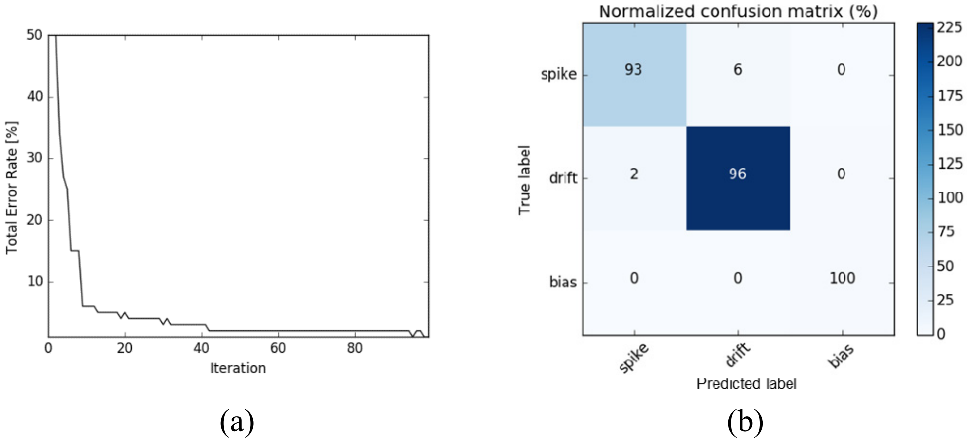


Figure 6.15 Test results of the proposed ANN: (a) learning curve (b) confusion matrix based on number of segments.

6.3.3 Fault Recovery

After the sensor fault type is successfully identified, fault recovery is performed on the original raw data. Different faults are recovered using different strategies, as illustrated in Figure 6.16.

For fault recovery of data with spikes, the location of the spike fault in time domain should be identified. For this purpose, the vector *dMax*, which is already calculated for the ANN input, is considered. When a spike occurs, a relatively large value of *dMax* is expected; therefore, an upper bound value of four times the standard deviation was set as a threshold, and any entry in *dMax* larger than this threshold implies a data segment that contains the spike fault. Finally, in the identified segment, all samples larger than three standard deviations plus the mean of the data segment are replaced by the mean value.

Fault recovery for data with drift is done by applying a correction function. For this purpose, the vector *mAvg*, which is already calculated for the ANN input, is considered; and a smooth polynomial fitting line based on the values of *mAvg* is obtained. The correction function is given by the fitting line (Fig. 16 (b)), and the data is recovered by subtracting this correction to the raw data. This simple recovery strategy works whether the original data has a drift or not; moreover if fault-free data is falsely identified as data with drift, this correction does not affect negatively the results.

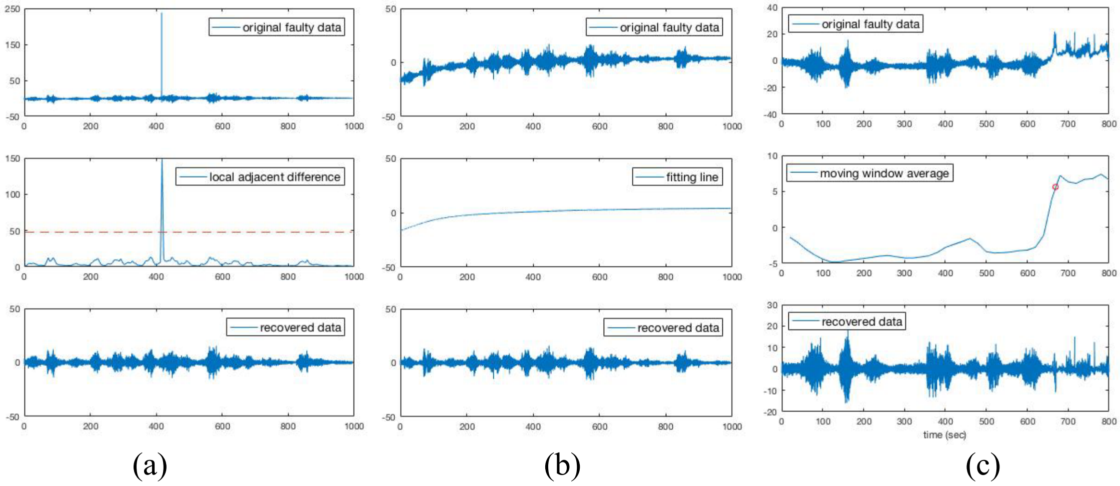


Figure 6.16 Raw data, fault recovery, and recovered data for: (a) data with spike (b) data with drift (c) data with bias.

For data with bias, the first step is to identify the initial occurrence of bias. For this purpose, a moving window average with 500 points is applied to smooth the raw data. The maximum numerical derivative of the moving window average approximately coincides with the occurrence of bias, and the raw data is divided into two segments: before and after the occurrence of bias. Finally, both segments are processed separately using the strategy of fault recovery for data with drift.

6.4 Case Study: Jindo Bridge Monitoring

6.4.1 Sensor Fault Management of Vibration Data Collected from the Jindo Bridge

The proposed strategy was evaluated using vibration data collected by a network of 26 sensor nodes deployed on the Jindo Bridge. The 26 sensor nodes were divided into 4 clusters for decentralized SHM applications, as shown in Figure 6.17. The collected vibration data was first checked visually. A total of 7 faulty sensors were detected; four of them had drift faults, and three of them had spike faults, as highlighted in the figure. Then, the proposed fault management strategy was conducted in each cluster. The number of sensor nodes in each cluster (n) is equal to 8, so the threshold value to determine faulty sensor data (0.85/n) was set as 0.106. The results of sensor fault detection are shown in Figure 6.18. The horizontal axis represents the Sensor ID in each cluster; the vertical axis represents the Ind_i described in Eq. (6.9). In Cluster 1, Sensor 1 and Sensor 8 have lower fault indices than the threshold value, 0.106; therefore, the faulty sensor data has been successfully detected using the proposed method. Similarly, in the results for Cluster 2, Cluster 3 and Cluster 4, Sensor 8, Sensor 14, Sensor 18 and Sensor 22 are successfully identified as faulty sensors. In sum, it can be concluded that most of the faulty sensors are detected. In Cluster 4, spike faults in Sensor 24 and Sensor 26 are not detected, because the magnitude of spike faults in these two sensors are relatively small and the change of PSDs due to these sensor faults are small. After the detected sensor faults are recovered, a second round of a distributed similarity test should be able to detect the spikes in sensor data of these two nodes. Moreover, false positive cases are found in data from Sensor 19 and Sensor 20. In fact, these two sensor nodes are far away from the other six nodes of Cluster 4. As a result, the PSD of sensor data from Sensor 19 and Sensor 20 are not similar to the PSD of the rest sensor nodes, and the assumption of the distributed similarity test is not true. However, for false positive cases, fault recovery strategy is not considered to affect the original data. Afterwards, the fault identification was performed using the pre-trained ANN, and the results of ANN were consistent with the data. Finally, fault recovery was performed based on the identified type of faults, as shown in Figure 6.19. In sum, this case study demonstrates that the strategy performs well for sensor fault detection, identification and recovery for WSSNs.

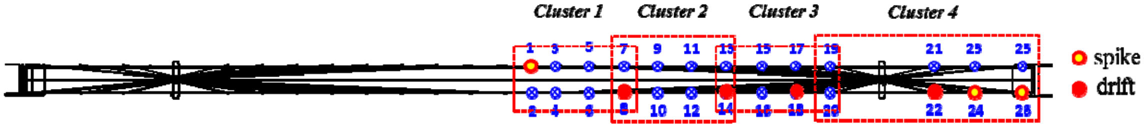


Figure 6.17 Topology of hierarchy WSSN on half-span deck of Jindo Bridge (plan view).

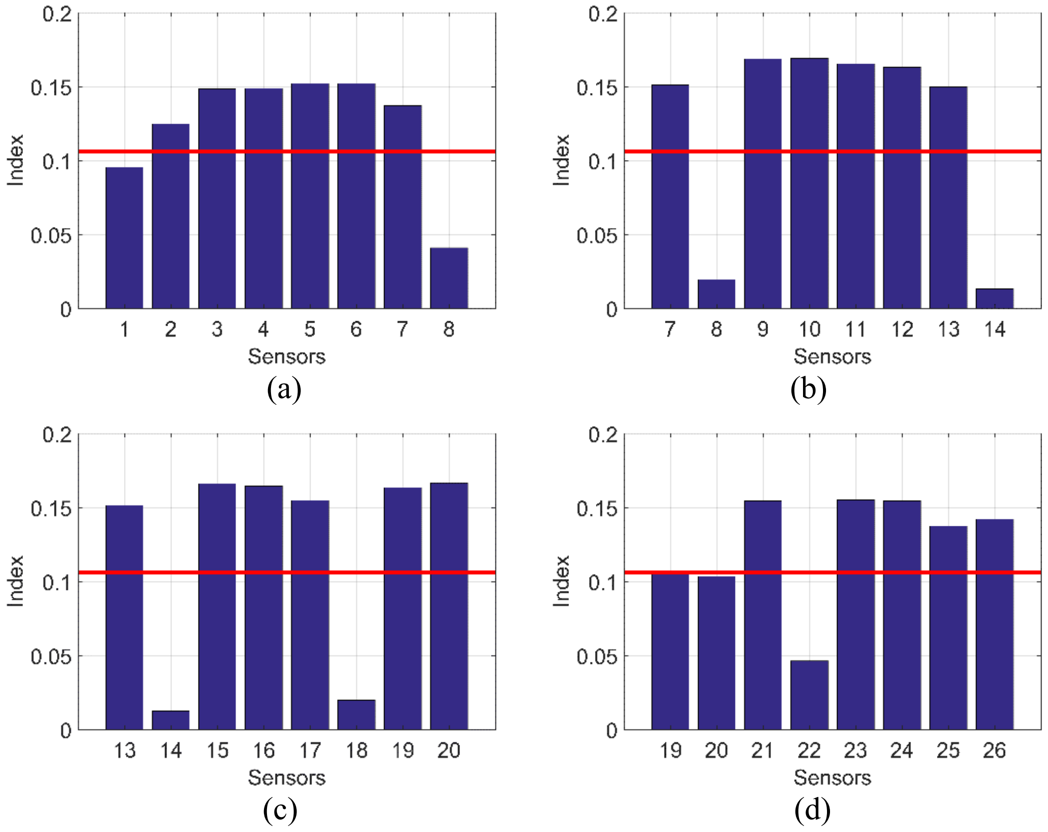


Figure 6. Figure 6.18 Results of sensor fault detection using the distributed similarity test: (a) Cluster 1, (b) Cluster 2, (c) Cluster 3, (d) Cluster 4.

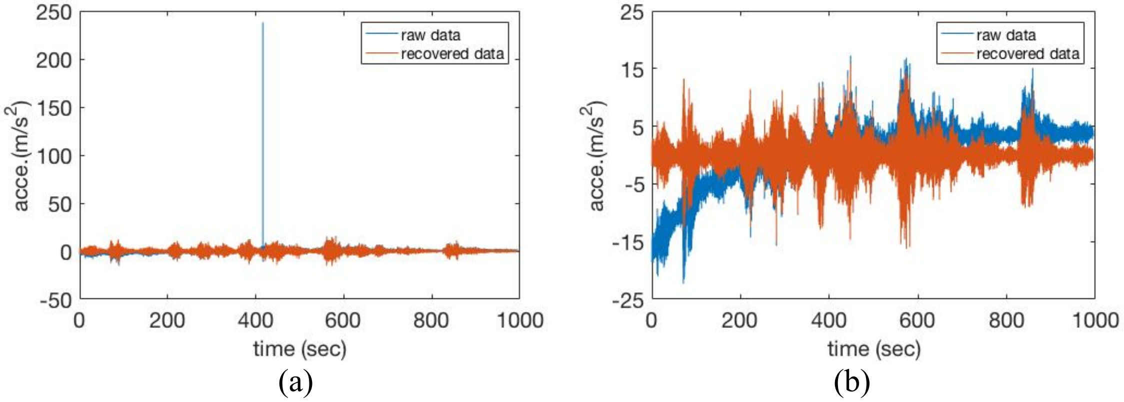
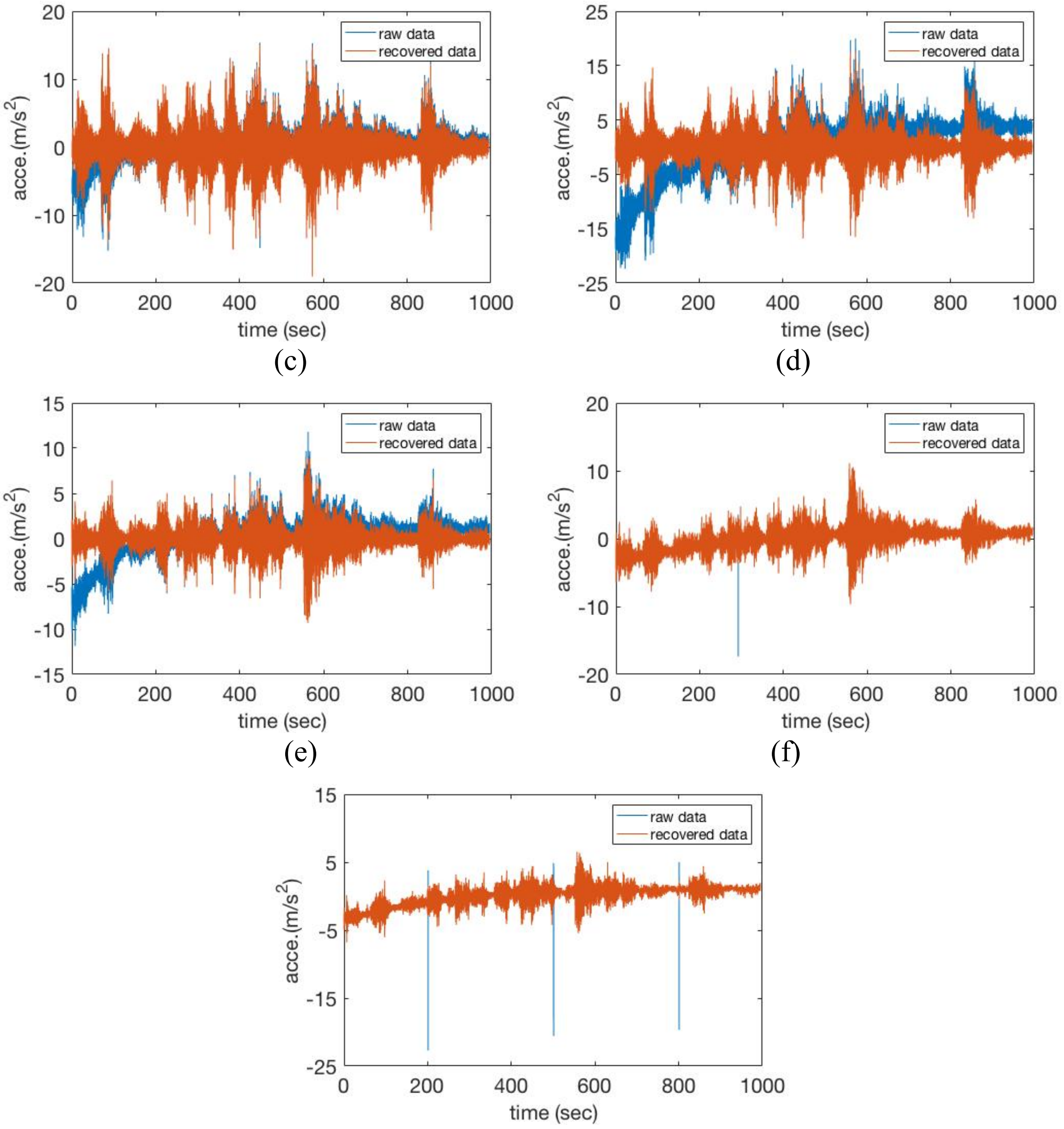


Figure 6.19 Results of sensor fault recovery: (a) Sensor 1, (b) Sensor 8, (c) Sensor 14, (d) Sensor 18, (e) Sensor 22, (f) Sensor 24, (g) Sensor 26.



(c) (d)

(e) (f)

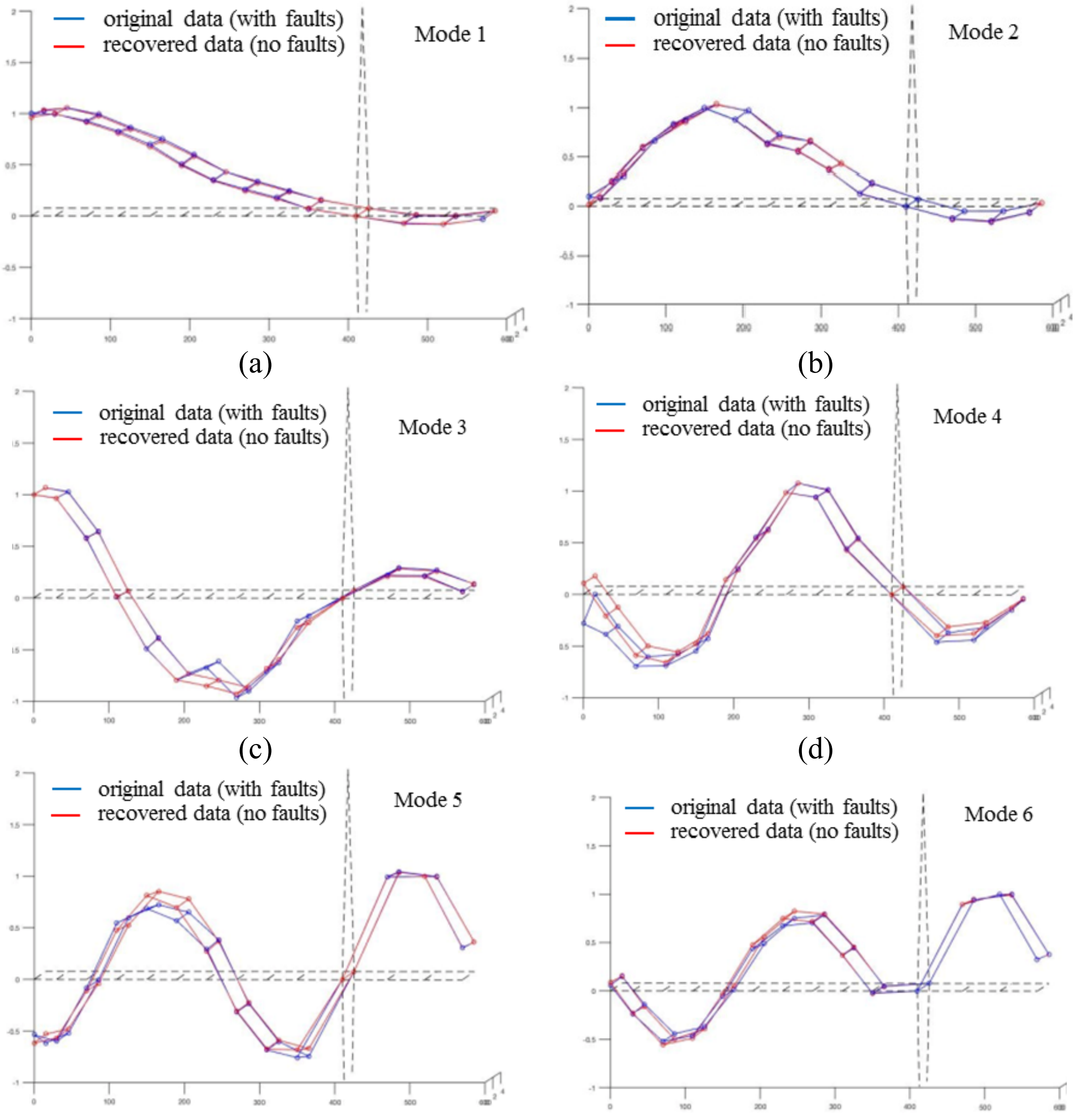
(g)

Figure 6.19 (cont.)

