A Lightweight Wireless Structural Health Monitoring System

Yinhao Wang[1,2], Zhiyi Tang[1,2*], Wei Xu[1,2], and Jinlong Wen[1,2]

[1] Faculty of Civil Engineering and Mechanics, Kunming University of Science and Technology, Kunming, 650500, China
[2] Intelligent Infrastructure Operation and Maintenance Technology Innovation Team of Yunnan Provincial, Department of Education, Kunming University of Science and Technology, Kunming 650500, China
tang@kust.edu.cn

Abstract. This paper proposes a green, lightweight, and cost-effective wireless system for structural health monitoring (SHM) of bridges. The system utilizes an edge computing model, eliminating the need for traditional cloud services by enabling sensor nodes to collaborate with a local intelligent hub for data collection, storage, and presentation. The system employs wireless sensor nodes based on MEMS accelerometers powered by batteries with support for solar energy supplementation. These nodes are equipped with low-power, low-cost wireless microcontrollers to ensure continuous and stable data collection. This paper details the hardware and software design of the system, with a focus on the issue of sensor node clock synchronization. Experimental validation shows that the system performs excellently in terms of data collection effectiveness and sensor node clock synchronization.

Keywords: Wireless Sensor Network, MEMS Accelerometer, Structural Health Monitoring, Clock Synchronization, High-Precision Data Collection.

1 Introduction

As one of the relatively expensive infrastructures, bridges inevitably develop structural cracks, performance degradation, and other types of damage during long-term operation due to creep, corrosion, and cyclic loads. Additionally, unexpected events such as vehicle overloading, collisions, floods, and earthquakes pose serious threats to bridge structures' normal operation and safety reserves. As vital nodes in the transportation network, bridges play a crucial role in regional traffic. Therefore, it is necessary to monitor the structural condition of bridges in real time and extend their service life through proper management and maintenance. Traditional bridge health diagnosis methods primarily rely on manual on-site inspections, which are time-consuming and labor-intensive and exhibit a delay in responding to sudden events [1].

W. Guo et al. (eds.), *Proceedings of the 3rd International Conference on Green Building, Civil Engineering and Smart City (GBCESC 2024)*, Advances in Engineering Research 264,
https://doi.org/10.2991/978-94-6463-728-1_73

Moreover, any inspection oversight or maintenance lapse can lead to significant future costs [2].

To overcome these drawbacks, bridge structural health monitoring (BSHM) technology has emerged. This technology uses advanced sensors and data analysis tools to monitor various parameters of bridges in real time, such as stress, vibration, temperature, and displacement. BSHM continuously collects structural health data through sensors placed at critical parts of the bridge. This data is transmitted to cloud servers for storage, analysis, and evaluation. However, during data collection by the sensors, transmitting data to the cloud server consumes a large amount of bandwidth and incurs higher server usage costs.

Common structural sensors used in BSHM include strain gauges [3], displacement sensors [4], and accelerometers [5]. Among them, accelerometers play a key role in capturing dynamic responses of structures. They are mainly classified into piezoresistive [6], piezoelectric [7], and micro-electromechanical systems (MEMS) [8]. With the development of MEMS accelerometers, their applications have significantly increased due to their low power consumption, high sensitivity, cost-effectiveness, and adequate sampling frequency [9]. These characteristics have made them play an important role in bridge health monitoring.

Structural monitoring systems can be divided into wired and wireless types. The former uses physical cables for data transmission [10], while wireless sensors use radio frequency or Bluetooth [11]. Although wireless sensing systems offer advantages such as easy installation, flexible positioning, remote monitoring, and low cost, they also face challenges such as battery power limitations, low data transmission reliability, and lack of time synchronization among multiple sensors. To address the data synchronization issues in wireless monitoring systems [12], various post-processing techniques, including timestamp estimation and correlation functions, have been developed. These aim to establish a consistent time reference across the network, ensuring accurate synchronization and facilitating reliable data collection [9]. The application of these technologies has significantly advanced the use of wireless monitoring systems in structural health monitoring (SHM).

