Hindawi
Structural Control and Health Monitoring
Volume 2023, Article ID 5240059, 16 pages
https://doi.org/10.1155/2023/5240059

Research Article

A Low-Cost Wireless Multinode Vibration Monitoring System for Civil Structures

Renan Rocha Ribeiro,[1] **Rafael de Almeida Sobral**,[1] **Ian Barreto Cavalcante**,[1] **Luís Augusto Conte Mendes Veloso**,[2] **and Rodrigo de Melo Lameiras**[1]

[1]*Department of Civil and Environmental Engineering, Faculty of Technology, University of Brasília, Brasília 70.910-900, Brazil*
[2]*Department of Mechanical Engineering, Faculty of Technology, University of Brasília, Brasília 70.910-900, Brazil*

Correspondence should be addressed to Renan Rocha Ribeiro; renan.rocha.ribeiro@gmail.com

Received 21 February 2023; Revised 10 July 2023; Accepted 15 July 2023; Published 1 August 2023

Academic Editor: Chia-Ming Chang

Structural health monitoring (SHM) has gained importance because many structures are approaching the end of their design life and demanding maintenance and monitoring. Low-cost solutions may push forward a widespread implementation of SHM on infrastructures but further investigation is still required to assess the performance of technically accessible, simple, and scalable low-cost systems. This work presents the development and validation of a low-cost vibration-based SHM multinode wireless system, based on the Arduino platform, for identification of modal parameters in civil infrastructures. Full details about the hardware and source code of the system are disclosed in an open repository, allowing its reproduction even by non-specialists in electronics. The sampling frequency stability of the system is experimentally characterized, and interpolation postprocessing algorithms are proposed to solve inherent limitations. The system is validated, and its performance is investigated in impulse and ambient vibration tests performed in a real-scale slab and a high-grade system. The data obtained from the proposed system in impulse tests allowed estimation of natural frequencies within 2%, and MAC values around 0.3 to 0.9, in relation to those estimated with the high-grade system. However, the low-cost system was unable to produce usable data in ambient vibration tests.

1. Introduction

Structural health monitoring (SHM) has gained importance due to the fact that many structures, such as bridges, buildings, dams, and roads, are approaching the end of their design life, leading to a growing need for condition monitoring and early damage identification [1]. For reliable SHM, data should come from automatic systems fed by electronic sensors, since traditional strategies of visual and manual inspections suffer drawbacks of cost, imprecision, and accessibility in large infrastructures [1, 2].

In this context, vibration-based SHM techniques are considered a mature approach [3, 4] for globally (instead of locally) investigating the condition of structures [2]. They are based on the identification of modal parameters of the structure, known to have multiple correlations with damage detection indices [3, 5]. Data are collected from a network of multiple sensor nodes (usually accelerometers [6]), to provide enough spatial resolution in damage evaluation [7]. Recently, wireless technologies have been the go-to option to enable such networks due to lower implementation and management costs in comparison to wired systems [2, 8].

Use of low-cost solutions for vibration-based SHM wireless systems is one of the efforts for pushing forward a widespread implementation of SHM on infrastructures. These efforts focus on developing electronic systems from scratch using low-cost components, which tends to involve intricate electronic aspects [9–11], or using standard low-cost components and accessible prototyping platforms [12, 13], such as Arduino, leading to a simpler approach that allows non-specialists in electronics to develop their own systems. Some representative works of this last case include developments of Wi-Fi sensor networks based on Arduino and ESP modules [14, 15]; XBee wireless sensor networks

with measurement nodes controlled by Arduino boards [16–18]; and radio-frequency networks based on Arduino boards [19, 20]. Low-cost SHM wireless systems may also be associated with IoT (Internet of Things) applications to enable data streams across platforms and support data-driven processes [21, 22]. Association of low-cost SHM wireless system to IoT has led to research addressing data fusion of different sensors, in order to compensate for the lower performance of individual low-cost sensors [12, 13], details about the system architecture (*e.g.*, system topology, synchronization strategy, and communication protocol) [21, 23–25], and power management solutions to ensure an autonomous system [24].

Some of the inherent difficulties in the deployment of a low-cost vibration-based SHM sensor network are related to a lack of detailed literature on how to reproduce systems, overly sophisticated systems from the point of view of electronics, no information on the expectable performance of low-cost systems to detect modal parameters of real-scale civil infrastructures, or various technical difficulties, such as data synchronization between nodes [26, 27], the achievement of high and stable sampling frequency rates, network scalability, and data transmission inside the network [8, 28].