6.4.2 Decentralized Modal Analysis Using Vibration Data

The decentralized modal analysis method (Sim, 2010) is a system identification approach employing DCS strategy. Data acquisition is conducted in leaf nodes in each cluster. After that, modal analysis is conducted independently in every cluster head to extract local modal information using NExT/ERA. Local mode shapes are then combined to construct global mode shape, using a least square approximation in the gateway node.

In this section, decentralized modal analysis was conducted on the half span of Jindo Bridge using both the original vibration data which had sensor faults and the recovered vibration data (as described in Section 6.4.1). Subsequently, the results of the decentralized modal analysis were compared between faulty sensor data and recovered sensor data. The identified natural frequencies of first six modes are listed in Table 6.2; the corresponding mode shapes are plotted in Figure 6.20. As can be seen, natural frequencies are not sensitive to sensor faults, but mode shapes of several modes improve a lot after sensor faults are recovered. In particular, the mode shape of higher modes is more sensitive to faults, compared to lower modes, which is very interesting for future investigation. The result of mode shapes is consistent as that in the paper (Sim, 2010). In sum, it demonstrates that the proposed sensor fault management technique helps to improve the results of the decentralized modal analysis.



(a)

(b)

(c)

(d)

(e)

(f)

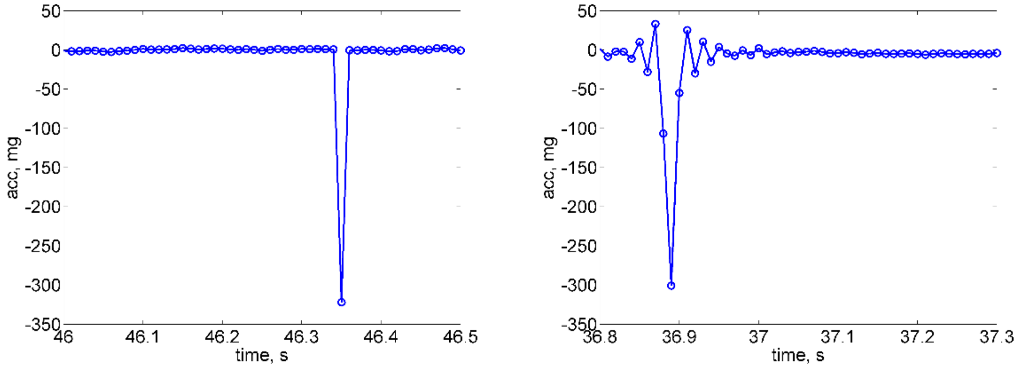
Figure 6.20 Decentralized modal analysis using original data and recovered data: (a) Mode 1, (b) Mode 2, (c) Mode 3, (d) Mode 4, (e) Mode 5, (f) Mode 6.

Table 6.2 Natural frequencies identified using decentralized modal analysis.

Modes	Natural frequencies (Hz)		
	data w/ fault	data w/o fault	Error
1	0.444	0.444	0.06%
2	0.650	0.651	0.06%
3	1.037	1.036	0.11%
4	1.363	1.361	0.15%
5	1.581	1.584	0.18%
6	1.670	1.669	0.06%

6.4.3 Discussion: Benefits and Challenges

The proposed strategy of sensor fault management has significant benefits, compared to previous techniques. First, the distributed similarity test benefits from inherent characteristic of decentralized SHM applications, in which sensor data within a cluster of nodes is similar to each other. This technique is scalable for large-scale WSSNs. Additionally, this method does not require a well-defined model to detect faults, and hence it will not be affected by changes of structural properties, e.g., structural damage. Moreover, it uses frequency domain for sensor fault detection, which significantly reduces data traffic. For example, in the case study of Section 6.4.1, the number of PSD data packets is only 2% of that of raw data in time domain. Furthermore, the ANN used for sensor fault identification is trained using vibration measurement data, and consequently, the ANN is able to identify the features in actual sensor faults. In contrast, most researchers employ artificial sensor faults to train and/or test their techniques, which may not be practical in field applications. A typical example is data with spike faults, as shown in Figure 6.21, in which an artificial spike fault is modelled as a single data point deviating far away from other data points. However, in a spike fault, a series of data points oscillate nearby the spike, because filters are usually implemented in sensor nodes to downsample the original raw data.



(a) (b)

Figure 6.21 Data with spikes: (a) artificial fault, (b) data fault.

On the other hand, the proposed strategy has a challenge of imbalanced learning to train the ANN. In field vibration measurements, only a small proportion of data is faulty. To address this challenge, appropriate techniques should be applied, such as random oversampling (Kotsiantis et al., 2006), to make the numbers of data with faults approximately the same with that of fault-free data.

6.5 Summary

In this study, sensor faults are investigated in vibration data collected from the Jindo Bridge. The effect of three types of sensor faults are analyzed on SHM outcomes, including frequency-domain results and damage detection. An efficient strategy is proposed to recover fault-free data, which consists of fault detection, fault identification, and fault recovery. A distributed similarity test is proposed and verified to detect sensor faults, using the PSD of raw data; an optimal ANN is used to identify sensor fault types, which is trained and tested using monitoring data collected from the Jindo Bridge. Finally, specific techniques are presented to recover data with spike, drift, or bias. A case study is conducted to demonstrate the efficacy of the proposed strategy of sensor fault management, and the benefits of using fault-free data in decentralized modal analysis. Further consideration of benefits and challenges of this strategy is also discussed.

Chapter 7

VADLIATION OF WIRELESS INTELLIGENT SUDDEN-EVENT MONITORING SYSTEM

In this chapter, a system, referred to as wireless intelligent sudden-event monitoring system (WISEMS), is described, which integrates components developed in previous chapters to support rapid condition assessment of civil infrastructures under natural disasters or human-induced hazards. First, the architecture of the integrated system is illustrated, and the functionality of each component is described. Then two applications are presented to validate its performance, including seismic building monitoring and bridge impact detection. The system provides a cost-effective wireless solution, which can capture the occurrence of sudden events and provide high-fidelity actionable information for emergency response and maintenance decisions in an efficient manner.

7.1 Integrated Smart Wireless Monitoring System: WISEMS

As illustrated in Figure 7.1, the developed system consists of five components: demand-based sensing prototypes, efficient time synchronization, high-throughput data acquisition, effective sensor fault management, and a graphic user interface for rapid condition assessment and real-time data visualization. These components are located in the wireless sensor nodes, smart base station, and MATLAB-based application, respectively. Each component is developed on the basis of research efforts in previous chapters, and their functionalities are summarized in Table 7.1. Depending on use cases, the WISEMS can be configured as either online or offline.

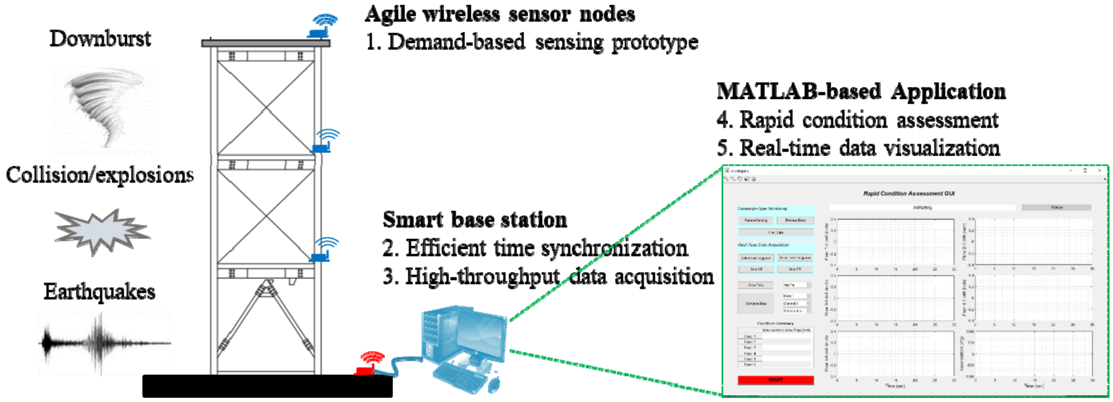

Agile wireless sensor nodes
1. Demand-based sensing prototype
Downburst
Collision/explosions
Earthquakes
MATLAB-based Application
4. Rapid condition assessment
5. Real-time data visualization
Smart base station
2. Efficient time synchronization
3. High-throughput data acquisition
Rapid Condition Assessment GUI

Figure 7.1 Architecture of WISEMS, an integrated system of components and services developed in previous chapters

Table 7.1 Functionalities for each component in the WISEMS.

Components	Online monitoring	Offline monitoring
sensing prototypes	• demand-based sensing to capture events • duty-cycled for user status inquiry • powered by rechargeable battery • solar panel for power harvesting	• same with online monitoring
time synchronization	• piecewise real-time time synchronization • clock synchronization is performed in sensor nodes • data synchronization is performed in the base station	• offline time synchronization • performed in sensor nodes
data collection	• live-streaming framework	• post-event data retrieval
sensor fault management	• distributed similarity test to find fault • performed in the base station	• same with online monitoring
condition assessment	• online condition assessment in real-time • performed in MATLAB-based application	• offline condition assessment
data visualization	• real-time data visualization using GUI • performed in MATLAB-based application	• offline data visualization

(1) *Online monitoring scheme*

If inspectors' objective is to monitor sudden events with user/PC presenting, the system can be configured as online monitoring. For example, the WISEMS can be deployed for seismic building monitoring, in which inspectors want to detect seismic motions and perform rapid structural condition assessment. In this case scenario, inspectors can stay in the building most time with PC being always-on to wait for the events to make emergency response if necessary. More precisely, the sensor nodes are *Demand-based WSS*s (see Section 3.1). They are in deep sleep mode most time to save battery energy, utilizing solar panel for power harvesting to prolong their life time over years. The main sensing mechanism of sensor nodes is event-triggered: they can wake up and start sensing immediately if a sudden event occurs; they automatically stop measurement if the event ends. In the meantime, the nodes record the time when the sudden event occurs, in the form of 24-hour clock and Gregorian calendar. In addition, they are also subjected to duty-cycling, such that inspectors can send requests to the nodes either for inquiry of their status (e.g., battery energy) or short-term measurement of the structure. On the other hand, the base station is always on, powered on by wall power. It waits for the notification message of events from sensor nodes and then coordinates them to perform real-time data acquisition on the basis of the framework (see Section 5.2). During the live streaming, the collected data samples are synchronized in the base station by the piecewise real-time time synchronization (see Section 4.3). In addition, the efficient sensor fault management techniques (see Section 6.3) are carried out to do quick check of the data quality. Afterwards, online condition assessments (see Section 5.3) are performed and the structural responses or condition assessment results are presented by the application in PC in real-time (see Section 5.4).

(2) *Offline monitoring scheme*

If inspectors' objective is to monitor sudden events without user presenting, the system can be configured as offline monitoring. For example, the WISEMS can be deployed for bridge impact detection, in which inspectors want to detect railway bridge collision from over-height vehicles or crossing sea bridge collisions from ships. In this scenario, inspectors cannot stay in the site all the time waiting for the unpredictable event; and a response latency of minutes for maintenance crews is allowed before an informed decision is made. Therefore, the offline monitoring using the WISEMS accommodates the requirements. More precisely, similar to online monitoring, sensor nodes installed on the structure is *Demand-based WSSs*, which can capture events automatically with minimal power budget; the base station is installed nearby and subjected to duty-cycling to save battery energy. After structural response is obtained, offline time synchronization strategy (see Section 4.3) is performed to synchronize data samples from multiple sensor node. Afterwards, sensor nodes send wake up commands to the base station. The station then retrieves sensor data from sensor nodes, perform quick data quality check (see Section 6.3), and store data locally. Finally, inspectors can come to the site and take the data for subsequent structural condition assessment.

7.2 Seismic Building Monitoring

7.2.1 Motivation

In last three years, several large earthquakes have occurred in North America. They are very difficult to predict, and their major shock is usually less than one minute. However, they can result in significant structural damage and casualties. In each earthquake, the public has a strong concern about the safety of their buildings. Especially for the earthquakes that have many aftershocks in several days, rapid condition assessment of the buildings is essential during each aftershock to let people know if their buildings are safe or not. Therefore, seismic building monitoring systems should provide enough information to tell (Celebi, 2013): (1) Is there structural damage in a building? (2) If damage occurred, what is its extent? (3) Does the damage threaten other neighboring structures? And (4) can the building be occupied immediately without compromising life safety?

Conventional seismic building monitoring using wired vibration sensors. For example, a total of 30 accelerometers installed in a 23-story building in San Francisco, CA. The monitoring system continuously obtained structural response at 200Hz and streamed the data back to remote users by high-speed internet (Celebi, 2012). The displacement of each floor was calculated by double integration of acceleration record, and interstory drift is calculated as the indicator to access floor conditions. However, they are generally expensive, making it extremely difficult to deploy every building in seismically-prone areas. Recently, a low-cost sensor, called Pulse, has been developed by Grillo, which cost $1000 or less and is affordable for seismically-vulnerable communities. The sensors can constantly monitor buildings and tell occupants when it has become damaged, using the same floor condition assessment strategies with wired counterparts (Wade, 2019). They leverage Power over Ethernet (PoE) technique to provide energy for always-on monitoring and transmit data back to end users. Though the cost-effect solution can detect earthquakes, it does not support building condition assessment in real-time. In contrast, the proposed system in this study can support real-time condition assessment of buildings using high-fidelity sensor data, comparable to the functionalities of conventional systems.

In this section, a laboratory test was carried out for seismic building monitoring, to validate the performance of the proposed system using online monitoring scheme.

7.2.2 Experimental Setup

As shown in Figure 7.2, the 6-story building model described in Section 5.3.2 was utilized and mounted on the uniaxial shaking table described in Section 3.3. The Kobe earthquake excitation was generated by the shaking table to represent a seismic motion for the building structure. 6 *Demand-based WSSs* were installed on each floor of the building model. The onset of a seismic motion was detected when the structural vibration was above 80mg for over 0.02s; sensor nodes immediately wake up and notify the base station. Afterwards, the base station broadcasts sensing parameters and beacons containing global timestamps for initial time synchronization. Then sensor nodes started real-time data acquisition and conducted high-quality sensing at 100Hz. The MATLAB-based application collected data from the base station and performed interstory drift estimation in real-time. In the meantime, Figure 1-5 in the GUI showed the plot of interstory drift estimations in real-time, from Floor #2 to Floor #6. In addition, Figure 6 in the GUI showed the acceleration measurement from all of 6 floors. During the seismic event, if the drift exceeds 0.2 inches, the animation line of associated interstory drift became in red, as a demonstration of rapid floor assessment. Note that the stiffness of each floor in the building model is very small, the threshold set in the test is much larger than the value used for real structures. The end of the event was detected when the vibration was below 40mg over 5s, at which time, sensor nodes stopped measurement and reentered deep sleep mode. Accordingly, the MATLAB-based application stopped animation line plotting and presented a summary of structural conditions.

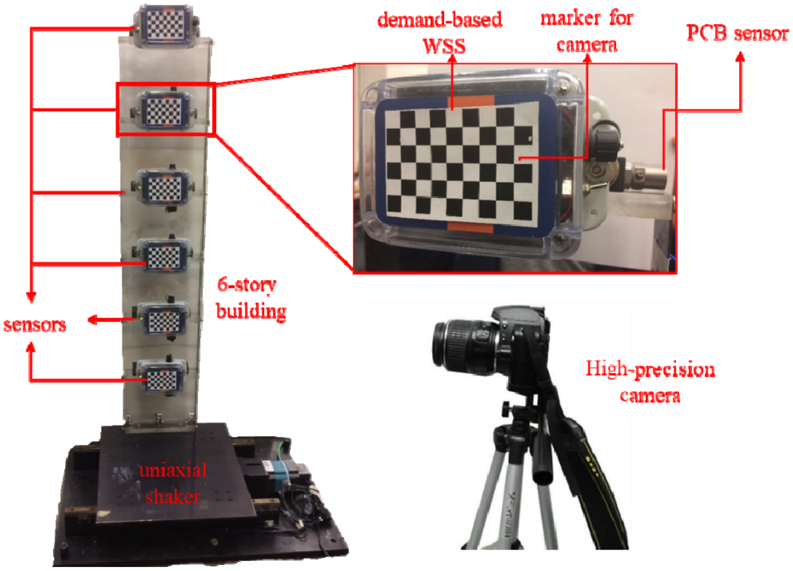


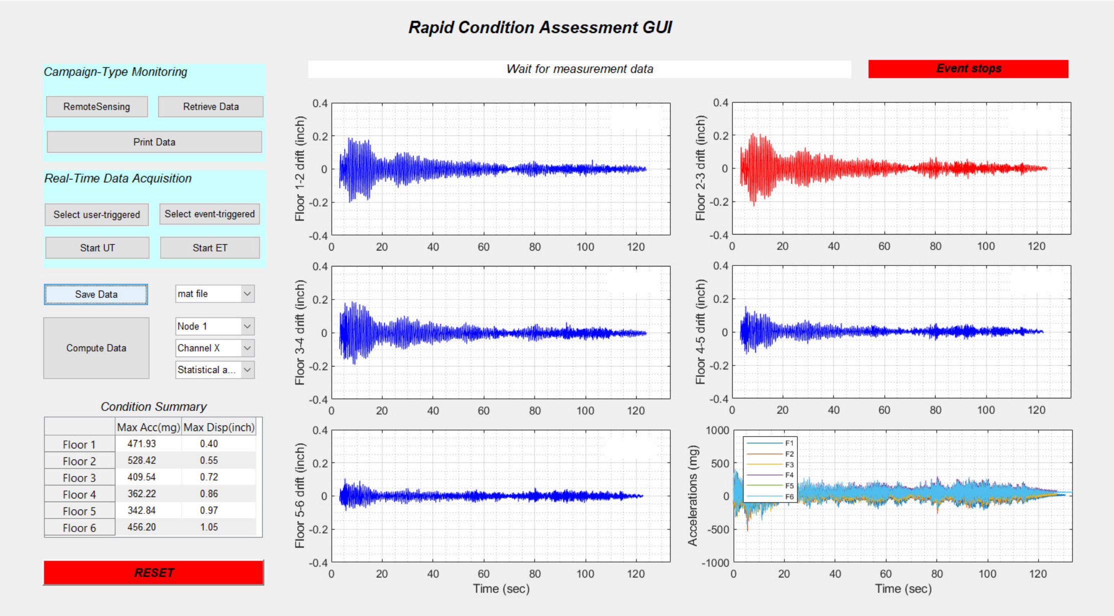
Figure 7.2 Test setup for seismic building monitoring.

As a reference for comparison, displacements were obtained using vision-based measurements. Each sensor node had a checkerboard pattern visible to the camera as target using for tracking. The size of the pattern was 6-by-9, and the MATLAB toolbox for detecting and tracking of checkerboard pattern was used directly. Nikon D3300 camera with 18-55mm lens were used, and data acquisition (video recording) was set to 60 frame-per-second (fps). In addition, wired piezoelectric accelerometers, model PCB353B33, were installed on the same floors and sampled at a frequency of 128 Hz. The same VibPilot was employed as a data acquisition system for the wired sensors. The acceleration from the wired sensors served as reference for acceleration comparison.

7.2.3 Results and Discussion

During the test, the WISEMS successfully capture the seismic motion and provide early estimation of story conditions in real-time. Figure 7.3 shows the screen capture of the GUI, in which structural responses are successfully captured, and interstory drifts between adjacent floors are estimated and plotted on the right. The plot of Floor #3 drift is red, indicating large drift occurs in this floor. Except Floor #3, other floors are considered as intact. In addition, the structural responses are summarized on the bottom left of the GUI, as listed in Table 7.2 for details. The structure response at Floor #2 is relatively larger than other floors, with the maximum acceleration of 528.42 mg. As the number of Floor increases, the displacement becomes larger. The top floor has the maximum displacement of 1.05 inch.