Among network time synchronization protocols, the most widely used is the Network Time Protocol (NTP). NTP achieves precise time synchronization through communication between sensors and servers. However, several key conditions must be met to ensure accurate time synchronization with NTP. First, network latency must be kept within a small range to minimize clock synchronization errors. Second, network latency needs to be symmetric, meaning the round-trip time should be as similar as possible in both directions. If network latency fluctuates significantly or is asymmetric, the accuracy of NTP time synchronization will be greatly affected. Therefore, in practical applications, the stability and symmetry of the network environment are crucial for the performance of the NTP protocol. However, NTP servers used in monitoring systems are typically located in the cloud, and data packets undergo multiple routing hops, where fluctuations in any segment of the network can affect the accuracy of time synchronization.

This study addresses the challenges of wireless monitoring systems in Structural Health Monitoring (SHM) of bridges, including system complexity, cost-effectiveness, data reliability, and time synchronization issues, by proposing a lightweight and cost-effective wireless data acquisition system. The system relies on edge devices, eschewing traditional cloud services, and combines sensor nodes with a local intelligent hub to achieve data collection, processing, storage, and display. By integrating a GNSS module into the intelligent hub and establishing an NTP server, the physical distance between sensor nodes and the NTP server is reduced, thereby decreasing the time calibration error between sensor nodes.

2 Monitoring System

Fig. 1 shows the architecture of the system. Sensor nodes transmit collected accelerometer data wirelessly to the intelligent hub via a 4G router. The intelligent hub collects accelerometer data from all nodes, battery voltage information, and performs real-time storage and display. Additionally, the intelligent hub collects temperature, humidity, and GPS time location information. Users can access a webpage via the Internet to view the operational status of the sensor system.

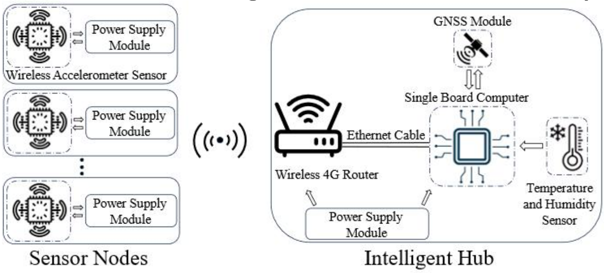

Fig. 1. Overview of the monitoring system

2.1 Sensor Nodes

Hardware. The sensors are designed using commercial components to meet application requirements. The PCB circuit board integrates an accelerometer, wireless microcontroller, serial communication module, and voltage divider circuits required for battery voltage collection. All components are connected to the wireless microcontroller for control and management, and the device is powered by an independent finished PCB board. Fig. 2 shows the internal diagram and PCB layout of the sensor node.

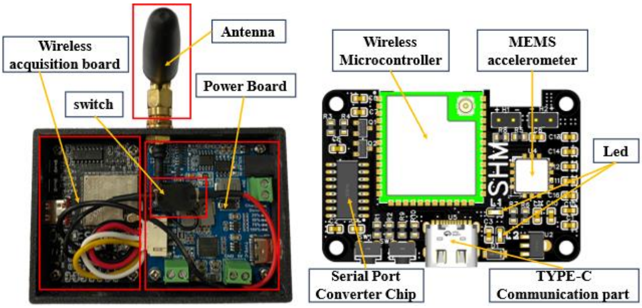

Fig. 2. Schematic of the sensor node internals

Accelerometer. In structural health monitoring, commonly used accelerometer sensors include piezoelectric and MEMS accelerometers. Voltage-type IEPE accelerometers exhibit high sensitivity, wide frequency response range, high accuracy, reliability, and durability. However, they are bulky, high-power consuming, and require complex power supplies. In contrast, MEMS accelerometers offer advantages such as miniaturization, lightweight, low power consumption, low cost, and simple interface, although they have limited frequency response and lower sensitivity.

After considering performance, power consumption, size, and price, we selected the ADXL355 three-axis MEMS accelerometer from Analog Devices Inc. The device consists of three independent MEMS sensor elements, employing differential signal paths to reduce noise and drift. It also reduces noise by applying filters before and after the ADC. The sensor's range, output data rate, and filter cutoff frequencies can be customized through register access.