The objective of this work was to develop a low-cost vibration-based SHM wireless system comprised of a scalable sensing network controlled by a central node, based on the Arduino platform, for identification of modal parameters in civil infrastructures. The sampling frequency stability of the system was characterized, and a postprocessing interpolation strategy to circumvent observed limitations was proposed and investigated. A real-scale laboratorial slab was used as a test benchmark, being monitored both with the proposed low-cost and a high-grade system, so as to validate the low-cost system and assess its performance for estimating natural frequencies and mode shapes in comparison to a high-grade system. Considerable effort was dedicated to make the electronic aspects as accessible as possible to non-experts in electronics, impacting decisions from the choice of components to the writing of the work. Additionally, all electronic schematics, system source codes, and post-processing algorithms are made available on an open-source repository [29]. In this way, the authors expect to enable researchers and practitioners, who eventually are not familiar with electronics, to use such electronic prototyping platforms to solve real-world problems in SHM for civil structures.

2. Materials and Methods

2.1. Overview. Figure 1 presents an overview of the work, which was comprised of three main stages: development of the low-cost system, characterization of the sampling frequency stability of the system, and validation of the system in impulse and ambient vibration tests by comparison to high-grade equipment.

The development of the low-cost system started from a simple version of isolated and unsynchronized nodes (Version A), followed by a version with wired and synchronized nodes (Version B), and finally the intended wireless system (Version C). Only the final version (Version C) was posteriorly characterized and validated in the subsequent stages. The development process also included the implementation of a postprocessing interpolation strategy to deal with non-equispaced sampling in time domain, and four different postprocessing interpolation strategies were implemented. All the development details are presented in Section 2.2.

After being fully developed, Version C was subjected to stability characterization of its sampling frequency and validation through real-scale slab vibration tests to assess its performance in practical applications. The sampling frequency characterization consisted of analysing the distribution of time intervals between consecutive samples of the system. The validation test consisted of monitoring a real-scale laboratorial slab under impulse and ambient vibration with both the developed low-cost wireless system and a high-grade wired system. The data collected on both systems were processed with the ARTeMIS Modal 7.2 to obtain estimates of the natural frequencies and mode shapes.

2.2. Development of Low-Cost System for SHM

2.2.1. General Requirements, Concepts, and Technical Difficulties. The general requirements that guided the development of the wireless low-cost system were as follows: (i) technical accessibility, as the system should be easily reproducible even by non-experts in electronics; (ii) simplicity, as the system should be developed as a minimal working example of a wireless SHM system and should provide simple solutions, even though not ideal for all application cases, for communication complexities typical of SHM wireless systems; and (iii) scalability, as the system should not have intrinsic limitations that may hinder its use on large structures. In this sense, the developed system was not meant to be readily deployed in real-world SHM but was instead a proof of concept of a system that would address basics related to wireless SHM and that could be further developed for real SHM applications.

The Arduino platform [30] was chosen as the microcontroller prototyping platform upon which the system would be developed, since it is a platform targeted to non-experts and hobbyists in electronics, with an active support community and a considerable amount of documentation for learning and development. The Arduino prototyping platform can also easily be converted to stand-alone energy-efficient systems built on PCBs after the prototyping phase is finished. Specifically, the Arduino Nano board [31] was chosen for being the most compact entry-level Arduino board with low power consumption and all the necessary interfaces to connect to the electronic modules required for the intended wireless SHM system. The other components were added accordingly to the successively improved versions and were chosen among off-the-shelf components that would not require developing specialized features, such as interface libraries and custom-made PCB boards, in order to comply with the general requirements of technical accessibility and simplicity.