To validate the measurement data from the WISEMS, the acceleration measurement data from wired sensors is compared with the WISEMS data, as shown in Figure 7.4. To make a direct comparison in the time domain, the data sets from wired sensors were decimated to 100Hz and then sent through an 8-pole elliptic low-pass filter with a cutoff frequency of 40 Hz. Finally, the wired data sets were synchronized with the WISEMS data by maximizing the correlation function between records.



Rapid Condition Assessment GUI

Wait for measurement data

Event stops

mat file

Node 1

Channel X

Statistical a...

Condition Summary

	Max Acc(mg)	Max Disp(inch)
Floor 1	471.93	0.40
Floor 2	528.42	0.55
Floor 3	409.54	0.72
Floor 4	362.22	0.86
Floor 5	342.84	0.97
Floor 6	456.20	1.05

Figure 7.3 Screen capture of rapid condition assessment GUI.

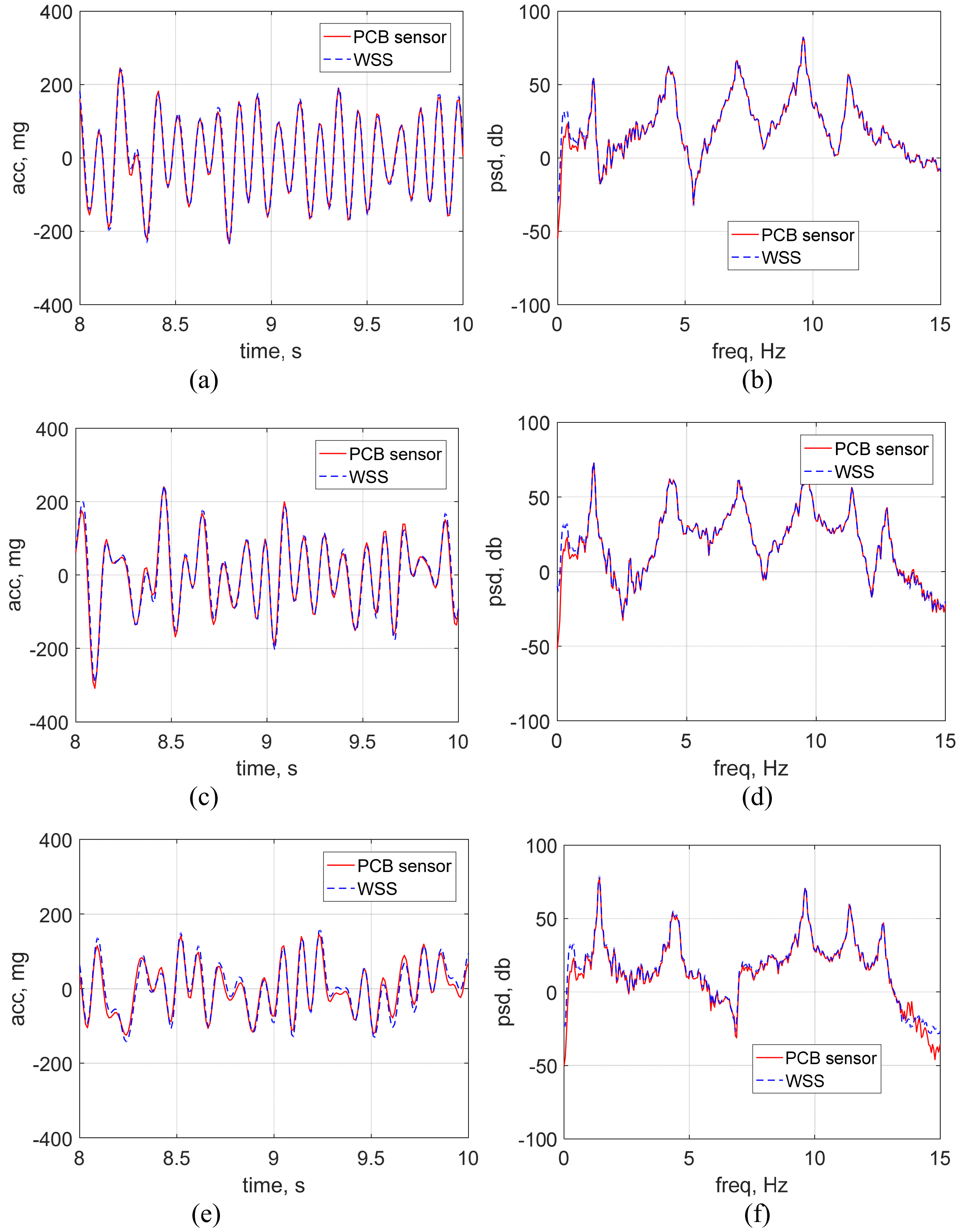


Figure 7.4 Acceleration measurements and results comparison: (a) time history data from Floor #1; (b) PSD from Floor #1; (c) time history data from Floor #3; (d) PSD from Floor #3; (e) time history data from Floor #5; (f) PSD from Floor #5.

Table 7.2 Summary of rapid condition assessment.

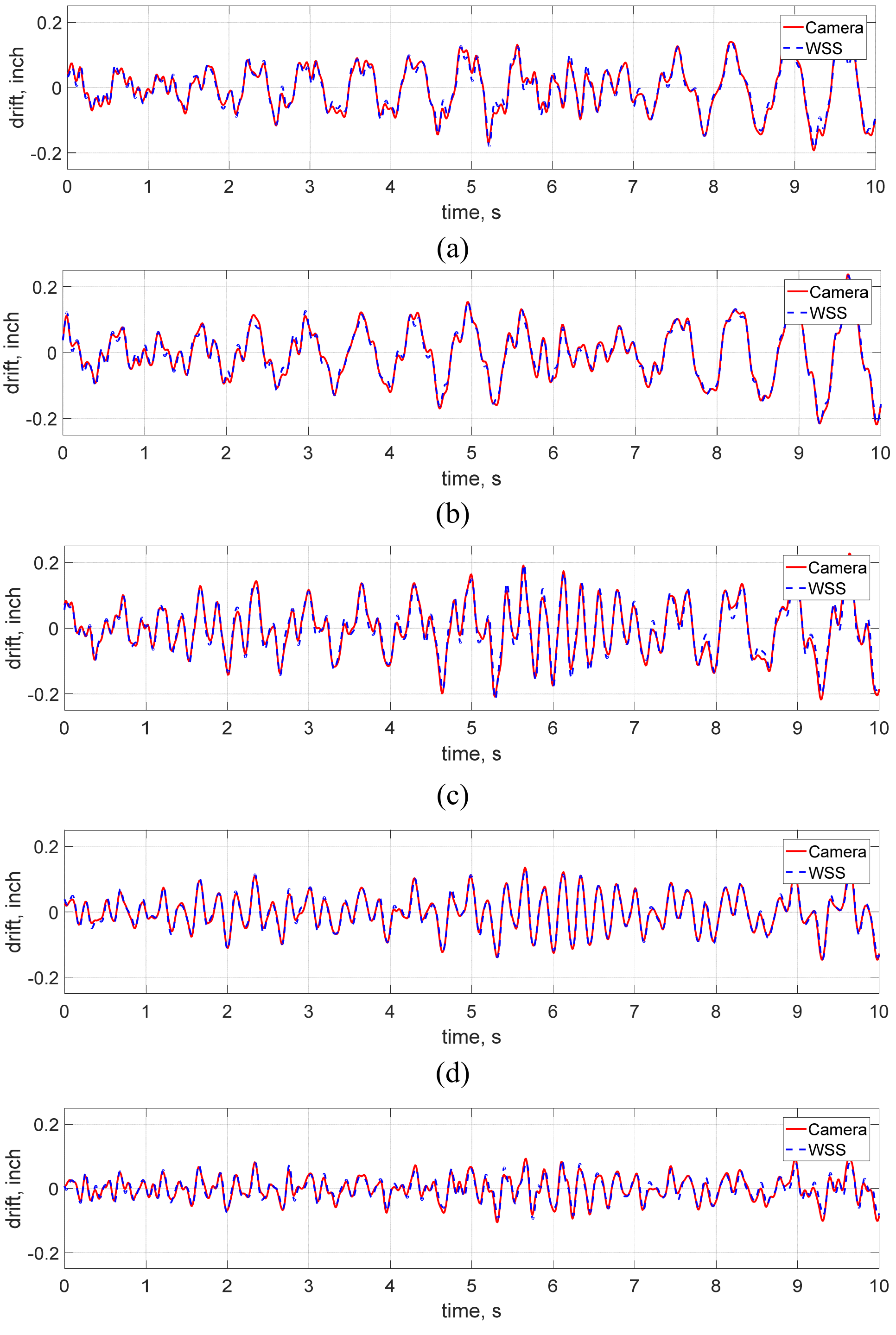
Floor #	Max acceleration (mg)	Max displacement (inch)
1	471.93	0.40
2	528.42	0.55
3	409.54	0.72
4	362.22	0.86
5	342.84	0.97
6	456.20	1.05

The comparison between wired and wireless sensor data reveals that the WISEMS successfully captured most of the 140-s structural response but lost approximate 2.5s data at the beginning of the event. This response latency is because of initial network configuration in the WISEMS. The initial configuration process consists of three parts: (1) sensor nodes send notification message to the base station, which continues for 1 seconds to compensate for uncertainty in the triggering time, such that the nodes wake up late can also be well-prepared for the next step of network configuration; (2) the base station sends commands to sensor nodes in two rounds, including sensing parameters and delay message for real-time data acquisition, respectively; (3) the base station broadcasts global timestamps for short clock synchronization to initiate TDMA schedule, which has been shorten to less than 1s, comparing to 30s pre-sensing time synchronization period for conventional TDMA.

In the test, the entire seismic motion has a duration of 140s, and the initial data loss only accounts for less than 2%. Therefore, the initial data loss in this case has negligible effect on story condition assessment. However, this phenomenon tells that, online monitoring using the TDMA is not suitable for some sudden events with a short duration of less than a minute. Instead, offline monitoring mechanism is recommended for short-duration events (e.g., bridge impacts).

Figure 7.4 shows the comparison between wired sensor data and the WISEMS in Floor #1, #3, and #5, respectively, in both time and frequency domain. The excellent agreement between the results of two monitoring systems demonstrate the ability of the proposed WISEMS to measure high-quality synchronized accelerations under sudden events. Figure 7.5 shows a time-window of comparison of the interstory drift estimation from Floor #2 to Floor #6 against the camera-based total interstory drifts. As these figures show, the estimated and the exact values agree well for all time steps.

The lab test results in this section demonstrate the ability of the developed WISEMS to detect sudden events, provide high-quality synchronized data, and present rapid condition assessment and real-time data visualization for inspectors.



(a)

(b)

(c)

(d)

(e)

Figure 7.5 Interstory drift estimation: (a) Floor #2; (b) Floor #3; (c) Floor #4; (d) Floor #5; (e) Floor #6.

7.3 Bridge Impact Detection

7.3.1 Motivation

Bridge strikes are identified as critical sudden events. A survey finds that over 30 states consider bridge strikes to be a major problem (Agarwal et al. 2012). Around 5,000 over-height vehicle bridge hits annually occur in U.S., resulting in over $100 million worth of damage to public & personal property (Singhal, 2015). Because of their unpredictable nature, many bridge impacts go unnoticed or unreported. Therefore, always-on monitoring is essential for deployed systems to enhance bridge safety through reliable detection and report of such events. In addition, many impact events are not sufficiently severe to meaningfully affect the bridge condition, but once reported, mandatory post-impact inspections must be performed, placing significant demands on bridge inspectors. Therefore, rapid condition assessment is another essential feature for monitoring system, to provide early estimation of bridge conditions, allowing the bridge engineers to prioritize resource allocation for timely inspection of the more severe impacts. To meet these two essential features, multiple solutions have been employed, but most of them are either too expensive for widespread implementation or insufficiently capable to perform the required tasks, as shown in Table 7.3. In contrast, the proposed system is a promising solution, providing satisfactory features.

This section describes the application of the WISEMS for bridge impact monitoring using offline monitoring scheme.

Table 7.3. Comparison of bridge strike assessment systems (Fu et al., 2019)

Characteristics	Visual inspection	NDT*	OHVDs**	Wired sensors	Proposed system
Total Cost	Very expensive	Very expensive	$100K	$90K	$10K
Event Detection	No	No	Yes	Yes	**Yes**
Rapid assessment	No	No	No	Yes	**Yes**
Traffic Closure	Yes	Yes	No	No	**No**
Accurate evaluation	No	Yes	No	Yes	**Yes**

* Non-Destructive Testing

** Over-Height Vehicle Detection Systems

7.3.2 Testbed and Proposed Strategies

A single-span pedestrian suspension bridge located over Lake of the Woods in Mahomet, Illinois, is considered as the testbed for bridge impact monitoring, as shown in Figure 7.6. The clear span of the bridge is 67 meters, with two bridge towers standing at each side of the lakeshores. The bridge contains 2 longitudinal girders and 2 suspended cables. The timber deck is supported on a series of 21 steel beams hanged by 42 suspenders. Boating is very popular in the Lake of the Woods, especially near the bridge where the boat peninsula is located. Therefore, one of the main threats of concern for the aging bridge is the collision between boats and the bridge. Installing a monitoring system is of great help to provide early warning of the bridge/boat collision and conduct rapid condition assessment. Accordingly, managers from the County Forest Preserve District can make emergency response and informed maintenance decisions (e.g., bridge closure for safety concern, or send maintenance technicians).

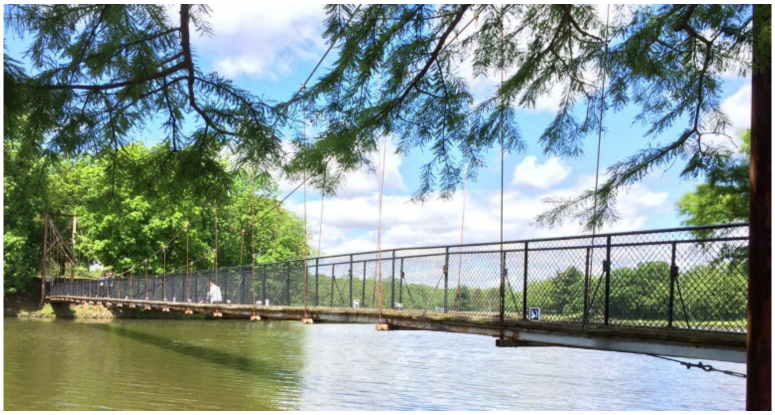


Figure 7.6 Suspension bridge in Lake of the Woods, Mahomet, IL.

While the proposed system has the essential functionalities to detect bridge impact events and capture structural response, a missing brick is the strategy for damage severity estimation using measurement data, which however is difficult. Lu et al. (2009) found that the bridge damage under impact events can be classified as two categories: location damage and global damage. Local damage, e.g., cracks, concrete crush, and reinforcement yielding, is positively correlated to peak impact force. Global damage, e.g., distortion, bending failure, and girder falling, is strongly related to impulse. Accordingly, both peak impact force and impulse can be used as two important indicators to infer bridge damage under impact events. The detailed flowchart of rapid condition assessment is illustrated in Figure 7.7. After deployed for bridge impact monitoring, the wireless sensor nodes in the WISEMS are turned off most time to save the limited energy. When an impact event occurs, the sensor nodes are quickly turned on to start sensing immediately. Once the event ends, the sensor nodes stop measurement and perform post-event time synchronization. Afterwards, the obtained bridge acceleration responses are transmitted back the base station. In the base station, the measurement data goes through a neural network model to estimate the impact information, including impact force (peak force and impulse) and location. Finally, the obtained impact force is compared with the maximum allowable impact force at corresponding location which is obtained from the numerical analysis. The bridge is suspected to be damaged, if the estimated impact force is larger than then maximum allowable force; then, the warning message will be sent to the bridge owners for them to make informed decision in an efficient manner. Otherwise, the system resets itself, and all the nodes are turned off to save the energy and wait for next event.

To achieve the proposed strategies, two models need to be established before testing, including finite element model (FEM) and neural network model (NNM). The FEM serves for two purposes: 1) to build the learning database for neural network modeling, and 2) to obtain maximum allowable impact forces at different bridge locations through nonlinear analysis. The maximum allowable impact force is estimated is defined as the maximum force that the bridge can sustain without damage. The NNM is used to identify impact locations and estimate impact forces. The following two subsections will describe the establishment of these two models.

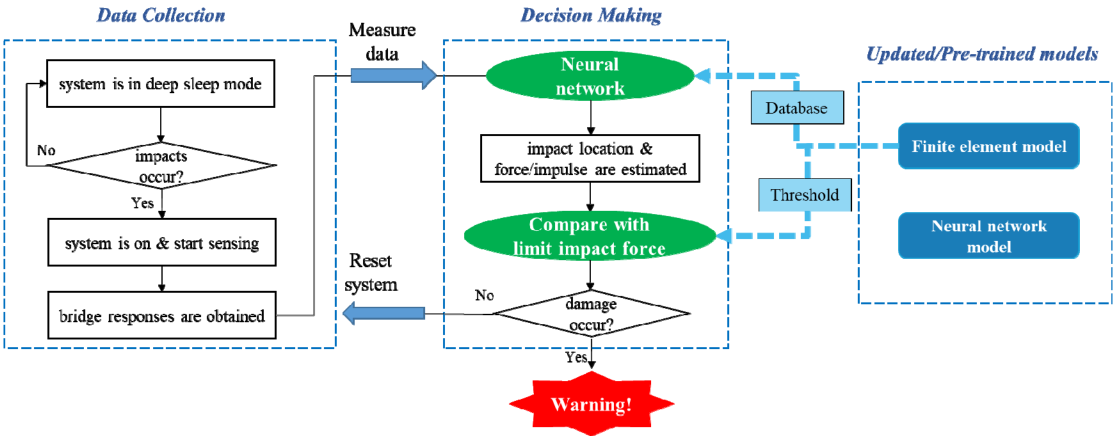

Data Collection
system is in deep sleep mode
No
impacts occur?
Yes
system is on & start sensing
bridge responses are obtained
Measure data
Reset system
Decision Making
Neural network
impact location & force/impulse are estimated
Compare with limit impact force
No
damage occur?
Yes
Warning!
Database
Threshold
Updated/Pre-trained models
Finite element model
Neural network model

Figure 7.7 Flowchart of bridge impact detection and rapid condition assessment.

7.3.3 Finite Element Modeling

As mentioned in Section 7.3.1, to realize rapid condition assessment of bridge impact monitoring, a reliable finite element model must be built and updated to represent the real bridge in terms of dynamic responses. To this end, full-scale modal analysis is required first to provide reference data for model updating.

A series of forced vibration tests on the bridge were carried out using a wireless monitoring system. The excitation method for the vertical test is a student heel-dropping on the bridge, and afterwards, the free vibration responses of the bridge were measured. For the lateral tests, the excitation method is a group of students' hand-impacting on the handrail. Because the bridge is flexible, both heel-dropping and hand-impacting should be large enough to excite the bridge in the frequency range of interest. A total of eight Xnodes were programmed with the *RemoteSensing* application and installed on the bridge. One node served as the gateway node, which was connected to a PC and located near the pier. The other seven nodes were installed on the bridge, used as leaf nodes to measure the bridge response. The sampling rate was 100Hz, with a cut-off frequency of around 48Hz to cover several major natural frequencies of the bridge. The measurement time for each test is configured as 1 minute, corresponding to 6000 points of data in each of the longitudinal, transverse and vertical directions. The sensors were located in half span of the bridge, as shown in Figure 7.8. This sensor deployment is aimed to capture first several natural frequencies and mode shapes for vertical and lateral modes, taking advantage of symmetric or antisymmetric of vibration modes. At each of seven measurement points, an Xnode was located on two 20-lb steel weights using magnets. Both the steel weights and magnet connectors are stable on the bridge during excitation. Wired sensors were also installed to serve for reference sensors during the test, but the results were not considered and discussed here for subsequent model updating.