Considering that acceleration values in structural health monitoring are typically less than 1g, this study sets the accelerometer range to ±2g. The sensor's frequency range is adjusted from 250Hz to 2000Hz. The ADXL355 offers diverse data synchronization modes, including no external sync, using external sync signals and internal clock sync, and enabling interpolation filters. To ensure stable sampling rates, we use external sync signals and internal clock synchronization, along with enabling interpolation filters. We utilize the ESP32-S3's built-in high-resolution timer as the external sync clock, with a clock frequency deviation of less than ±10ppm and a time resolution of 1μs, superior to the 1668ppm internal clock error of MEMS sensors [13]. The specific parameters of the ADXL355 are shown in Table **1**.

Table 1 Key parameters of the ADXL355 version [15].

Attribute	Testing Conditions/Remarks	Average Value	Measurement Unit
Sensitivity	±2g (X, Y, and Z)	256,000	LSB/g
Noise (spectral density)	±2g (X, Y, and Z)	22.5	$\mu g/\sqrt{Hz}$
Sensitivity change due to temperature	−40℃ to +125℃	±0.01	%/℃
Nonlinearity	±2g	0.1	%
Zero offset	±2g (X, Y, and Z)	25	mg

Wireless Microcontroller. To meet the study's functional and power requirements, the ESP32-S3 was chosen as the wireless microcontroller for the sensor nodes. This high-performance, low-power microcontroller integrates Wi-Fi and Bluetooth 5 (LE), enabling efficient data transmission. With dual 32-bit LX7 CPU cores clocked at up to 240 MHz, the ESP32-S3 handles complex tasks and supports high sensor sampling rates. It features multiple interfaces (SPI, I2C, UART, ADC, DAC, PWM) for sensor and peripheral connections, with SPI used for MEMS accelerometers. The device includes 512 KB SRAM and 8 MB PSRAM for data buffering and supports low-power modes, making it ideal for battery.

Battery and Power Supply. Two 3400mAh 18650 lithium batteries were selected to power the sensor nodes, offering high energy density, low self-discharge, long lifespan, and fast charging. The devices can be charged via mains electricity or a solar panel, with the batteries providing over 30 hours of operation without solar input. The design includes an 18W Type-C charging interface and supports up to 20W solar charging, ensuring stable, long-term operation in various environments.

Miscellaneous. The sensor housing is 3D printed using ABS material, compact in size with external dimensions of 75mm x 50mm x 45mm. There are 6 holes with strong magnets placed at the bottom of the housing, allowing the sensor node to be conveniently and securely attached to metal surfaces, reducing installation complexity. The top cover and bottom part are also magnetically connected, allowing users to inspect the internal hardware status of the sensor node through internal indicator lights, as shown in Fig. 3.

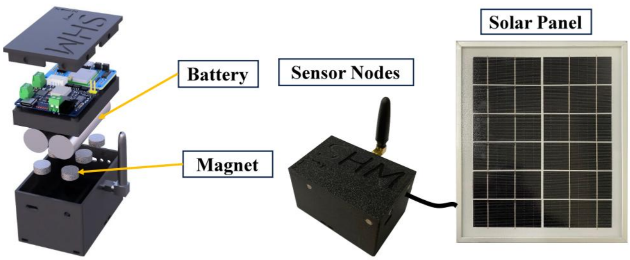

Fig. 3. Internal architecture and physical image of the sensor node

Firmware. The firmware for the sensor node is developed using the Espressif IoT Development Framework (ESP-IDF), based on the FreeRTOS (Real-Time Operating System) kernel. Unlike traditional microcontroller bare-metal development using FIFO task management, FreeRTOS supports multitasking development and offers real-time capabilities. Utilizing the high-performance dual-core processor of the ESP32-S3, it achieves multitasking parallel processing, enabling high-speed acquisition and transmission of accelerometer data. The firmware is divided into three main tasks: a timed NTP time calibration service, a sensor data acquisition task, and an MQTT data transmission task. To ensure the continuity and reliability of data acquisition, a large-capacity data buffer queue is established in the ESP32-S3's PSRAM, ensuring that data collection can proceed normally even if data is not sent in a timely manner.

For data transmission, the lightweight MQTT protocol is used, with the Quality of Service (QoS) level set to 1. This ensures that each piece of data can be reliably received by the recipient. The MQTT protocol is particularly suitable for low bandwidth and unstable network environments. QoS level 1 means that the message will be delivered at least once, with the sender continuing to resend the message until an acknowledgment is received, ensuring reliable data transmission. The entire operation process is shown in Fig. 4.

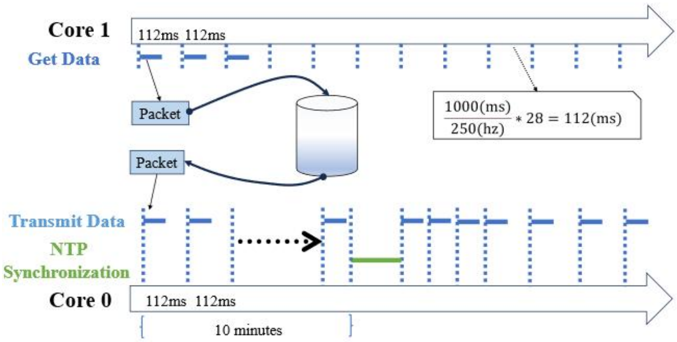

Fig. 4. Multitasking operation flow of the sensor node

For data acquisition, a hardware PWM controller generates a square wave with the same sampling rate, providing an external synchronization signal to the ADXL355. The INT pin is pulled high whenever the ADXL355 is ready with new data. The ESP32-S3 is configured for external hardware interrupts to capture the data-ready signal. When the data volume meets the preset packet size, the acquisition task is initiated to read the FIFO data from the ADXL355. The collected data is formatted into a uint8 type array, as shown in the packet structure in Fig. 5. The first 8 bytes of each data packet are the millisecond-level timestamp, followed by 1 byte for the device ID, 4 bytes for the sensor node's battery voltage information, and the acceleration data for each axis in 20-bit binary format. A data packet contains multiple sets of x, y, and z-axis acceleration data. The ADXL355 has a 96-entry 21-bit FIFO, capable of storing 32 sets of x, y, and z-axis data. In actual tests at a sampling rate of 250Hz, if each data packet contains only 1 set of acceleration data, the sending frequency becomes too high, and the MCU cannot send all the data in time. Conversely, if the data packet contains more than 28 sets of data, the FIFO will overflow due to the inability to read data in time. Table 2 shows the ESP32-S3's transmission power consumption and bandwidth usage for different packet sizes. Considering the sensor node's power consumption and bandwidth usage, each data packet is ultimately configured to contain 28 sets of acceleration data.

Table 2 Sensor node power consumption and bandwidth utilization at 250 Hz sampling rate for various packet sizes.

Packet Size	2	10	20	28
5-Minute Average Power Consumption (mW)	673.3176	593.1691	584.2711	583.1520
Average RX/TX Rate (kbit/s)	57.73/87.32	12.49/32.34	6.39/24.90	4.74/22.72

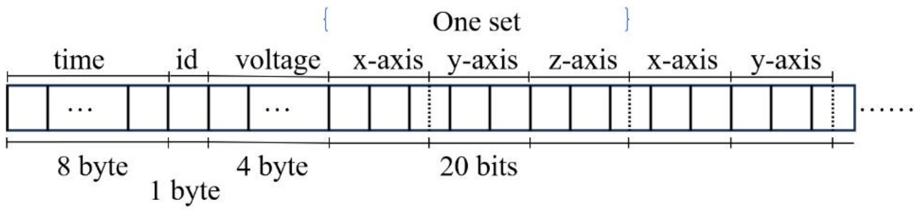

Fig. 5. Schematic diagram of the packet structure

In the time synchronization of multiple sensor nodes, the internal clocks of the sensor nodes rely on crystal oscillators, but the oscillation frequency shifts with temperature changes. Typically, the crystal oscillator frequency increases with rising temperature and decreases with falling temperature. The extent of this thermal drift depends on the internal structure of the crystal oscillator and external environmental factors. Therefore, the clocks of the sensor nodes become unsynchronized after a period of time. To address this issue, a Network Time Protocol (NTP) calibration algorithm is used, synchronizing with a locally established NTP server every ten minutes.

2.2 Intelligent hub

Unlike traditional monitoring systems that use cloud solutions for data storage and display, this study employs the single-board computer Orange Pi5 as the core unit for data collection, processing, storage, and display of the monitoring system. All services run under the Linux framework, and developers can easily modify the code remotely via SSH through a 4G router.