The bridge responses were recorded using wireless sensors. Figure 7.9 shows the acceleration obtained from the Xnodes, for vertical and horizontal tests. Modal analysis was conducted using an output-only algorithm (NExT-ERA). The identified natural frequencies and associated mode shapes are shown in Figure 7.10. First several bending modes are identified, except 1st and 4th modes. The 1st bending mode is not captured for both wired and wireless sensors because it is estimated to be below 0.5Hz, and the performance of both sensors is not satisfactory in the low frequency range. For the 4th bending mode, it is related to the excitation location. Heel-dropping location is very close to the stationary point of forth mode shape, which is unable to excite the bridge to vibrate in this mode. Therefore, the 4th bending mode is not captured as well. Moreover, first three lateral modes are all successfully identified. These identified modal properties are then used for subsequent model updating of finite element model. In particular, horizontal modes are critical in this scenario, because bridge response is mainly in horizontal subjected to an impact.

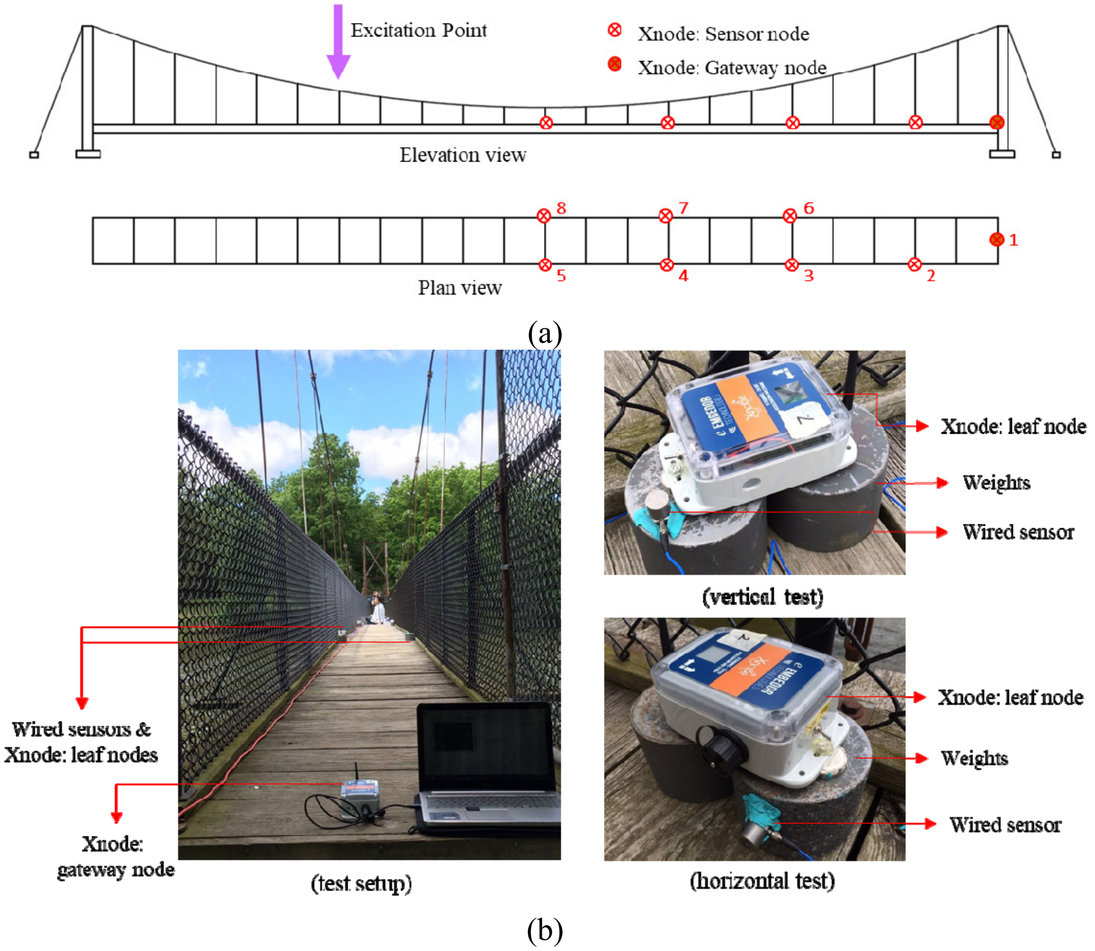
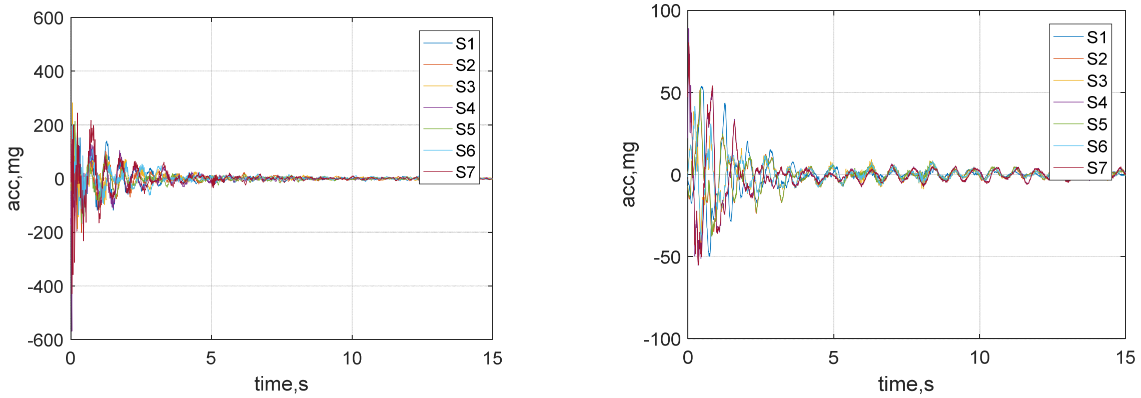
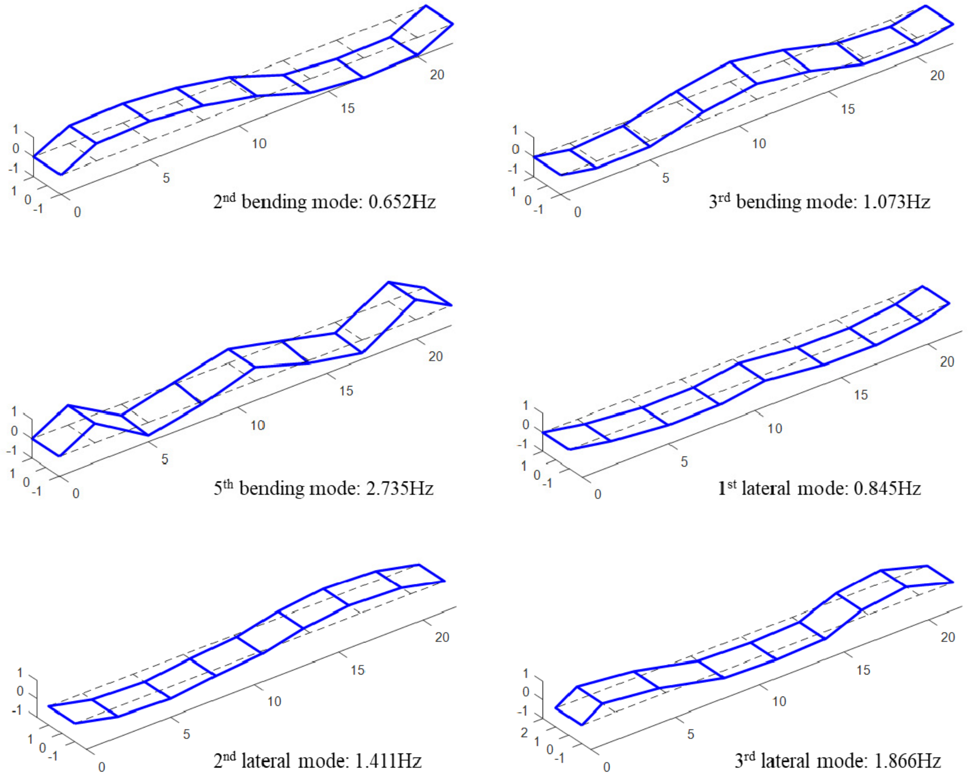


Figure 7.8 Campaign-type monitoring: (a) sensor and excitation locations (b) sensor deployments.



(a) (b)

Figure 7.9 Vibration responses using Xnodes: (a) vertical test, (b) horizontal test.



2nd bending mode: 0.652Hz

3rd bending mode: 1.073Hz

5th bending mode: 2.735Hz

1st lateral mode: 0.845Hz

2nd lateral mode: 1.411Hz

3rd lateral mode: 1.866Hz

Figure 7.10 Identified vertical and horizontal modes: natural frequencies and mode shapes.

A bridge FE model made by Hoskere et al. (2019) in *ABAQUS* is employed as the initial FE model as shown in Figure 7.11, containing steel cables, wooden decks and bridge towers. The three-dimensional (3D) model consists of 8220 nodes and 2016 3D beam elements with six degrees of freedom at each node. The boundary conditions for the support of bridge towers are assumed to be fixed. However, this model has not been updated, and it does not represent the actual conditions of the bridge. The dynamic properties from the initial model and the measured data are compared in Table 7.4. Their differences are considerable, especially for horizontal modes which are dominant modes during bridge/ship impacts. The differences are mainly attributed to two aspects: 1) the corrosion of the bridge is severe, resulting in stiffness reduction; 2) several modeling details are not sufficiently considered, such as boundary conditions of the braces, which will be addressed by model updating and discussed in the following paragraph.

Table 7.4 Comparison between the dynamic properties from the initial FEM and measurements.

Modes		Measurement (Hz)	Initial finite-element model (Hz)	Error (%)
Vertical modes	2	0.652	0.501	30.09
	3	1.073	1.045	2.68
	5	2.735	2.738	-0.11
Horizontal modes	1	0.845	1.290	-34.48
	2	1.411	2.009	-29.77
	3	1.866	3.328	-43.93

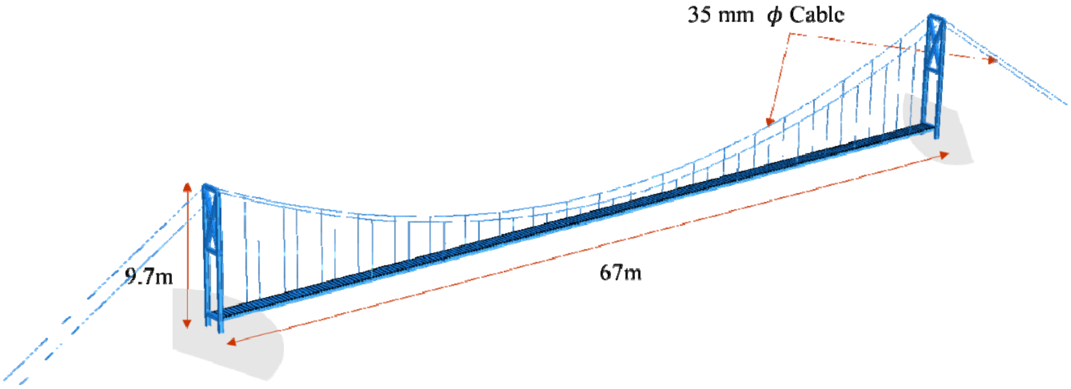
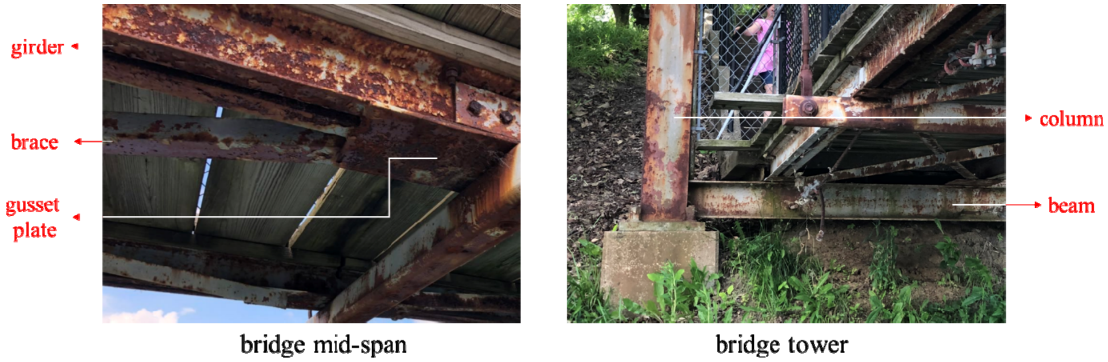


Figure 7.11 Finite element model for the pedestrian suspension bridge.

As shown in Figure 7.12, severe corrosion was found in various components of the bridge, such as main girders, braces, gusset plates for connections, beams and columns in the bridge tower. The corrosion is more severe in the mid-span, beneath the deck, as it is more moisture, closer to the water surface, and normal maintenance is not easy. The corrosion observed in most of the thin-walled bridge components is uniform corrosion or shallow-wide trough pitting corrosion. In this scenario, steel mass doesn’t change a lot, as most of the rust is still adhered to the bridge surface; the bridge stiffness, however, may significantly decreases, due to the reduction of effective cross section. Accordingly, the main strategy of model updating is to reduce Young’s modulus of the bridge components, aimed at reducing stiffness but remaining mass. The reduction is more in the mid-span of the bridge, especially for the main girders and braces.

In addition, special attention is paid to improve model details. First, the boundary conditions of the main longitudinal girders are enhanced. In the preliminary model, the girders are just supported by steel beams in the bridge towers in both ends which have relatively low rotational stiffness. In the real structure, rotational stiffness is expected to be higher, because the girders are buried in the soil in one end, and they are connected to stairs in the other end. Therefore, the height of the support beam in the bridge towers is increased to incorporate the additional rotational stiffness for girder ends. Second, the boundary conditions of braces are weakened. Preliminary, the boundary conditions for braces are simplified to as fixed joints. However, in reality, the ends of braces are connected to the gusset plates by two bolts, which are weaker than the fixed joints. Also, these gusset plates and bolts are considerably corroded, further weakening the boundary conditions. Accordingly, the Young's modulus of the braces is reduced to achieve the equivalent behavior of boundary conditions, without introducing model complexity.

Besides of the two main points described above, other details of model updating are not discussed, as they are not the main focus of this study. Finally, the modal properties from the updated model is listed in Table 7.5 and Figure 7.13. The result of modal analysis using updated FE model agrees well with the measured data, demonstrating that it can be used for bridge impact analysis.



bridge mid-span

bridge tower

Figure 7.12 Corrosion of the bridge in mid-span and bridge tower.

Table 7.5 Comparison between the natural frequencies from the updated FEM and measurements.

Modes		Measurement (Hz)	Updated finite-element model (Hz)	Error (%)
Vertical modes	2	0.652	0.634	2.84
	3	1.073	1.139	-5.79
	5	2.735	2.685	1.86
Horizontal modes	1	0.845	0.847	-0.24
	2	1.411	1.477	-4.47

The updated model is then employed for simulation of bridge/ship impact, which is a highly nonlinear dynamic process. In current study, the performance of a boat is not the focus. So, it is simply modeled as a cube with 7.6 cm long to represent the superstructure of a boat impacting the bridge. In contrast to the Jumbo Hopper barge which is used as a standardized ship for bridge/ship impact analysis in AASHTO Guide Specification, the boat considered in this study is the Sundolphin 4 seat pedal boat. This type of boats is commonly used for boat rental in the Lake of Woods. If fully loaded, the boat has an approximate mass of 205 kg. Accordingly, the density of the cube model is set to be 1195 kg/m^3. Young's modulus and Poisson's ratio of the boat material are 206.9GPa and 0.3, respectively. Solid elements are used to model the boat.

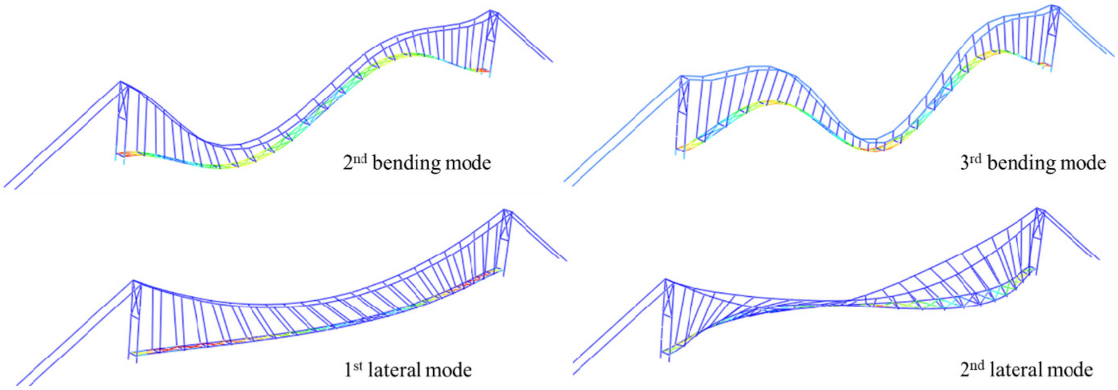


Figure 7.13 Updated FE model dynamic mode shapes.

The bridge model consists of several components: (1) suspension cables are modeled using truss elements; (2) bridge beams and columns are modeled using beam elements; and (3) wooden decks are modeled using solid elements. Their material parameters are listed in Table 7.6. To consider material nonlinearity, bilinear stress-strain curve is employed for steel components. Because wooden decks are nonstructural components, so their damage is not considered in impact analysis, and they are modeled as linear materials. In addition, to better represent the bridge vibration, damping ratio is determined by employing Rayleigh damping assumption. Considering both 2nd and 3rd bending modes have a 3% modal damping factor, the associated damping parameters can be determined: α=0.15289, and β=0.00554.

Table 7.6 Material properties of suspension bridge model.

Component	Parameter	Value
steel components (cable, beam, colum)	ρ_s = mass density E = Young's modulus v = Poisson's ratio $\sigma_{y\text{-cable}}$ = cable yielding stress $\sigma_{y\text{-comp}}$ = other components yielding stress	7910 kg/m^3 206.9 GPa 0.30 1850 MPa 345 MPa
wooden decks	ρ_s = mass density E = Young's modulus v = Poisson's ratio	400 kg/m^3 12.0 GPa 0.30

The contact between the bridge and the boat is defined using standard surface-to-surface contact. The contact area in the bridge model is defined in the longitudinal girders. The impact force is estimated by multiplying the stress value in the boat and cross section area of the boat. Various initial boat speeds give rise to different values of peak impact force. The initial boat location is adjusted in each analysis, such that the boat always hits the bridge at 0.1s. Three-axis accelerations are recorded at a total of 10 points on the bridge in the impact analysis. The detailed information regarding impact analysis can be seen in Figure 7.14. An example of acceleration record at 10 measurement points is shown in Figure 7.15, in which the boat hit the bridge at the speed of 2.6 m/s at the distance of 16.95 m to the right end. Y axis in the record is the direction in line with bridge span; whilst, X axis and Z axis in the record are the main vibration directions perpendicular to the bridge span in horizontal and vertical, respectively.