Hardware. In terms of hardware, the intelligent hub mainly consists of the following components:

- 4G Router: This system uses the Pomegranate X4C, which needs to perform two functions within the entire system:
  a) Data Collection: Sensor nodes send data to the 4G router via Wi-Fi, and the router then transmits the data to the intelligent hub through a wired connection.
  b) Internet Access: Through the 4G router, the entire system can access the internet wherever there is a 4G signal. This allows firmware upgrades and

sets up FRP intranet penetration, mapping necessary ports to the internet so that homeowners can connect to the system anytime and anywhere to view real-time accelerometer data and temperature and humidity information.

- Single Board Computer (SBC): The Orange Pi 5, a cost-effective, high-performance SBC used in this study, features an ARM-based 64-bit 8-core processor, 16GB RAM, and supports 4K dual screens. Its extensive peripheral interfaces include Gigabit Ethernet, M.2 SSD, and 26-pin GPIO, making it ideal for development. Running Ubuntu, it simplifies application development and provides ample storage with a 500GB M.2 SSD. With only 20% CPU utilization during operation, it offers substantial computing power for additional services.
- GNSS Module: This component is responsible for acquiring location information at the bridge site and providing a high-precision clock source for the entire system. Considering price and power consumption, the ATGM336H was chosen. This module supports BDS, GPS, and GLONASS satellite systems, with ultra-low power consumption (less than 25mA during continuous operation). The clock error of the GNSS module is less than 200ns [14]. The component connects to the SBC via serial port and connects the PPS (Pulse Per Second) pin to a GPIO pin to ensure high-precision time synchronization, thereby enhancing the accuracy and reliability of data collection.
- Temperature and Humidity Module: This component uses the SHT31 sensor, chosen for its cost-effectiveness and accuracy, to measure temperature and humidity at the bridge site. It offers a 0-100%RH humidity range with ±2%RH accuracy and a -40~125°C temperature range with ±0.3°C accuracy. The module connects to the SBC via I2C, delivering precise environmental data for the system.
- Battery and Power Supply: To ensure the stable and continuous operation of the entire system, the router and smart central unit will use a 74Wh lithium battery as the main energy storage unit, supplemented by a 20W solar panel for additional energy. In actual operation, the average power consumption of the 4G router and single-board computer is 2.80W and 4.66W, respectively.

Software framework. In this system, the backend framework of the smart central unit is shown in Fig. 6. Below are introductions to key services:

- MQTT Message Service: Built using Docker, EMQX is responsible for receiving data from all sensor nodes. It decomposes the raw binary data into timestamps, device IDs, device voltage, and 28 sets of acceleration data, then stores the decomposed data in tables named after the device ID and date. For example, "s1_2024_06_11" where "s1" denotes sensor ID 1, followed by the date.
- Database Service: TDengine database was chosen for its open-source nature, high performance, cloud-native characteristics, and strong scalability. Each row represents a set of data including an int64 millisecond-level timestamp, int32 sensor node voltage information, and BINARY acceleration data.

- Temperature and Humidity Acquisition Service: Using C language to write I2C protocol code for reading temperature and humidity sensors, data collected every 1 second is timestamped and saved to the database.
- NTP Service: Using the gpsd software on the SBC to receive NMEA sentences from the GNSS module via serial port, parsing them to obtain UTC time, then using the PPS second pulse to calibrate the delay in obtaining time from the serial port. Finally, based on the high-precision GNSS clock source, establishing a local NTP time server using Chrony.
- Data Display Service: Utilizing the Grafana open-source data visualization tool, real-time monitoring data is displayed by loading data from the TDengine database.

These backend services work together to support the system's data collection, processing, storage, and display functions, providing efficient and reliable technical support for bridge structural health monitoring.