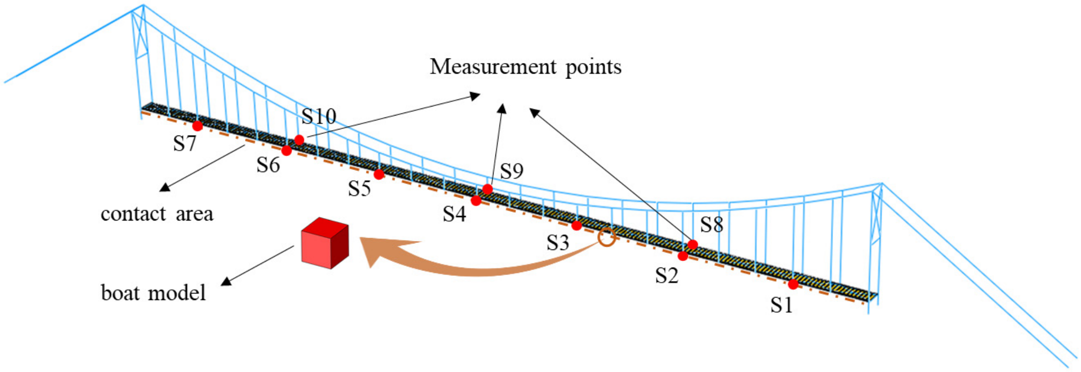


Figure 7.14 Bridge/boat impact analysis illustration.

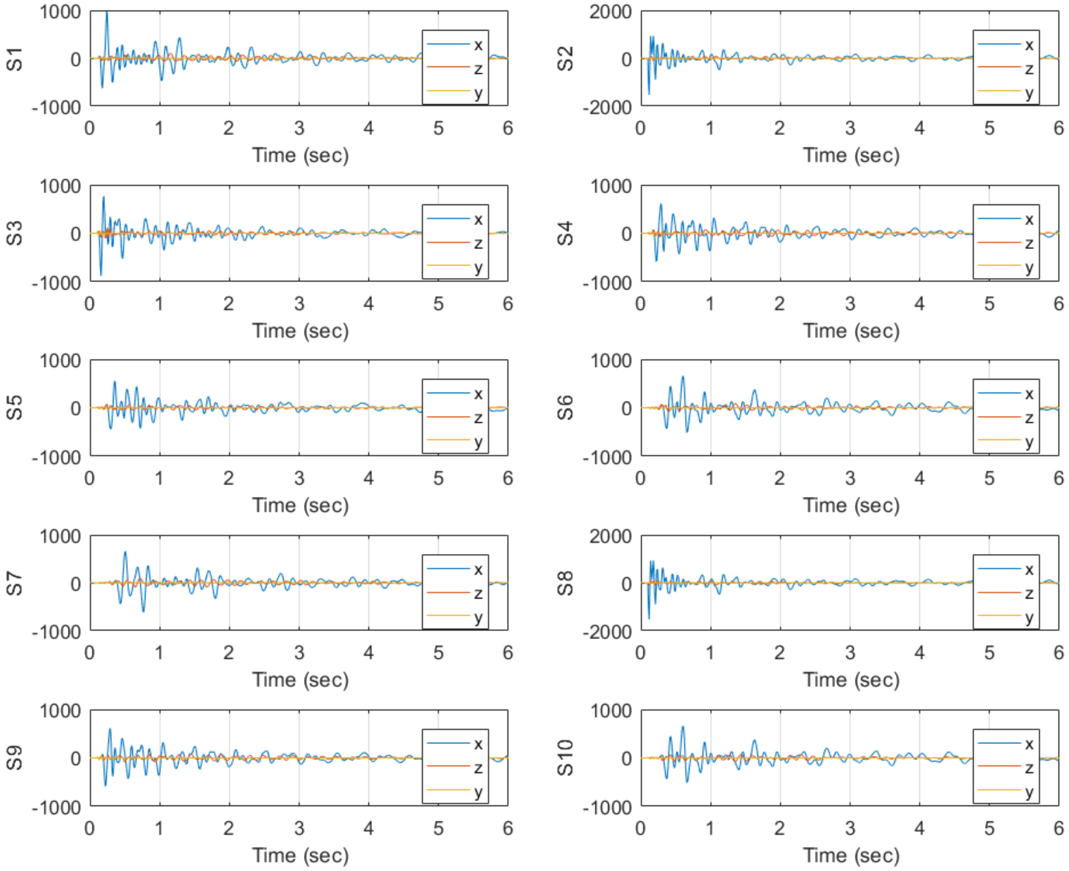


Figure 7.15 Acceleration time history record during impact analysis.

The bridge impact simulation is employed for two purposes: 1) estimating the maximum allowable impact forces (peak force and impulse) that can serve as threshold conditions for rapid damage assessment; 2) generating measurement records to train neural network which can be further used to identify impact locations and forces. The first part is described in this section, whilst the second part will be discussed in Section 7.3.4.

After we detect and estimate the impact force value (peak force and impulse) and location, the impact force value is then compared against the limit record obtained by impact simulation at the corresponding location. If the impact force value exceeds the limit record, we consider the structural is damaged, and an alert message will be sent to maintenance technicians to further check bridge conditions at the impact location.

To obtain the maximum allowable values of impact force/impulse, a series of nonlinear impact analysis is performed at various locations of the bridge. In particular, the positions of impact analysis include all the brace joints (collocated with suspenders) and the intermediate midpoints between joints (located on the girders). Changing the initial speed of boat in each impact analysis results in different impact forces. The force that give rise to the maximum bridge stress of equal to steel yielding stress is considered as the maximum allowable impact force. The corresponding impulse is considered as maximum allowable impulse. Assuming that the bridge is symmetric about its midspan, only the right half span of the bridge is investigated, and the results are collected. The maximum allowable impact force/impulse varies at different locations of the bridge, and the summary of these values is presented in Figure 7.16. The impact locations are divided into two categories: suspenders and main girders, because they have different local stiffness and hence different responses. In particular, suspenders have larger local stiffness than girders. If the impact is collocated with the suspenders, the peak impact force is larger than adjacent girders; however, the contact time between boat and the bridge is shorter, which results in smaller impulse than adjacent girders.

Figure 7.17 shows the bridge stress distribution under an impact that occurs at the distance of 41.15m from the midspan. The boat has an initial speed of 2.6 m/s. It can be revealed that, the braces are the most critical components which have highest stress values, because they are the main contributor for lateral resistance. Generally, the braces are considered as secondary structural components; if they are damaged, the bridge still has the capabilities to sustain its main functionalities. Therefore, the early warning of braces damage can provide maintenance crew enough time to make emergency response to prevent overall failure of the entire structure.

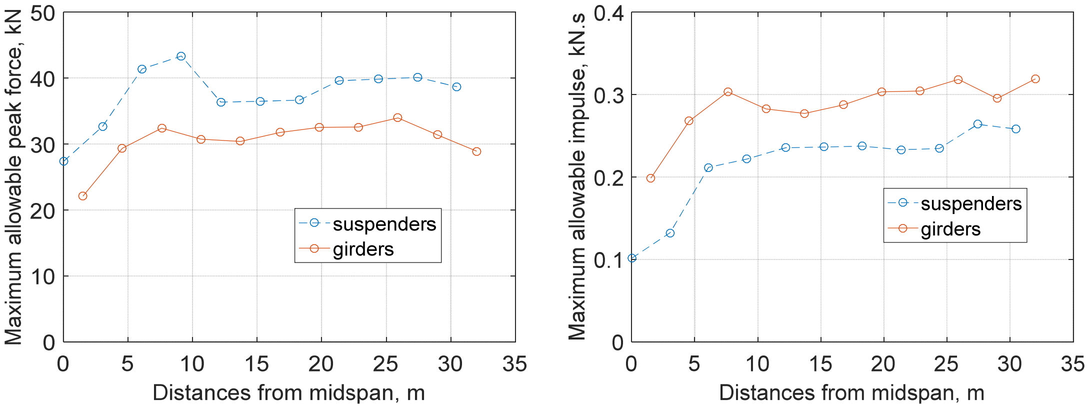


Figure 7.16 maximum allowable impact force & impulse for the right half span of the bridge.

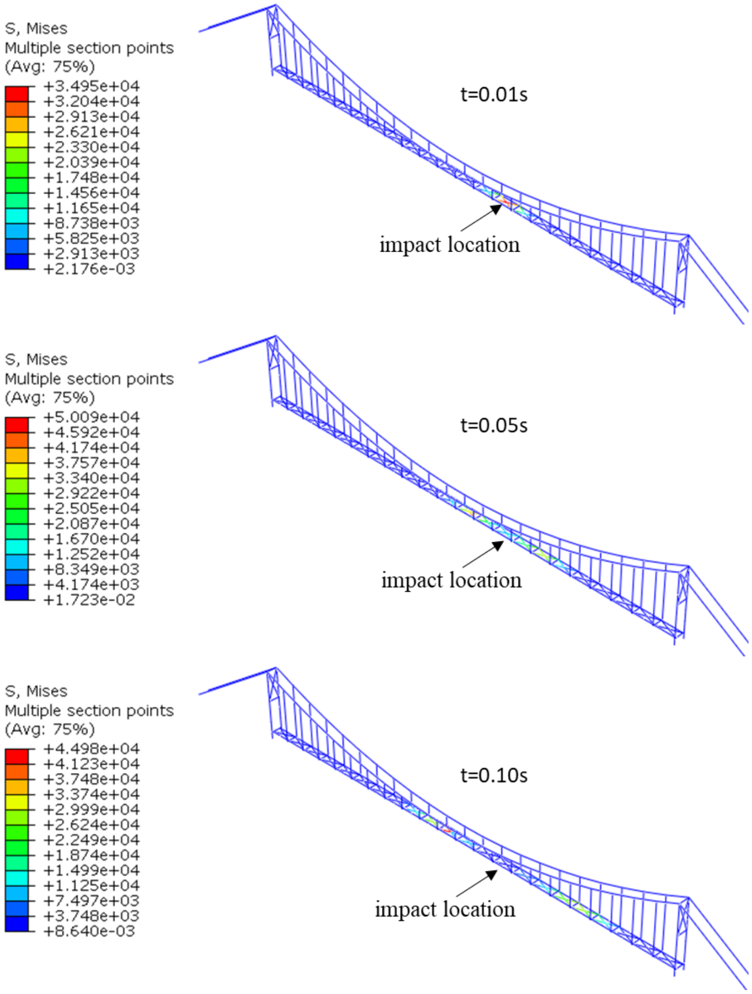


Figure 7.17 Stress distribution on the suspension bridge under an impact.

7.3.4 Neural Network Modeling

An artificial neural network is established to process impact measurement data and identify impact location and impact force information, which can be further used for rapid condition assessment of bridges under impact events.

Vibration data collected at 10 locations on the bridge via numerical simulation was employed to train and test the ANN. In the numerical simulation, the boat model hits the bridge horizontally at a random location and at a random speed. In particular, the impact analysis is only conducted within the right half span of the bridge, by taking advantage of bridge symmetry. The initial boat speed is set to be a random value between 0.25 m/s and 3.05m/s. Boat impacts with varying initial speed result in different impact forces at different locations. The specific speed range is intentionally configured to include both linear (undamaged) and nonlinear (damaged) scenarios in the database. Three-axis acceleration recording is collected at 100Hz for each measurement point, with a total length of 6 seconds. During impacts, the force history is also recorded to calculate peak impact force and impact impulse. The initial location of the boat model is adjusted to ensure that the bridge/boat collision starts at 0.01s. A total of 16895 data sets with varying impact locations and initial boat speeds are obtained through a series of numerical simulation, serving as the learning database for ANN training and testing. The impact scenarios are illustrated in Figure 7.18.

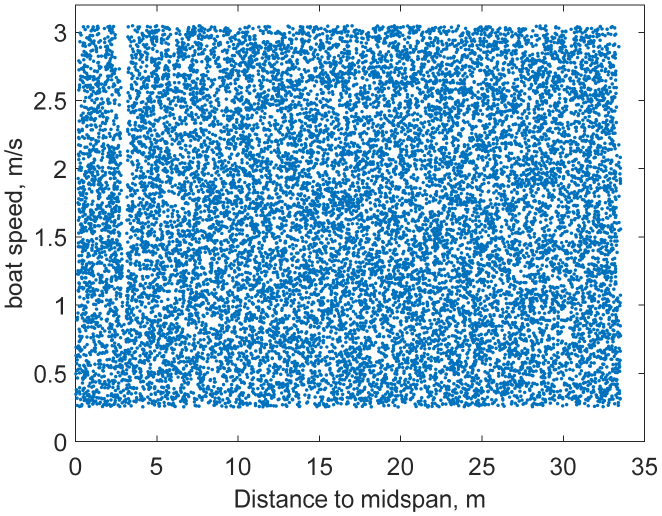


Figure 7.18 Learning database for ANN training and testing.

Following the impact signal processing strategies developed for aircraft structures (Sharif-Khodaei et al., 2012; Ghajari et al., 2013), important features are extracted from the bridge response measurements as inputs for the ANN model. Specifically, features for impact location detection include: (1) time of arrival of vibration signals (ToA), (2) maximum acceleration record, (3) time at maximum acceleration record, (4) maximum envelope of acceleration record, and (5) time at maximum envelope of acceleration record. These features are considered to contain adequate information of impact location. For example, after an impact occurs, similar to wave propagation, the induced vibration evolves earlier at the measurement point that is closer to the impact location. Accordingly, the arrival time of vibration signals is employed for impact location detection, and it is defined when the absolute acceleration first exceeds 40mg in this study. Specifically, ToA at measurement point *i* is expressed as,

$$ToA_i = t_i - \min(t) \tag{7.1}$$

which is the offset between the arrival time at point *i* and the smallest arrival time within all the points. In addition, the impact energy distributes unevenly, indicating that the maximum absolute acceleration record is larger if the measurement point is closer to the impact location. Therefore, both maximum acceleration record and maximum envelope of the acceleration record are calculated for impact location detection. To obtain signal envelope, a Hilbert transform is performed for each acceleration record *a(t)*,

$$H(a(t)) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{a(\tau)}{t-\tau} d\tau \tag{7.2}$$

The real part of the transformed result is the original real data, whilst the imaginary part is the actual Hilbert transform. The magnitude of the transformed result is the envelope of acceleration record. Though the acceleration record contains three-axes measurement data, only data in vertical axis and lateral axis is processed to generate features, respectively; the data in longitudinal axis has very small amplitude and hence it is neglected. Therefore, a total of 100 features obtained from 10 measurement points at two axes are used as inputs for the ANN to detect impact locations.

Features for impact force identification include: (1) maximum acceleration record, (2) time at maximum acceleration record, (3) the maximum value of detailed coefficients after discrete wavelet transform (WT) using level 4 Daubechies wavelet (db4), (4) the maximum value of approximated coefficients of WT (db4), as well as (5) detected impact location. The first four types of features have been considered effective for impact force reconstruction in the paper (Ghajari et al., 2013). They are extracted from 10 measurement points at vertical axis and lateral axis, respectively. The last feature is the identified impact location from the ANN, which can help to increase the accuracy of impact force estimation. In sum, a total of 81 features are extracted for impact force estimation which include peak impact force and impact impulse. The summary of inputs for impact information identification is listed in Table 7.7.

Table 7.7 Features extracted for impact location identification and force estimation.

No.	Feature	Usage	
		Location detection	force estimation
1	time of arrival of vibration signals	√	
2	maximum acceleration record	√	√
3	time at maximum acceleration record	√	√
4	maximum envelope of acceleration record	√	
5	time at maximum envelope of acceleration record	√	
6	maximum value of detailed coefficients (WT-db4)		√
7	maximum value of approximated coefficients (WT-db4)		√
8	identified impact location		√

The neural network model is established using multi-layer perception in Keras using python language. The ANN model contains two subnetworks: (1) one for impact location detection, and (2) one for impact force estimation. In particular, the first subnetwork has one input layer, several fully connected hidden layers and one output layer. The result from the output layer of the first subnetwork is subsequently used as the input for the second subnetwork. The second subnetwork has similar architecture with the first one. But the last hidden layer of the second subnetwork is split into two parts, one for max impact force estimation and one for impact impulse estimation, respectively. Between adjacent layers, the neuron behaviors were defined and tested using different activation functions (e.g., ReLU and sigmoid). The loss function is defined using mean squared error, and the metric function is defined using mean absolute error to judge the performance of the trained ANN model. The two subnetworks are trained independently. Because the second subnetwork has two different outputs which have different scales, the weight value of loss function is defined for each output. Considering that the mean value of peak impact force is around 3500 times larger than that of impact impulse, the loss function is expressed as,

$$Loss\ function = 1 \cdot MSE_{pf} + 3500 \cdot MSE_{m} \tag{7.3}$$

where MSE_{pf} is the mean squared error of peak impact force; MSE_{m} is the mean squared error of impact impulse. To identify an optimal ANN, different topologies of the ANN are trained and tested, containing various numbers of hidden layers as well as various hidden neurons per layer. In the meantime, the learning database is split into three segments: 50% for training, 25% for validation, and 25% for testing. The finalized network architecture is shown in Figure 7.19, which achieves the best performance of testing. Particularly, the first subnetwork has two hidden layers, one with 32 neurons and the other with 16 neurons; the second subnetwork has three hidden layers, the first two layers has the same architecture with the first subnetwork, whilst the last layer is divided into two 4-neuron parts. In addition, the RELU is employed as the activation function for all the layers.

Figure 7.20 presents the learning curves of two subnetworks in the finalized ANN model, both of which converge to a very low rate, demonstrating that the number of epochs is adequate to achieve high accuracy. The accuracy of the trained ANN is expressed through mean absolute error and summarized in Table 7.8. The error is satisfactory for rapid condition assessment of bridges.

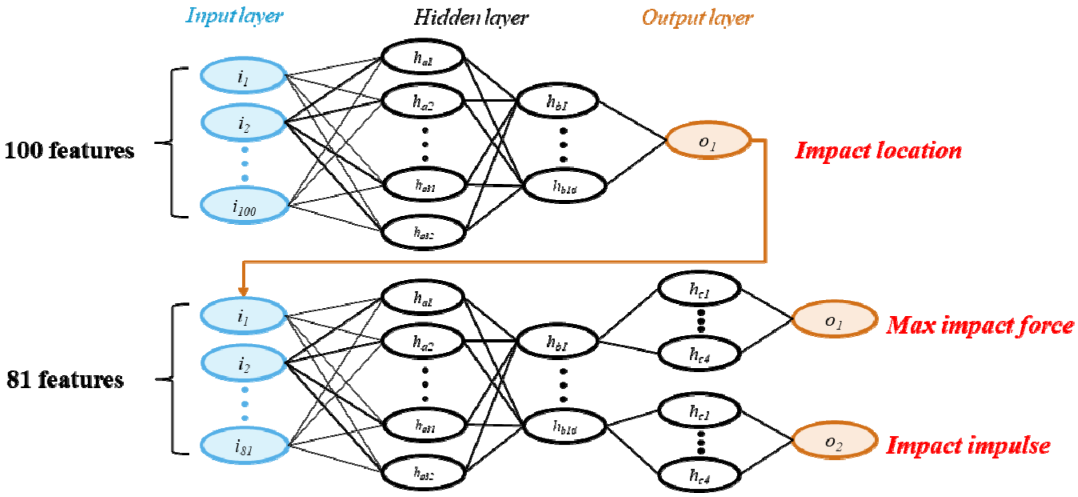
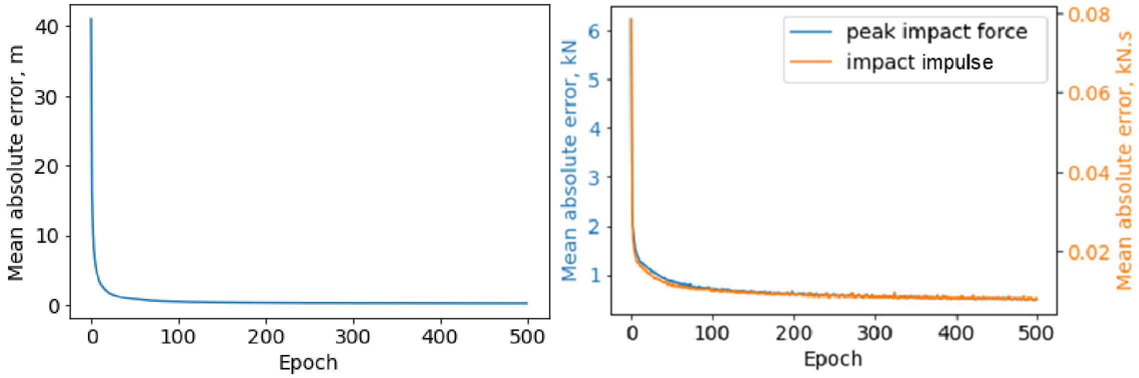


Figure 7.19 Artificial neural network model for impact analysis.