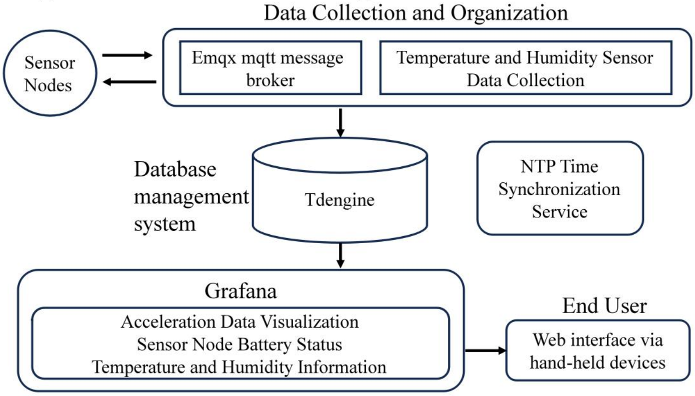

Fig. 6. Software framework of the intelligent hub

3 Experimental Verification

3.1 Vibration Measurement

To verify the performance of the sensor in bridge health monitoring, this study conducted a 30-minute measurement of the fundamental frequency under natural excitation on a double-span model steel bridge. The study also used finite element software to build a bridge model and calculate its dynamic modes. To provide a reference signal, an Integrated Electronics Piezo-Electric (IEPE) accelerometer was installed next to the sensor, as shown in Fig. 7. The IEPE accelerometer (Donghua Test 1A110E) has a measurement range of ±100g, a sensitivity of $5mV/m \cdot s^{-2}$, and a maximum sampling rate of 7kHz.

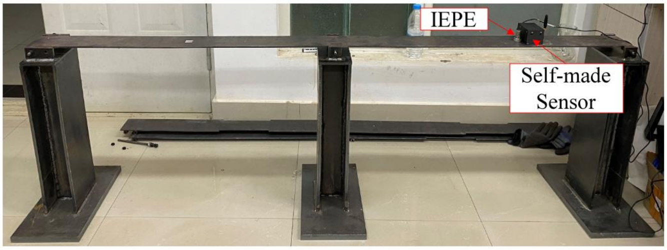

Fig. 7. Deployment diagram of IEPE sensor and self-made sensor

Fig. 8(a) shows the sensor's measurement results of the steel bridge under natural excitation. To convert the data from the time domain to the frequency domain and calculate the structure's natural frequencies, a Fast Fourier Transform (FFT) was applied, with the results shown in Fig. 8(b). Fig. 9 presents the first four modes calculated using finite element software.

As indicated in Table 3, the sensor successfully identified the first four frequencies, with the frequency peaks clearly visible in the FFT data. Although there is a 5% discrepancy between the frequencies calculated by the finite element software and the actual measured frequencies attributable to the software's omission of the sensor's weight during modeling the comparison with the IEPE sensor results shows a high degree of consistency, with errors less than 0.01%. These findings suggest that the sensor performs excellently in monitoring bridge vibrations. However, the sensor's weight may influence modal analysis in smaller structures, whereas in larger actual bridge structures, the effect of the sensor's weight is negligible.

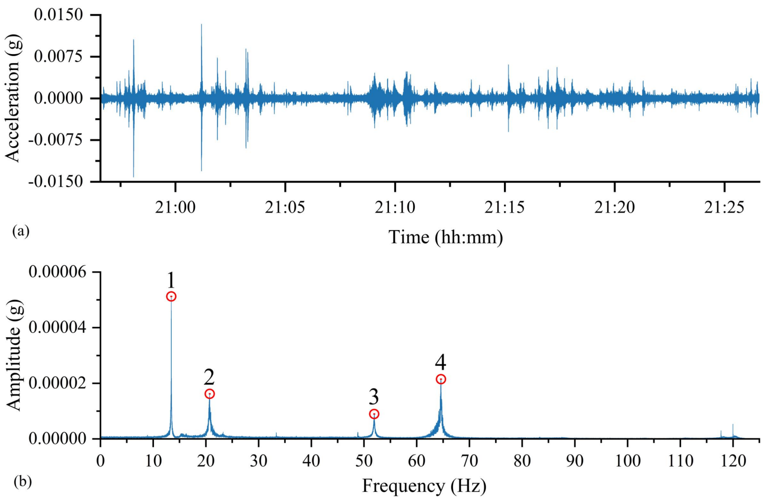

Fig. 8. The developed sensor measurement results under natural excitation: (a) time domain plot, (b) frequency domain plot

Table 3. Comparison of modal characteristics measured by different methods

Mode	1	2	3	4
Finite element calculation (Hz)	13.334	20.829	53.322	67.467
IEPE sensor measurement (Hz)	13.130	20.252	51.886	63.765
The developed sensor measurement (Hz)	13.129	20.252	51.885	63.764
Difference between the developed sensor and finite element (%)	-1.537	-2.770	-2.695	-5.489
Difference between the developed sensor and IEPE sensor (%)	-0.008	0.000	-0.002	-0.002

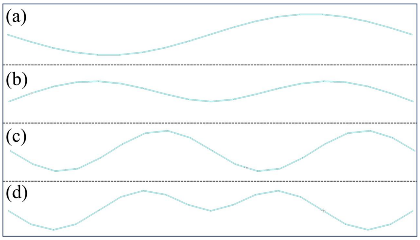

Fig. 9. Comparison of mode shapes: (a) first bending mode, (b) second bending mode, (c) third bending mode, (d) fourth bending mode,

3.2 NTP Synchronization Performance

Experimental Method: Two sensor nodes were configured to the same NTP server, with one placed on the bridge and the other under the bridge, as shown in Fig. 10, ensuring they receive the same vibration signal. A small hammer was used to strike the bridge surface, allowing both nodes to simultaneously receive the vibration signal. By collecting identical waveforms, the error between the sensor nodes was calculated to compare the local NTP server with public NTP servers. To ensure that the measured error in this experiment was solely due to NTP synchronization and not affected by the sensor's own crystal oscillator, the hammer test was conducted immediately after NTP synchronization. In this experiment, the sensor nodes' sampling rate was set to 1000Hz, and the NTP server addresses used were as follows:

- Local GPS NTP server address: 192.168.1.10
- Public server 1 address: ntp.aliyun.com
- Public server 2 address: cn.ntp.org.cn
- Public server 3 address: ntp.tencent.com

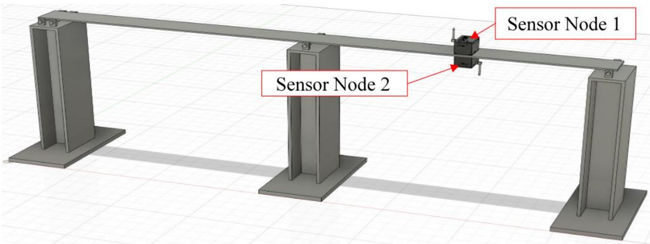

Fig. 10. Deployment diagram of two sensor nodes

Fig. 11 shows the synchronization performance when using a local NTP server and Alibaba Cloud's NTP server. When using the local NTP server, the two sensors almost simultaneously collected vibration data after the strike, indicating minimal error during time calibration. The acceleration curves of the two sensors were nearly identical with no phase difference. However, when using public cloud servers, the synchronization effect was poorer, with a time difference of up to 6ms between the two sensors, and their acceleration curves had a noticeable phase difference. Table 4 shows the results of four tests using different NTP servers. The results indicate that the synchronization effect varies depending on the NTP server used, and each synchronization attempt yields different results.

Table 4. Clock deviation between two sensor nodes using different NTP servers

Condition	Ping (ms)	Test1 (ms)	Test2 (ms)	Test3 (ms)	Test4 (ms)
Local GPS NTP server	<1	0	0	1	0
Public server 1	59	3	1	7	4
Public server 2	29	1	3	11	4
Public server 3	29	1	4	2	1

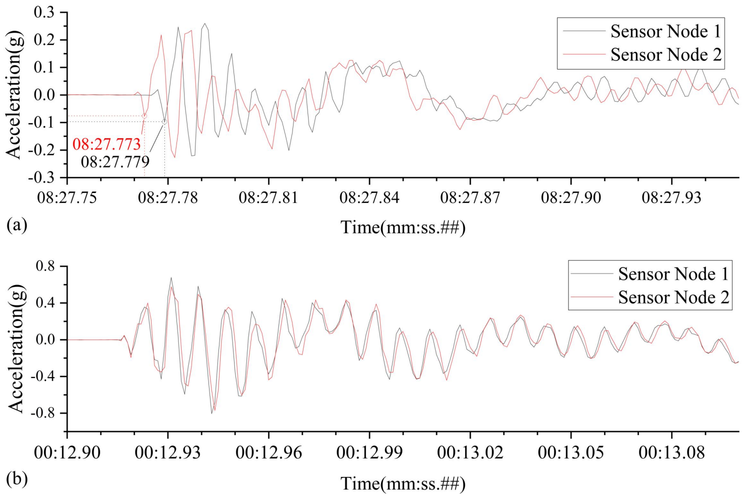

Fig. 11. Clock synchronization effect between two sensor nodes: (a) using a local NTP server; (b) using the public server: ntp.aliyun.com

When using local NTP services, network latency depends solely on the physical distance between the sensor nodes and the smart central unit. This method significantly reduces the transmission distance of timestamp packets over the network, ensuring consistent and high-precision sensor clock calibration.