(a)

(b)

Figure 7.20 Artificial neural network learning curves: (a) impact location, (b) impact force.

Table 7.8 Neural network model testing results.

Impact Info	Training data sets		Testing average error
	Mean	Std	
Location (m)	50.45	9.55	0.28
Peak force (kN)	11.17	8.56	0.54
Impulse (kN.s)	0.25	0.12	0.01

7.3.5 Full-scale Demonstration

To validate the capabilities of developed system and the efficiency of proposed decision-making framework, a full-scale demonstration is performed in the pedestrian bridge in Lake of the Woods in Mahomet, Illinois, as described in Section 7.3.1. The impact is generated by a large-sledge impulse hammer in horizontal direction. The hammer, model PCB086D50 (PCB Piezotronics, Inc., Depew, NY, USA), is equipped with a force sensor on the tip.

The monitoring system is deployed on the bridge, as shown in Figure 7.21. A total of 10 *Demand-based WSSs* were installed on the beam joints beneath the bridge. They were attached to the steel beams using magnets. Another wireless sensor is deployed far away from the bridge, serving as the base station. The threshold for event-start detection was configured to be 40mg over 0.02s; the threshold for event-stop detection was set to be 10mg over 5s. In addition, the sampling rate for high-fidelity measurement was 100Hz. For comparison, six uniaxial wired accelerometers, model PCB353B33, were selected as reference sensors and collocated with wireless sensors on the right half-span bridge. They were amounted horizontally on the enclosure of wireless sensors using hot glues. Both wired sensors and the force sensor on the impact hammer were connected to the same DAQ system mentioned in Section 7.2.1. The sampling rate was 8192Hz, aimed to capture transient peak of force sensor signal. The measurement data from wired sensors was later decimated to 100Hz for comparison. In addition, the measurement mode of the DAQ system was pulse-triggered mechanism, i.e., the system started measurement when it detected a pulse like signal in the force sensor.

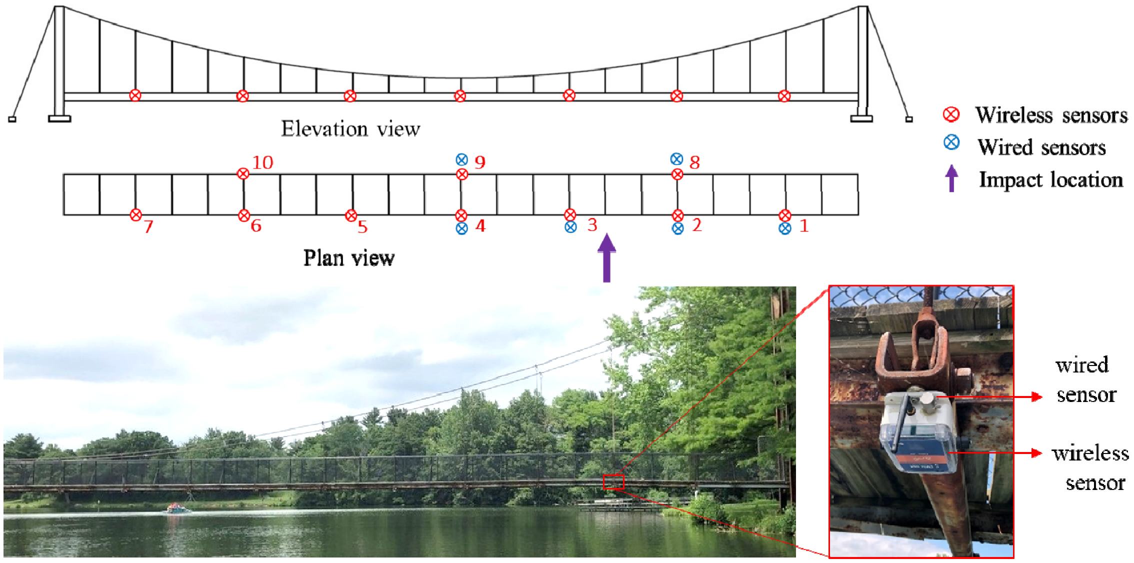
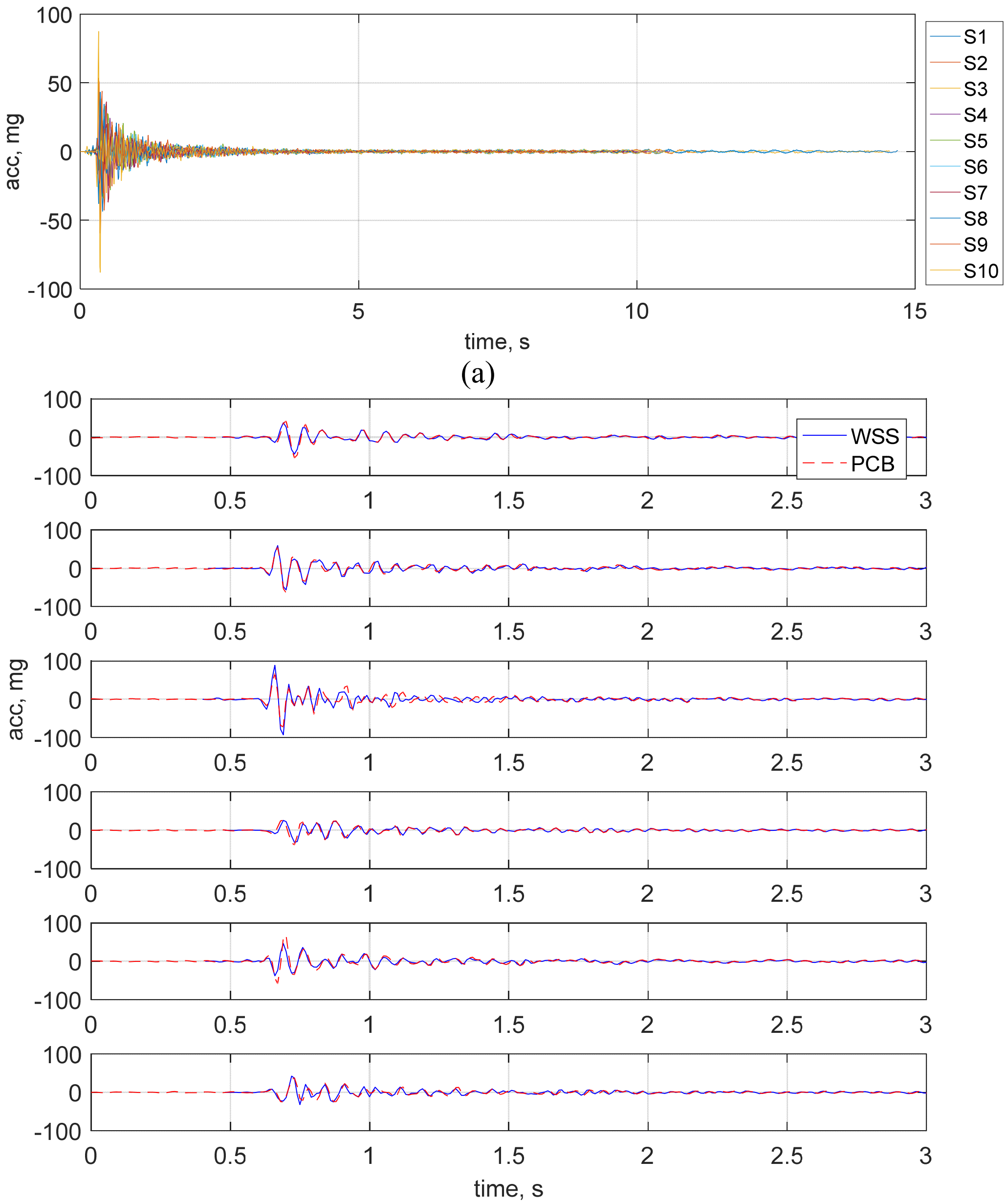


Figure 7.21 Sensor node deployment for bridge impact monitoring.

After the impact occurs, *Demand-based WSS* woke up and started measurement. In the meantime, the DAQ system was triggered to collect data from wired sensors and the impact hammer. Figure 7.22 shows the comparison between the bridge responses from the developed system and wired sensors. As can be seen, the critical data has short-duration of less than 3 seconds. The developed monitoring system successfully captured the onset of impact events and obtained the complete bridge responses. And the measurement data matches very well with the wired sensor data, demonstrating that the developed system provided high-quality synchronized data.



(b)

Figure 7.22 Impact response measurements: (a) vibration responses from WSS, (b) measurement comparison for node 1, 2, 3, 4, 8, 9.

The obtained measurement data is preprocessed, and extracted features are then fed into pretrained ANN model. Impact information is identified and summarized in Table 7.9. The error is slightly larger than the testing error in Table 7.8, but they are satisfactory and in a reasonable range. For example, the distance between two suspenders is 3 meters, if the error of impact location detection is 2 meters, it is still helpful for bridge owners to identify the location between two suspenders. In addition, the impact force is much smaller than the maximum allowable impact forces, indicating that, no need to send warning message to upset bridge owners though an impact event occurs.

Table 7.9 Artificial neural network prediction results.

Impact Info	Test	Estimation	Error=|Test-Estimation|
Location (m)	45.72	47.76	2.04
Peak force (kN)	3.13	4.07	0.94
Impulse (kN.s)	0.03	0.04	0.01

7.4 Summary

This chapter presents the integrated monitoring system, referred to as wireless intelligent sudden-event monitoring system (WISEMS). The system can enable rapid condition assessment of structures in two different schemes: online monitoring scheme and offline monitoring scheme. Particularly, its five main components are introduced. The integrated system has been successfully applied to two applications, seismic building monitoring in a lab test and bridge impact detection in a field test. The results demonstrate that the integrated system can not only send early warning of the sudden events to engineers for emergency response, but also enable rapid condition assessment of civil infrastructures using high-quality synchronized measurement data for decision making.

CONCLUSION AND FUTURE STUDIES

8.1 Conclusion

This dissertation presented an effective wireless solution for rapid damage assessment of civil infrastructure subjected to sudden events. The solution integrates a series of key components, involving hardware design, software development, and digital signal processing. Particularly, *Demand-based WSS* prototype is designed to enable sudden event detection. Efficient time synchronization strategies are proposed to achieve high-precision synchronized data measurement. A high-throughput live streaming framework is built to realize real-time data acquisition. Three-stage sensor fault management strategy is developed to ensure high-quality data for reliable informed decision making. Finally, a MATLAB-based application is presented to support real-time damage assessment and data visualization. The integrated system, WISEMS, is validated through a lab test of seismic building monitoring and a field test of bridge/ship impact detection.

A comprehensive investigation of research background in structural health monitoring is carried out, with a particular focus on the state-of-the-art smart technologies and associated challenges that are essential for sudden-event monitoring, including wireless sensor platforms, time synchronization, data acquisition, sensor fault management, and online condition assessment. Wireless smart sensors have the potential to reduce cost and replace existing prohibitive wired systems for sudden-event monitoring. However, this solution remains elusive, until several critical obstacles are fully addressed. In particular, wireless sensors are generally duty-cycled to preserve a limited battery energy, so they will miss events when they are in power-saving sleep mode. Time synchronization is critical for WSSN, but existing strategies are not applicable for sudden-event monitoring, in the face of certain challenges such as unpredictability of sudden events, data loss of the initial transient response, and temperature variation during sensing. After synchronized measurement data is collected, rapid damage assessment using the data is required for emergency response or informed decisions. To this end, response latencies must be minimized. However, the wireless communication throughput is strictly limited, so data transmission is generally time-consuming. In addition, sensor malfunction is a main concern for WSSN, and conventional faulty data management requires user interaction before rapid damage assessment. Furthermore, very few efforts have been made to support real-time damage assessment and data visualization for WSSNs. These identified challenges draw a roadmap for the following research tasks in this study.

To overcome the stringent energy constraint of wireless smart sensors, an ultralow-power on-demand sensing prototype, *Demand-based WSS*, is developed. It leverages a high-fidelity sensor platform (Xnode) as a host device, and its key part is a programmable event-based switch. The switch employs a trigger sensor to power on/off the sensor platform when an event start/stop, respectively, and it uses a real-time clock to record the onset time of events. The prototype has a current draw of 170mA when it is turned on, but only 365 μA when it is shut down. Considering that sudden events are rare and short-duration, most of time the prototype is powered off. Therefore, employing the proposed WSS can extend the lifetime of always-on monitoring from 3 days to over 3 years using a single 10,000 mAh lithium battery. In addition, the trigger sensor data and high-fidelity sensor data are synchronized and fused to produce a complete acceleration record, addressing the issue of initial data loss due to cold booting delay.

To address unique challenges of time synchronization in a network of *Demand-based WSSs*, several efficient time synchronization strategies have been developed for different use cases. After examining the Xnode's clock behavior, a post-event time synchronization strategy is first developed for most scenarios (e.g., earthquakes). It employs reference node election for relative clock synchronization and postpones the time synchronization operation until event-triggered sensing is finished, hence, introducing zero sensing delay. However, the first time synchronization strategy does not perform well for long-duration events (e.g., downbursts) in which nonlinear clock drift is not negligible. Therefore, a variant version, called pre-post-event time synchronization, is developed, in which two rounds of point synchronization are conducted before and after event-triggered sensing. For real-time monitoring, the previous time synchronization strategies are no longer practical, due to their offline nature. Therefore, a piecewise real-time time synchronization strategy is proposed, in which time synchronization is conducted periodically during event-triggered sensing. These time synchronization strategies can achieve high-precision synchronized sensing with maximum error of less than 20 μs.

To minimize the latency of data transmission in WSSN during sudden-event monitoring, a live steaming framework is built to enable high-throughput data acquisition in real-time. First, in each sensor node, preemptive multitasking is implemented to address the scheduling conflicts between sensing process and data transmission. Second, to allow multiple sensor nodes to transmit data to the base station in real time, adaptive Staggered Time Division Multiple Access (S-TDMA) protocol is developed, in which local clock in each sensor node is resynchronized periodically (e.g., every 60s) to keep the S-TDMA stable. Lab tests demonstrate that the live streaming framework can achieve 100% packet reception rate over 10-minutes measurement, with a high throughput of 115.2 kbps. Furthermore, to enable real-time damage assessment of structures, a MATLAB-based application is developed for two purposes: (1) processing the data collected through the live streaming framework for damage assessment; (2) plotting measurement data and results in real-time. Particularly, two efficient online condition assessment strategies are developed and implemented in the application, including interstory drift estimation using acceleration record and sudden damage detection based on WT-ICA.

Sensor fault data management is also studied to ensure high-quality data for informed decisions in an automatic manner. Sensor faults are found to have a significant effect on power spectral density, making the faulty data distinguishable from fault-free data in the neighborhood. Therefore, a three-stage strategy is proposed, including faulty data detection through distributed similarity tests, fault type identification using artificial neural network, and recovery of different types of faulty data by applying a correction function or replacing faulty data with estimated values. A case study is performed to validate the capabilities of the proposed strategy, demonstrating that it can successfully detect, identify and recover sensor faults.

Finally, a wireless intelligent sudden-event monitoring system (WISEMS) is presented, incorporating the above techniques which are involved with hardware, software and signal processing. The main objective is to provide an integrated system as a wireless solution for sudden-event monitoring of civil infrastructure. The system can be configured to work either in an offline manner or in an online manner, depending on use cases. To demonstrate the capabilities of the WISEMS, two applications are performed: (1) seismic monitoring of a 6-story building model in online manner, and (2) bridge/ship impact detection of a pedestrian bridge in offline manner. The results demonstrate that, the developed system can capture the occurrence of sudden events, provide high-fidelity actionable information for emergency response and maintenance decisions in an efficient manner.

8.2 Future studies

The study has explored many research areas and addressed associated challenges towards the development of wireless solution for sudden-event monitoring. Based on the results of current methodologies and the remaining challenges, several new topics for future studies are identified, which will be discussed below.

8.2.1 Multi-source Triggered Monitoring

To enable event-driven monitoring under sudden events, the *Demand-based WSS* developed in this study is triggered by motion, leveraging a trigger accelerometer. In some cases, event of interests may not give rise to large structural vibrations, e.g., heavy snow, flood, and landslide. Therefore, multi-source-triggered prototypes are desired for a broader range of SHM applications.

One attractive feature for multi-source-triggered prototypes is radio-triggered monitoring. Particularly, each sensor node is equipped with an off-the-shelf ultralow-power wake-up radio. Similar to the current *Demand-based WSS*, sensor nodes are powered off, whilst wake-up radio is always-on to receive external radio messages. The nodes can wake up immediately, once user-specified messages are received. The benefits of radio-triggered monitoring include three main aspects: (1) the prototypes can make quick response to user inquiry or gateway commands in any time; they can thus make the best of limited battery power supply, leaving duty-cycling strategy completely obsolete; (2) instead of merely relying on user-defined thresholds, the prototypes can be triggered by external systems which can have more complex and more reliable triggering conditions. A typical example is to build a connection between existing Earthquake Early Warning system and radio-triggered sensing prototypes for seismic structural monitoring; (3) because of various vibration levels in a structure, it is possible that some *Demand-based WSSs* will not wake up, when a sudden event occurs. In this scenario, radio triggering can serve as a supplementary solution to wake up all the nodes. Other interesting features of multi-source-triggered prototypes include tilt-triggered sensing for landslide monitoring, temperature-triggered sensing for fire monitoring, and acoustic-triggered sensing for acoustic event detection.

In addition, the on-demand sensing prototype can be triggered by external integrated circuits. In the aforementioned examples, prototypes are triggered by signals directly from ultralow-power trigger sensors. Though this type of triggering mechanisms is straightforward, they are sometimes not reliable, with a relatively high false-positive rate. The false-positive scenario is generally acceptable, if users only initiate monitoring system in response to it. But they may result in serious consequence, if users employ it both for sensing and structural/machine control. To address this concern, the cutting-edge integrated circuits (e.g., field-programmable gate array and application-specific integrated circuit) can be used, in which users can program complex algorithms on them. The circuits can read and process measurement data continuously from ultralow-power always-on trigger sensor, and finally wake up the prototype if the obtained results indicate the events of interests.

8.2.2 Network Adaptation and Optimization

The software development in this study is built on preemptive multitasking framework using FreeRTOS. The framework enables several benefits for the development of efficient and multi-functional WSSNs, in part due to its real-time scheduler. More details discussion is presented below, focusing on wireless network adaptation and optimization.

(1) Network repurposing

In a large-scale SHM applications, WSSN usually consists of hundreds of sensor nodes. During their service for extended periods, however, reprogramming WSSs are required for a variety of reasons, such as updating algorithms, changing application purposes and fixing software issues. It is extremely costly to reprogram all sensor nodes by manually connecting each node with a PC or external programming devices. Therefore, over-the-air (OTA) programming is highly imperative for WSSN. Many efforts have been made to achieve OTA programming, which however are mainly based on TinyOS. Two main concerns must be addressed: significant software development efforts for OTA service, and high battery power consumption during OTA process, especially for large-scale WSSNs.