4 Conclusion

This paper presents a comprehensive introduction and validation of a bridge wireless structural health monitoring system based on edge devices, utilizing an Orange Pi for local data collection, storage, and display. This approach effectively avoids the high costs and data transmission overhead associated with cloud servers. We resolved the issue of inconsistent time synchronization among wireless sensor nodes by establishing a local NTP server using a GPS clock. The effectiveness of the system's sensors was validated through comparisons with measurements from a commercial IEPE sensor and simulations from a finite element model of the steel bridge. Additionally, impact experiments demonstrated that sensor node clock consistency was significantly higher when using the local NTP server compared to other NTP services.

Through code optimization, the system utilizes ARM-based single-board computers to significantly reduce the power consumption of sensor nodes and the smart central unit. This novel, lightweight bridge structural health monitoring system achieves efficient and cost-effective monitoring, with good scalability for broader structural health monitoring applications. The design and implementation of the system provide an efficient, reliable, and economical solution for future bridge monitoring.

Acknowledgement

This research was supported by the Yunnan Fundamental Research Projects (Grant No. 202301AT070394, 202301BE070001-037).

References

1. He Shuanghai, Zhao Xiangmo, Ma Jian, et al. A review of highway bridge inspection and evaluation techniques. China Journal of Highway and Transport, 30(11): 63-80(2017).
2. Rashidi M, Mohammadi M, Sadeghlou Kivi S, et al. A decade of modern bridge monitoring using terrestrial laser scanning: Review and future directions. Remote Sensing, 12(22): 3796. (2020).
3. Teixeira de Freitas S, Kolstein H, Bijlaard F. Structural monitoring of a strengthened orthotropic steel bridge deck using strain data. Structural Health Monitoring, 11(5): 558-576(2012).
4. Vicente M A, Gonzalez D C, Minguez J, et al. A novel laser and video-based displacement transducer to monitor bridge deflections. Sensors, 18(4): 970(2018).
5. Ragam P, Devidas Sahebraoji N. Application of MEMS‐based accelerometer wireless sensor systems for monitoring of blast‐induced ground vibration and structural health: a review IET Wireless Sensor Systems, 9(3): 103-109(2019).
6. Ghemari Z. Study and analysis of the piezoresistive accelerometer stability and improvement of their performances. International Journal of System Assurance Engineering and Management, 8: 1520-1526(2017).

7. Chen Y, Xue X. Advances in the structural health monitoring of bridges using piezoelectric transducers. Sensors, 18(12): 4312(2018).
8. Sabato A. Pedestrian bridge vibration monitoring using a wireless MEMS accelerometer board//2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE: 437-442(2015).
9. Hasani H, Freddi F. Operational Modal Analysis on Bridges: A Comprehensive Review. Infrastructures, 8(12): 172(2023).
10. Bayraktar A, Altunişik A C, Türker T. Structural condition assessment of Birecik highway bridge using operational modal analysis. International Journal of Civil Engineering, 14: 35-46(2016).
11. Zhang G, Moutinho C, Magalhães F. Continuous dynamic monitoring of a large - span arch bridge with wireless nodes based on MEMS accelerometers. Structural Control and Health Monitoring, 29(7): e2963(2022).
12. Römer K, Blum P, Meier L. Time synchronization and calibration in wireless sensor networks. Handbook of sensor networks: Algorithms and architectures: 199-237(2005).
13. Narayanan S. Synchronization of Wireless Accelerometer Sensors for Industrial Application. Delft University of Technology, (2019).
14. Vanier J, Godone A, Levi F. Proceedings of the 1999 Joint Meeting of the European Frequency and Time Forum and the IEEE International Frequency Control Symposium (Cat. No. 99CH36313). (1999).
15. Devices A. Low Noise, Low Drift, Low Power, 3-Axis MEMS Accelerometer ADXL355. Analog Devices, Available: https://www.analog.com/media/en/technical-documentation/data-sheets/adxl354_355.pdf, [Accessed: 02-Dec-2019], (2016).