With the aid of developed software framework, the solution of OTA service scalable for large-scale network can be envisioned. In particular, the framework provides an isolation of individual tasks/services and hence a separation of concerns; programming efforts can also be reduced, using the standardized C programming language. Specifically, the OTA can be realized through three main steps: In-system programmer to load the program code into flash, reliably distribution of application image to all nodes leveraging *RemoteCommand* service, and distributed consensus/two-phase commitment for all nodes to agree to switch to the new image at once.

(2) Network topology management

In long-term SHM using WSSN, sensor nodes are expected to work autonomously, and malfunctioning of sensors are likely to occur frequently and unexpectedly, due to issues with battery, harsh environment or calibration errors. There are two levels of malfunctioning: sensor fault (sensors produce abnormal data) and sensor failure (sensors are inactive). The first aspect has already been investigated in this research, but the second aspect has not been studied. Sensor failure is critical, because it may either break the multi-hop network topology, or disrupt the decentralized data acquisition process, especially when a cluster head fails.

Therefore, sensor failure management is very important, which can be organized into three phases, 1) failure detection, 2) failure diagnosis, and 3) failure recovery. The third phase is the most challenging, as it is involved with optimization of radio communication, measurement information, and sensor node resource redistribution (e.g., battery power and memory). The challenge can be resolved using state-of-the-art optimization strategies. The failure of cluster heads in decentralized WSSN can be detected by implementing heart-beating protocols. If failure of a cluster head is detected, consensus-based bully election will be applied within the cluster to select one leaf node as a new cluster head (CH). In addition, using the flexible software framework, developers can configure the system with dynamic resource allocation mechanism, hence resources can be allocated and deallocated at run-time. In this way, highly efficient operation is enabled for long-term structural health monitoring, especially for a large-scale wireless sensor network.

8.2.3 Towards Rapid Damage Prediction

The ultimate objective of this research is to enable rapid damage assessment using the event-driven measurement data from the developed system. The early detection of structural damage can help to both make emergency response and mitigate extended maintenance and downtime costs. In some cases, initial structural damage may result in collapse in seconds. For example, most of structural failures under blizzards involve instability problem, which usually initiated from individual member buckling and then developed into progressive collapse in seconds. The entire process can only take very short time (e.g. seconds). Even if initial member buckling is detected in timely manner, maintenance crew may not have enough time to make emergency response to prevent overall failure of the entire structure. Therefore, smart technologies towards rapid damage prediction is desired to deal with this special scenario.

One possible solution is to calculate structural damage probability by incorporating risk and reliability analysis. Since structural damage initiation is generally under sudden events, event-driven measurement data from the developed system can still be leveraged as the data base for probability analysis. Subsequently, initial structural condition assessment is performed to confirm whether structural damage occurs or not. If so, warning messages must be sent to the maintenance crew. Otherwise, structural condition information at the moment (e.g., modal properties) is saved in a database. Afterwards, reliability analysis algorithms (e.g., first-order reliability method) are processed to calculate probability of structural damage considering material and geometric uncertainties. Based on the obtained results, an alert will be sent prior to the onset of structural damage.

8.2.4 Broader Applications of Sudden-Event Monitoring

The intelligent system in this study has been successfully implemented for many sudden-event monitoring, including seismic building monitoring, train-crossing event monitoring, and bridge/ship impact detection. In addition, the developed system is versatile and applicable for a broader class of applications. The detailed discussion is presented as below, regarding application objective and associated remaining challenges.

(1) Bridge/vehicle strike

Bridge/vehicle strikes are identified as critical sudden events for railway systems. Transportation Technology Center, Inc. conducted a study upon request of the FRA (Joy et al. 2013), finding that half of the railroad service interruptions reported were caused by collisions with bridges. The railroad infrastructure manager in the U.K., Network Rail (2017), reported nearly 30,000 bridge strikes between 2000 and 2017. In 2017 alone, two or more strikes were reported for 353 bridges, and eleven bridges experienced over 10 over-height collisions. Therefore, the development of monitoring system is critical to detect, quantify, and report these events.

The initial attempts for bridge/ship impact detection in Section 7.3 has demonstrated the capabilities of the developed system. Similar methodologies can be applied for railway bridge/over-height vehicle detection. However, some potential issues must be addressed to achieve this objective. First of all, the measurement range of trigger accelerometer may not be adequate for impact loading, especially for steel bridge/vehicle impact. The saturation issue has already been reported in train-crossing event monitoring, in which the bridge vibration under rail wheels impact can exceed 50g or even hundreds of G in local areas. In this scenario, though trigger accelerometer in this study can send trigger signals to wake up sensor nodes, the obtained FIFO buffer acceleration data may not be useful, unless saturation issue is fully addressed. On possible solution is to employ state-of-the-art high-g shock accelerometers as trigger accelerometer, but its resolution is very low for accurate damage assessment. Therefore, hybrid solution of integrating low-g and high-g trigger accelerometer may be a possible solution.

(2) Overweight vehicle detection

Overweight vehicles are another type of critical sudden events of concern for highway transportation systems. Greater demands on the road transport infrastructure as a result of economic growth have manifested themselves in an increase in the number of Heavy Duty Vehicles (HDV) nationwide. They can contribute to bridge fatigue damage, or even sudden collapse of bridges. Moreover, the overweight trucks can result in more traffic accidents, disorder in transportation systems, and eventually more cost to bridge users. For example, AZDOT estimates that overweight vehicles impose somewhere between $12M and $53M per year in uncompensated damages to Arizona roadways (Straus & Semmens, 2006). Efficient technology, primarily weigh-in-motion (WIM), is significantly on demand for weight enforcement.

Conventional WIM system is very expensive, ranging from $55K to $120K. The developed system is an attractive cost-effective solution, which can not only detect over-weight vehicles but also provide bridge response data for rapid condition assessment. However, high false positive/negative rate may be a critical concern, as the loading conditions on highway bridges are complicated. In addition, an overweight vehicle may not give rise to high dynamic bridge response, if it is driving slowly. As a result, trigger accelerometer may not be applicable. To this end, trigger strain circuit may be a possible solution, considering that overweight vehicles should result in large strain measurement.

(3) Event-driven structural control

The promising technologies developed in this study encourage many applications beyond the scope of SHM. A typical example is event-driven structural control. Control devices installed on structures are typically used for long time, but the control action may only be taken for rare events (e.g., earthquakes). Maintaining always-on monitoring for structural control under rare events can be significant prohibitive for some control applications (e.g., active control). The developed system is a well-suited tool to enable event-driven structural control.

REFERENCES

Abdel-Ghaffar, A.M., and Scanlan, R.H. (1985). Ambient vibration studies of golden gate bridge: I. Suspended structure. *Journal of Engineering Mechanics*, 111(4), 463-482.

Adler, R., Flanigan, M., Huang, J., Kling, R., Kushalnagar, N., Nachman, L., Wan, C.Y., and Yarvis, M. (2005). Intel mote 2: an advanced platform for demanding sensor network applications. In *Proceedings of the 3rd international conference on Embedded networked sensor systems* (pp. 298-298). ACM.

Agrawal, A.K., Xu, X., and Chen, Z. (2011). *Bridge-vehicle impact assessment* (No. C-07-10). University Transportation Research Center.

Analog Devices. (2017). *ADXL372 Datasheet* at: http://www.analog.com/media/en/technical-documentation/data-sheets/adxl372.pdf

Asadollahi, P., and Li, J. (2017). Statistical analysis of modal properties of a cable-stayed bridge through long-term wireless structural health monitoring. *Journal of Bridge Engineering*, 22(9), 04017051.

ASCE Steering Committee. (2007). *The ACSE vision for civil engineering in 2025*. Reston, VA: American Society of Civil Engineers, 15.

Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., and Inman, D.J. (2018). Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks. *Journal of Sound and Vibration*, 424, 158-172.

Bao, Y., Li, H., Sun, X., Yu, Y., and Ou, J. (2013). Compressive sampling–based data loss recovery for wireless sensor networks used in civil structural health monitoring. *Structural Health Monitoring*, 12(1), 78-95.

BeanAir. (2016). *Smart Sensor User Manual* at: http://beanair.com/wa_files/Datasheet-wifi-accelerometer-beandevice-Wilow-AX-3D.pdf.

Bernal, D. (2002). Load vectors for damage localization. *Journal of Engineering Mechanics.* 128(1):7-14.

Bischoff, R., Meyer, J., Enochsson, O., Feltrin, G., and Elfgren, L. (2009). Event-based strain monitoring on a railway bridge with a wireless sensor network. In *Proceedings of the 4th International Conference on Structural Health Monitoring of Intelligent Infrastructure, Zurich, Switzerland* (Vol. 2224, p. 7482).

Bocca, M., and Eriksson, L.M. (2011). A synchronized wireless sensor network for experimental modal analysis in structural health monitoring. *Computer-aided Civil and Infrastructure Engineering*. 2011; 26:483-499.

Bourdenas, T., and Sloman, M. (2010). Starfish: policy driven self-management in wireless sensor networks. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems* (pp. 75-83). ACM.

Caicedo, J.M., Clayton, E., Dyke, S.J., Abe, M. and Tokyo, J., (2002). Structural health monitoring for large structures using ambient vibrations. *Proc. of the ICANCEER Conference*, Hong Kong, August, 15-20.

Celebi, M. (2006). Real-time seismic monitoring of the new cape girardeau bridge and preliminary analyses of recorded data: An overview. *Earthquake Spectra*, 22, 609.

Çelebi, M. (2013). Seismic monitoring of structures and new developments. *Earthquakes and health monitoring of civil structures*, 37-84. Springer, Dordrecht.

Celebi, M., Sanli, A., Sinclair, M., Gallant, S., and Radulescu, D. (2004). Real-time seismic monitoring needs of a building owner—and the solution: a cooperative effort. *Earthquake Spectra*, *20*(2), 333-346.

Chang, C.M., Chou, J.Y., Tan, P., and Wang, L. (2017). A sensor fault detection strategy for structural health monitoring systems. *Smart Structures and Systems*, *20*(1), 43-52.

Chen, B., Chen, Z.W., Wang, G.J., and Xie, W.P. (2014). Damage detection on sudden stiffness reduction based on discrete wavelet transform. *The Scientific World Journal*.

Cheng, L., and Pakzad, S.N. (2009). Agility of wireless sensor networks for earthquake monitoring of bridges. In *Networked Sensing Systems (INSS), 2009 Sixth International Conference*, IEEE. June.

Crossbow, (2003), *MICA2 Data Sheet*, Crossbow, Milpitas, CA.

Crossbow, (2004), *MICAz MPR2400CA Data Sheet*, Crossbow, Milpitas, CA.

Ding, M., Chen, D., Xing, K., and Cheng, X. (2005). Localized fault-tolerant event boundary detection in sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.* (Vol. 2, pp. 902-913). IEEE.

Dong, X.J., Chen, S., Zhu, D.P., Kane, M., Wang, Y., and Lynch, J.P. (2014). High-Speed Heterogeneous Data Acquisition using Martlet-A Next-Generation Wireless Sensing Node. *Structure* 8: 9.

Dorvash, S., Li, X., Pakzad, S., and Cheng, L. (2012). Network architecture design of an agile sensing system with sandwich wireless sensor nodes. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2012* (Vol. 8345, p. 83450H). International Society for Optics and Photonics.

Dragos, K., and Smarsly, K. (2016). Distributed adaptive diagnosis of sensor faults using structural response data. *Smart Materials and Structures*, *25*(10), 105019.

Dragos, K., Theiler, M., Magalhães, F., Moutinho, C., and Smarsly, K. On-board data synchronization in wireless structural health monitoring systems based on phase locking. *Structure Control and Health Monitoring.*, 2018; 25(11), p.e2248.

Elson, J.E. (2003). *Time synchronization in wireless sensor networks*. Ph.D. Thesis, University of California, Los Angeles.

Fu, Y., Hoang, T., Mechitov, K., and Spencer, B. (2019). Rapid Condition Assessment of Bridges under Impacts: from System Design to Decision Making. In: *9th*

International Conference on Structural Health Monitoring of Intelligent Infrastructure., *St. Louis, IN, August 2019*.

Fu, Y., Hoang, T., Mechitov, K., Kim, J., Zhang, D., and Spencer, B. (2018). Sudden-Event Monitoring of Civil Infrastructure Using Demand-Based Wireless Smart Sensors. *Sensors*, 18(12), 4480.

Fu, Y., Mechitov, K.A., Hoskere, V., and Spencer, Jr., B.F. (2016). Development of RTOS-based wireless SHM system: benefits in applications. *International Conference on Smart Infrastructure and Construction*. Cambridge, UK, June 27-29.

Fu, Y., Mechitov, K., Hoang, T., Kim, J.R., Lee, D.H., and Spencer Jr, B.F. (2019). Development and full-scale validation of high-fidelity data acquisition on a next-generation wireless smart sensor platform. *Advances in Structural Engineering*, 1369433219866093.

Fu, Y., Peng, C., Gomez, F., Narazaki, Y., and Spencer Jr. B.F. (2019). Sensor Fault Management Techniques for Wireless Smart Sensor Networks in Structural Health Monitoring. *Structural Control and Health Monitoring*, e2362.

Fu, Y., Zhu, L., Hoang, T., Mechitov, K., and Spencer Jr, B.F. (2018). Demand-based wireless smart sensors for earthquake monitoring of civil infrastructure. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*. International Society for Optics and Photonics.

Gao, Y., and Spencer Jr, B. F. (2005). Online damage diagnosis for civil infrastructure employing a flexibility-based approach. *Smart Materials and Structures*, *15*(1), 9.

Gao, Y., and Spencer Jr, B.F. (2008). *Structural health monitoring strategies for smart sensor networks*. Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign.

Ganeriwal, S., Kumar, R., and Srivastava, M.B. (2003). Time-sync protocol for sensor networks. *Proc. the First ACM Conference on Embedded Networked Sensor System (SenSys)*. 138-149.

Garner, A.C., and Huff, W.A. (1997). The wreck of Amtrak's Sunset Limited: News coverage of a mass transport disaster. *Disasters*, 21, 4-19.

Ghajari, M., Sharif-Khodaei, Z., Aliabadi, M.H., and Apicella, A. (2013). Identification of impact force for smart composite stiffened panels. *Smart Materials and Structures*, 22(8), 085014.

Gomez, F., Park, J.W., and Spencer Jr, B.F. (2018). Reference-free structural dynamic displacement estimation method. *Structural Control and Health Monitoring*, 25(8), e2209.

Guo, S., Kim, S.M., Zhu, T., Gu, Y., and He, T. (2011). Correlated flooding in low-duty-cycle wireless sensor networks. *In Proceedings of the 2011 19th IEEE International Conference on Network Protocols*, Vancouver, BC, Canada, October.

Guo, S., Kim, S.M., Zhu, T., Gu, Y., and He, T. (2011). Correlated flooding in low-duty-cycle wireless sensor networks. In *Proceedings of the 2011 19th IEEE International Conference on Network Protocols*, Vancouver, BC, Canada, 383–392.

Hackmann, G., Sun, F., Castaneda, N., Lu, C., and Dyke, S. (2012). A holistic approach to decentralized structural damage localization using wireless sensor networks. *Computer Communications*, *36*(1), 29-41.

Hedley, M., Hoschke, N., Johnson, M., Lewis, C., Murdoch, A., Price, D., Prokopenko, M., Scott, A., Wang, P., and Farmer, A. (2004). Sensor network for structural health monitoring. In *Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference,* 361-366. IEEE.

Hill, J.L., and Culler, D.E., (2002). MICA: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro*, Vol. 22, No. 6, 12–24.

Hoskere, V., Park, J.W., Yoon, H., and Spencer Jr, B.F. (2019). Vision-Based Modal Survey of Civil Infrastructure Using Unmanned Aerial Vehicles. *Journal of Structural Engineering*, 145(7), 04019062.

Huang, G., and Chen, X. (2009). Wavelets-based estimation of multivariate evolutionary spectra and its application to nonstationary downburst winds. *Engineering Structures*, 31(4), 976-989.

Hung, S.L., Ding, J.T., and Lu, Y.C. (2019). Developing an energy-efficient and low-delay wake-up wireless sensor network-based structural health monitoring system using on-site earthquake early warning system and wake-on radio. *Journal of Civil Structural Health Monitoring*, 9(1), 103-115.

Hsu, T.Y., Huang, S.K., Lu, K.C., Loh, C.H., Wang, Y., and Lynch, J.P. (2011). On-line structural damage localization and quantification using wireless sensors. *Smart Materials and Structures*, *20*(10), 105025.

InvenSense. (2013). *MPU-6050 Datasheet* at: https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/.

Jalsan, K.E., Soman, R.N., Flouri, K., Kyriakides, M.A., Feltrin, G., and Onoufriou, T. (2014). Layout optimization of wireless sensor networks for structural health monitoring. *Smart Structures and Systems.* 14(1), 39-54.

Jang S.A. (2010). *Structural health monitoring for bridge structures using wireless smart sensors*. Ph.D dissertation. University of Illinois at Urbana-Champaign.

Jang, S., Jo, H., Cho, S., Mechitov, K.A., Rice, J.A., Sim, S.H., Jung, H.-J., Yun, C.-B., Spencer, B.F., Jr., and Agha, G. (2010). Structural health monitoring of a cable-stayed bridge using smart sensor technology: Deployment and evaluation. *Smart Structures and Systems*. 6, 439-459.

Jang, S., Spencer Jr, B.F., and Sim, S.H. (2012). A decentralized receptance-based damage detection strategy for wireless smart sensors. *Smart Materials and Structures*, *21*(5), 055017.

Jo, H., Sim, S.H., Nagayama, T., and Spencer Jr., B.F. (2011). Development and application of high-sensitivity wireless smart sensors for decentralized stochastic modal identification. *Journal of Engineering Mechanics* 138(6): 683-694.

Joy, R., Jones, M.C., Otter, D., and Maal, L. (2013). *Characterization of Railroad Bridge Service Interruptions*. Transport Research International Documentation - TRID.

Kane, M., Zhu, D., Hirose, M., Dong, X., Winter, B., Häckell, M., Lynch, J.P., Wang, Y., and Swartz, A., (2014). Development of an extensible dual-core wireless sensing node for cyber-physical systems. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems* (Vol. 9061, p. 90611U). International Society for Optics and Photonics.

Kim, J., Swartz, A., Lynch, J.P., Lee, J.J., and Lee, C.G. (2010). Rapid-to-deploy reconfigurable wireless structural monitoring systems using extended-range wireless sensors. *Smart Structures and Systems*. 6(5-6), 505-524.

Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., and Turon, M. (2007). Health monitoring of civil infrastructures using wireless sensor networks. *In Proceedings of the 6th international conference on Information processing in sensor networks*, ACM. April.

Kim, R.E., Li, J., Spencer, B.F., Nagayama, T., and Mechitov K.A. (2016). Synchronized sensing for wireless monitoring of large structures. *Smart Structures and Systems.* 18(5), 885-909.

Kirianaki, N.V., Yurish, S.Y., Shpak, N.O., and Deynega, V.P. (2002). *Data Acquisition and Signal Processing for Smart Sensors*. 1st edn, Wiley & Sons, Ltd., 1–8.

Kling, R.M. (2003). Intel Mote: An Enhanced Sensor Network Node. In *Proceedings of the International Workshop on Advanced Sensors, Structural Health Monitoring and Smart Structures*, Tokyo, Japan, November 10–11.

Kling, R., Adler, R., Huang, J., Hummel, V., and Nachman, L. (2005). Intel Mote-based sensor networks. *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures*, 12(3-4), 469-479.

Kotsiantis, S., Kanellopoulos, D., and Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1), 25-36.

Krishnamurthy, V., Fowler, K., and Sazonov, E. (2008). The effect of time synchronization of wireless sensors on the modal analysis of structures. *Smart Materials and Structures*, 17:1-13.

Kurata, M., Kim, J., Zhang, Y., Lynch, J.P., Van Der Linden, G.W., Jacob, V., Thometz, E., Hipley, P. and Sheng, L.H. (2011). Long-term assessment of an autonomous wireless structural health monitoring system at the new Carquinez Suspension Bridge. *In Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2011*. International Society for Optics and Photonics.

Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., and Culler, D. (2005). TinyOS: an operating system for sensor networks. *Ambient Intelligence*, Weber, W., Rabaey, J.M., Aarts, E., Eds. 115-148, Springer, Berlin, Heidelberg.

Li, J., Mechitov, K.A., Kim, R.E., and Spencer, Jr. B.F. (2016). Efficient time synchronization for structural health monitoring using wireless smart sensor networks. *Structural Control and Health Monitoring*. 23(3), 470-486.

Li, J., Nagayama, T., Mechitov, K.A., and Spencer, B.F. (2012). Efficient campaign-type structural health monitoring using wireless smart sensors. *In Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems* Apr 3 (Vol. 8345, p. 83450U). International Society for Optics and Photonics.

Libelium. (2016). *Waspmote Datasheet* at: http://www.libelium.com/downloads/documentation/waspmote_datasheet.pdf.

Linderman, L.E., Jo, H., and Spencer, B.F. (2014). Low-latency data acquisition hardware for real-time wireless sensor applications. *IEEE Sensors Journal*, *15*(3), 1800-1809.

Linderman, L.E., Mechitov, K.A., and Spencer Jr, B.F. (2013). TinyOS-based real-time wireless data acquisition framework for structural health monitoring and control. *Structural Control and Health Monitoring*, *20*(6), 1007-1020.

Liu, Y., Voigt, T., Wirström, N., and Höglund, J. (2019). ECOVIBE: On-Demand Sensing for Railway Bridge Structural Health Monitoring. *IEEE Internet of Things Journal*, *6*(1), 1068-1078.

Lo, C., Bai, Y., Liu, M., and Lynch, J.P. (2015). Efficient Sensor Fault Detection Using Group Testing. *arXiv preprint arXiv:1501.04152*.

Lo, C., Lynch, J.P., and Liu, M. (2016). Distributed model-based nonlinear sensor fault diagnosis in wireless sensor networks. *Mechanical Systems and Signal Processing*, *66:* 470-484.

Lo, C., Lynch, J. P., and Liu, M. (2011). Reference-free detection of spike faults in wireless sensor networks. In *2011 4th International Symposium on Resilient Control Systems* (pp. 148-153). IEEE.

Lu, G., De, D., Xu, M., Song, W.Z., and Cao, J. (2010). TelosW: Enabling ultra-low power wake-on sensor network. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)* (pp. 211-218). IEEE.

Lu, K.C., Loh, C.H., Yang, Y.S., Lynch, J.P., and Law, K.H. (2008). Real-time structural damage detection using wireless sensing and monitoring system. *Smart Structures and Systems*, 4(6), 759-778.

Lu, X., Wang, P., Niyato, D., and Han, Z. (2015a). Resource allocation in wireless networks with RF energy harvesting and transfer. *IEEE Network.* 29(6), 68-75.

Lu, X.., Wang, P., Niyato, D., Kim, D.I., and Han, Z. (2015b). Wireless networks with RF energy harvesting: A contemporary survey. *IEEE Communications Surveys & Tutorials*, 17(2), 757-789.

Lu, X., Zhang, Y., Ye, L. and He, S.T. (2009). Failure Modes and Load Calculation of Collision Between Over-high Truck and Bridge Superstructure. *China Journal of Highway and Transport*, 22(5), pp.60-67.

Lu, W. (2010). *The structural health monitoring system of large span space structure based on fieldbus*. Ph. D dissertation, Harbin institute of Technology, 2010.

Luo, X., Dong, M., and Huang, Y. (2005). On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on computers*, *55*(1), 58-70.

Lynch, J.P., Law, K.H., Kiremidjian, A.S., Kenny, T.W., Carryer, E., and Partridge, A. (2001). The design of a wireless sensing unit for structural health monitoring. in *Proceedings of the 3rd International Workshop on Structural Health Monitoring*, Stanford, CA.

Lynch, J.P., and Loh, K.J. (2006). A summary review of wireless sensors and sensor networks for structural health monitoring. *Shock and Vibration Digest*, *38*(2), 91-130.

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.

Lynch, J.P., Wang, Y., Loh, K.J., Yi, J.H., and Yun, C.B. (2006). Performance monitoring of the Geumdang Bridge using a dense network of high-resolution wireless sensors. *Smart Materials and Structures*, *15*(6), 1561.

Mahapatro, A., and Khilar, P.M. (2013). Fault diagnosis in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, *15*(4), 2000-2026.

Maroti, M., Kusy, B., Simon, G., and Ledeczi, A. (2004). The flooding time synchronization protocol. *Proc. 2nd International Conference on Embedded Networked Sensor Systems.* Baltimore, MD, 39–49.

Microstrain (2017), *G-Link-200 Wireless Accelerometer Node* at: http://www.microstrain.com/sites/default/files/g-link-200_datasheet_8400-0102_2_0.pdf.

Miller, T.I., Spencer, B.F., Jr., Li, J., and Jo, H. (2010). *Solar Energy Harvesting and Software Enhancements for Autonomous Wireless Smart Sensor Networks*; NSEL Report No. 022; University of Illinois at Urbana-Champaign: Urbana, IL, USA.

Moreu, F., Kim, R. E., and Spencer Jr, B.F. (2017). Railroad bridge monitoring using wireless smart sensors. *Structural Control and Health Monitoring*, 24(2), e1863.

Moustapha, A.I., and Selmic, R.R. (2008). Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection. *IEEE Transactions on Instrumentation and Measurement*, *57*(5), 981-988.

Nagayama, T., and Spencer Jr, B.F. (2007). *Structural health monitoring using smart sensors. Newmark Structural Engineering Laboratory*. University of Illinois at Urbana-Champaign.

Nagayama, T., Sim, S.H., Miyamori, Y., and Spencer Jr, B.F. (2007). Issues in structural health monitoring employing smart sensors. *Smart Structures and Systems*, *3*(3), 299-320.

Nagayama, T., Moinzadeh, P., Mechitov, K., Ushita, M., Makihata, N., Leiri, M., Agha, G., Spencer Jr, B.F., Fujino, Y., and Seo, J.W. (2010). Reliable multi-hop communication for structural health monitoring. *Smart Structures and Systems*, *6*(5-6), 481-504.

Network Rail. (2017). The risk of bridge strikes. at: https://www.networkrail.co.uk/running-therailway/looking-after-the-railway/bridges-tunnels-viaducts/risk-bridge-strikes/

Ni, K., Ramanathan, N., Chehade, M.N.H., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., Hansen, M., and Srivastava, M. (2009). Sensor network data fault types." *ACM Transactions on Sensor Networks (TOSN)* 5, 3: 25.

Ni, Y.Q., Ye, X.W., and Ko, J.M., (2010). Monitoring-based fatigue reliability assessment of steel bridges: analytical model and application. *Journal of Structural Engineering*, 136(12), 1563-1573.

Niu, J., Deng, Z., Zhou, F., Cao, Z., Lui, Z., and Zhu, F. (2009). A structural health monitoring system using wireless sensor network. *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference*, Beijing, China, 2009.

Ou, J., and Li, H. (2010). Structural health monitoring in mainland China: review and future trends. *Structural Health Monitoring*, 9(3), 219-231.

Okada, K., Nakamura, Y., and Saruta, M. (2009). Application of earthquake early warning system to seismic-isolated buildings. *Journal of Disaster Research*, *4*(4), 242-250.

Pakzad, S.N., Fenves, G.L., Kim, S., and Culler, D.E. (2008). Design and implementation of scalable wireless sensor network for structural monitoring. *Journal of Infrastructure Systems*, 14(1), 89-101.

Paradis, L., and Han, Q. (2007). A survey of fault management in wireless sensor networks. *Journal of Network and systems management*, 15(2), 171-190.

Park, J.W., Jung, H.J., Jo, H., and Spencer, B.F., Jr. (2012). Feasibility study of micro wind turbines for powering wireless sensors in a cable-stayed bridge. *Energies.* 5, 3450–3464.

Peng, C., Fu, Y.G., and Spencer, B.F., Jr. (2017). Sensor fault detection, identification, and recovery techniques for wireless sensor networks: a full-scale study. In *Proc.13th International Workshop on Advanced Smart Materials and Smart Structures Technology,* Tokyo, Japan.

Polastre, J., Szewczyk, R., and Culler, D. (2005). Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th international symposium on Information processing in sensor networks* (p. 48). IEEE Press.

Popovic, N., Feltrin, G., Jalsan, K. E., and Wojtera, M. (2017). Event-driven strain cycle monitoring of railway bridges using a wireless sensor network with sentinel nodes. *Structural Control and Health Monitoring*, 24(7), e1934.

Potenza, F., Federici, F., Lepidi, M., Gattulli, V., Graziosi, F., and Colarieti, A. (2015). Long-term structural monitoring of the damaged Basilica S. Maria di Collemaggio through a low-cost wireless sensor network. *Journal of Civil Structural Health Monitoring*.5: 655–676.

Qu, W., Teng, J., Xiang, H., Zhong, L., Liu, H., Wang, J., and Li, G. (2006). Intelligent health monitoring for roof space truss structure of the Shenzhen Citizen Center under wind load. *Jianzhu Jiegou Xuebao(Journal of Building Structures)*, *27*(1), 1-8.

Rault, T., Bouabdallah, A., and Challal, Y. (2014). Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*, 67, 104-122.

Rice, J.A., Mechitov, K., Sim, S.H., Nagayama, T., Jang, S., Kim, R., Spencer Jr, B.F., Agha, G., and Fujino, Y. (2010). Flexible smart sensor framework for autonomous structural health monitoring. *Smart Structures and Systems*, 6(5-6), 423-438.

Rice, J.A., and Spencer Jr, B.F. (2009). *Flexible smart sensor framework for autonomous full-scale structural health monitoring*. Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign.

Sazonov, E., Krishnamurthy, V., and Schilling, R. (2010). Wireless intelligent sensor and actuator network -- a scalable platform for time synchronous applications of structural health monitoring. *Structure Health Monitoring*. 9(5):465–476.

Scheerer, M., Lager, D., and Göral, F. (2013). Development and Testing of a Hybrid Active–Passive Acoustic SHM System for Impact Damage Detection in Honeycomb Aircraft Structures. *Proc. of the 19th ICCM, Montreal, Canada*, 7154-7165.

Sharif-Khodaei, Z., Ghajari, M., and Aliabadi, M.H. (2012). Determination of impact location on composite stiffened panels. *Smart Materials and Structures*, 21(10), 105026.

Sharma, A.B., Golubchik, L., and Govindan, R. (2010). Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks (TOSN)*, *6*(3), 23.

Singhal, A. (2015). LaRa-OHVD: An innovative laser ranging over-height vehicle detection system to prevent bridge strikes. *In: Intelligent Transportation Society of New York 22nd Annual Meeting*.

Straus, S.H., and Semmens J. (2006). *Estimating the Cost of Overweight Vehicle Travel on Arizona Highways*. Final report, Arizona Department of Transportation.

Sutton, F., Da Forno, R., Gschwend, D., Lim, R., Gsell, T., Beutel, J., and Thiele, L. (2016). A heterogeneous system architecture for event-triggered wireless sensing. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks* (p. 33). IEEE Press.

Sim, S.H. (2011). *Decentralized identification and multimetric monitoring of civil infrastructure using smart sensors*. Ph.D dissertation, University of Illinois at Urbana-Champaign.

Sim, S.H., and Spencer Jr, B.F. (2009). *Decentralized strategies for monitoring structures using wireless smart sensor networks*. Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign.

Sim, S.H., Li, J., Jo, H., Park, J.W., Cho, S., Spencer Jr, B.F., and Jung, H.J. (2013). A wireless smart sensor network for automated monitoring of cable tension. *Smart Materials and Structures*, 23(2), 025006.

Smarsly, K., and Law, K.H. (2014). Decentralized fault detection and isolation in wireless structural health monitoring systems using analytical redundancy. *Advances in Engineering Software*, 73, 1-10.

Sohn, H., Farrar, C., 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* (No. LA-13761-MS). Los Alamos National Lab., NM (US).

Sohn, H., Farrar, C.R., Hemez, F.M., Shunk, D.D., Stinemates, D.W., Nadler, B.R., and Czarnecki, J.J. (2003). A review of structural health monitoring literature: 1996–2001. *Los Alamos National Laboratory*.

Spencer, B.F., Park J.W., Mechitov K.A., Jo, H., and Agha, G. (2016). Next Generation Wireless Smart Sensors toward Sustainable Civil Infrastructure. *Proc. of Sustainable Civil Engineering Structures and Construction Materials*, *SCESCM* 2016.

Spencer, B.F., Ruiz-Sandoval, M.E., and Kurata, N. (2004). Smart sensing technology: opportunities and challenges. *Structural Control and Health Monitoring*, *11*(4), 349-368.

STMicroelectronics. (2017). *LIS3DSH Datasheet* at: https://www.st.com/resource/en/datasheet/lis3dsh.pdf

Straser, E.G., Kiremidjian, A.S., Meng, T.H., and Redlefsen, L. (1998). A modular, wireless network platform for monitoring structures. in *Proceedings-SPIE The International Society for Optical Engineering,* 1, 450–456, SEM, Santa Barbara, CA.

Sun, Z., Krishnan, S., Hackmann, G., Yan, G., Dyke, S.J., Lu, C., and Irfanoglu, A. (2015). Damage detection on a full-scale highway sign structure with a distributed wireless sensor network. *Smart Structures and Systems*, 16, 223-242.

Swartz, R.A., Jung, D., Lynch, J.P., Wang, Y., Shi, D., and Flynn, M.P. (2005). Design of a wireless sensor for scalable distributed in-network computation in a structural health monitoring system. in *Proceedings of the 5th International Workshop on Structural Health Monitoring*, Stanford, CA.

Wade, L. (2019). Is your building safe after an earthquake? These cheap sensors could tell you. *Science*, at: https://www.sciencemag.org/news/2019/02/your-building-safe-after-earthquake-these-cheap-sensors-could-tell-you.

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.

Whelan, M.J., and Janoyan, K.D. (2009). Design of a robust, high-rate wireless sensor network for static and dynamic structural monitoring. *Journal of Intelligent Material Systems and Structures*, 20(7), 849-863.

Wong, K.Y. (2004). Instrumentation and health monitoring of cable-supported bridges. *Structural Control and Health Monitoring*, 11(2), 91-124.

Wong, K.Y. (2007). Design of a structural health monitoring system for longspan bridges. *Structure and Infrastructure Engineering*, 3,169–185.

Xu, J., Spencer Jr, B.F., Lu, X., Chen, X., and Lu, L. (2017). Optimization of structures subject to stochastic dynamic loading. *Computer-Aided Civil and Infrastructure Engineering*, 32(8), 657-673.

Yamazaki, F. (2001). Seismic monitoring and early damage assessment systems in Japan. *Progress in Structural Engineering and Materials*, *3*(1): 66-75.

Yamazaki, F., Katayama, T. and Yoshikawa, Y. (1994). On-line damage assessment of city gas networks based on dense earthquake monitoring. *Proc. of 5th U.S. National Conference on Earthquake Engineering* 4:829-837.

Yamazaki, F., Motomura, H., and Hamada, T. (2000). Damage assessment of expressway networks in Japan based on seismic monitoring. *In Proceedings of the 12th world conference on earthquake engineering*, January.

Yang Y., and Nagarajaiah, S. (2014). Blind denoising of structural vibration responses with outliers via principal component pursuit. *Structure Control and Health Monitoring*. 21(6):962-978.

Yang, Y., and Nagarajaiah, S. (2016). Harnessing data structure for recovery of randomly missing structural vibration responses time history: Sparse representation versus low-rank structure. *Mechanical Systems and Signal Processing*, 74, 165-182.

Yang, Y., and Nagarajaiah, S. (2014). Blind identification of damage in time-varying systems using independent component analysis with wavelet transform. *Mechanical Systems and Signal Processing*, 47(1-2), 3-20.

Yao, R., and Pakzad, S.N. (2012). Autoregressive statistical pattern recognition algorithms for damage detection in civil structures. *Mechanical Systems and Signal Processing*, 31, 355-368.

Yu, C.B., Hu, J.J., Li, R., Deng, S.H., and Yang, R.M. (2014). Node fault diagnosis in WSN based on RS and SVM. In *2014 International Conference on Wireless Communication and Sensor Network* (pp. 153-156). IEEE.

Zhao, C., Sun, X., Sun, S, and Jiang, T. (2011). Fault diagnosis of sensor by chaos particle swarm optimization algorithm and support vector machine. *Expert Systems with Applications* 38, 8: 9908-9912.

Zhao, F., and Guibas, L. (2004). *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufman, San Francisco, CA.

List of Recent NSEL Reports

No.	*Authors*	*Title*	*Date*
038	Sim, S-H. and Spencer, B.F.	Decentralized Identification and Multimetric Monitoring of Civil Infrastructure using Smart Sensors	June 2015
039	Giles, R.K. and Spencer, B.F.	Development of a Long-term, Multimetric Structural Health Monitoring System for a Historic Steel Truss Swing Bridge	June 2015
040	Spencer, B.F., Moreu, F., and Kim, R.E.	Campaign Monitoring of Railroad Bridges in High-Speed Rail Shared Corridors using Wireless Smart Sensors	June 2015
041	Moreu, F. and Spencer, B.F.	Framework for Consequence-based Management and Safety of Railroad Bridge Infrastructure Using Wireless Smart Sensors (WSS)	June 2015
042	Spencer, B.F. and Gardoni, P. (Eds.)	Innovations and Advances in Structural Engineering: Honoring the Career of Yozo Fujino	August 2015
043	Asai, T. and Spencer, B.F.	Structural Control Strategies for Earthquake Response Reduction of Buildings	August 2015
044	Kim, R. and Spencer, B.F.	Modeling and Monitoring of the Dynamic Response of Railroad Bridges using Wireless Smart Sensors	September 2015
045	Wierschem, N. and Spencer, B.F.	Targeted Energy Transfer using Nonlinear Energy Sinks for the Attenuation of Transient Loads on Building Structures	September 2015
046	Fahnestock, L.A. and Hashash, Y.M.A.	Structural and Geotechnical Observations after the April 25, 2015 M7.8 Gorkha, Nepal Earthquake and its Aftershocks	December 2016
047	Yoon, H. and Spencer, B.F.	Enabling Smart City Resilience: Post-disaster Response and Structural Health Monitoring	December 2016
048	Kozak, D.L., Luo, J., Olson, S.M., LaFave, J.M., and Fahnestock, L.A.	Modification of Ground Motions for use in Central North America: Southern Illinois Surface Ground Motions for Structural Analysis	September 2017
049	Fermandois, G.A. and Spencer, B.F.	Development and Implementation of a Multi-axial Real-time Hybrid Simulation Framework	September 2018
050	Eick, B.A. and Spencer, B.F.	Structural health monitoring of inland navigation infrastructure	August 2021
051	Najafi, A. and Spencer, B.F.	Multi-axial Real-time Hybrid Simulation Framework for Testing Nonlinear Structural Systems with Multiple Boundary Interfaces	August 2021
052	Eick, B.A., Levine, L.M., Smith, M.A., and Spencer, B.F.	Full-scale Laboratory Testing of Embedded Miter Gate Anchorages	October 2021
053	Fu, Y. and Spencer, B.F.	Sudden-event Monitoring of Civil Infrastructure using Wireless Smart Sensors	June 2022