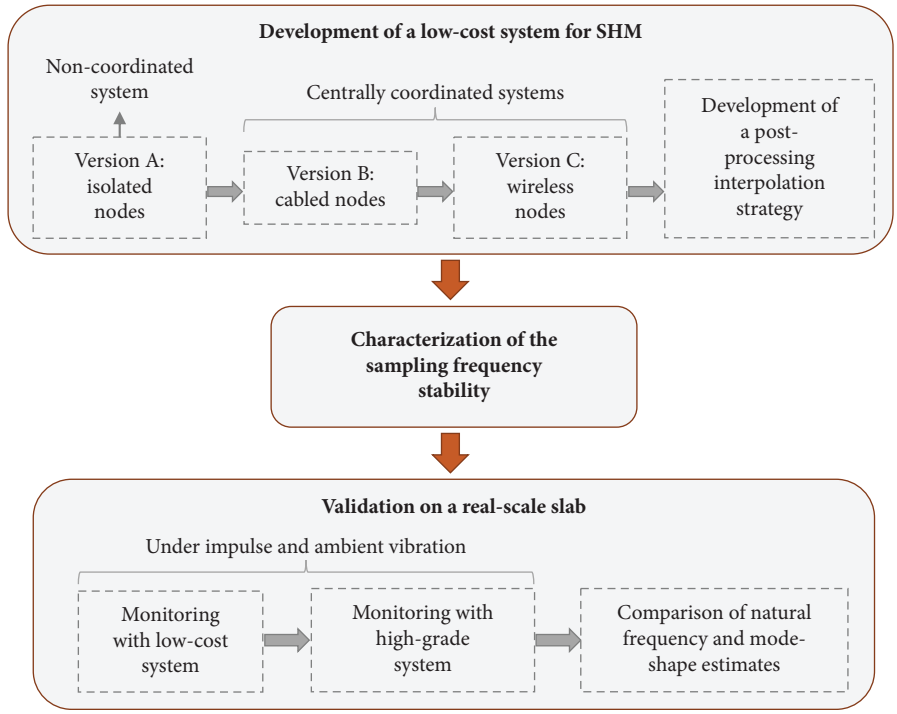

Figure 1: Overview of the work.

The concept of the sensor network was to allow a central node to coordinate all measurement nodes, in order to achieve synchronized measurements. A simple one-way communication protocol was devised in which a central node, identified as the Controller, would trigger a measurement session simultaneously on all measurement nodes, identified as the Measurers. Each Measurer node individually sampled data and stored it locally in a microSD card, chosen as the storage media due to ease of compatibility with the Arduino platform. After measurements were taken, it would be necessary to take the microSD cards from each measurement node and manually unload them on a computer, so as to allow postprocessing of the data for modal identification. This approach was deemed suitable to meet the three aforementioned requirements and create a SHM wireless system with all the minimal features.

The acceleration sensor chosen was the MPU6050, an IMU (inertial measurement unit) that has a triaxial MEMS accelerometer integrated to a 16 bit analog-to-digital converter (ADC) in a single chip, which can be interfaced via the I2C protocol. The MPU6050 was chosen due to its satisfactory performance for structural health monitoring (SHM) on civil structures and its easy-to-use interface for the Arduino platform, as reported in previous research [32]. The accelerometer has a range of ±2 g, a sampling frequency of up to 1000 Hz, a noise spectral density of 400 $\mu g/\sqrt{Hz}$, and an experimentally measured RMS noise of 3.8 mg [32, 33]. The MPU6050 sensor also has open-source libraries that can be used to readily interface with Arduino microcontrollers.

2.2.2. Version A: Independent Nodes. Figure 2 presents the electronic schematics and the network functionality of the first version of the proposed system, named Version A. Being the first version of the system, its main objective was to successfully integrate the components necessary to comprise a measurement node, whose main function was to sample acceleration and locally store them in a microSD card. In this sense, the network of Version A consisted of individual and independent measurement nodes that operated asynchronously in relation to each other, being manually activated by the press of an activation button on each node. Due to lack of any synchronization strategy, this network is not reliable for SHM applications, especially those relying on identification of mode shapes.

Data sampled from the accelerometer are stored locally using a generic microSD card module that communicates with the microcontroller via SPI interface [34], and a DS3231 real-time clock (RTC) is used to provide a time stamp at the beginning of each measurement session [35]. All of these components have open-source interface libraries available in Arduino online repositories for direct use in coding.

The power for each node was supplied individually by a commercial standard 2500 mAh power bank, commonly used to charge portable gadgets, connected to the USB port of the Arduino board. Due to constraints in the design of the power bank, which turned off if a minimum current drain was not drained, additional resistors were positioned between the Arduino digital pins set to 5 V and the ground line to ensure the required minimum current drain. The value and quantity of resistors were empirically defined until the system remained constantly on, also considering the maximum current drain allowed at each pin.

Figure 3 presents a pseudocode to represent the source code of Version A of the system. This code is executed by each measurement node once it is turned on. In the Arduino platform, the source code is mainly implemented around

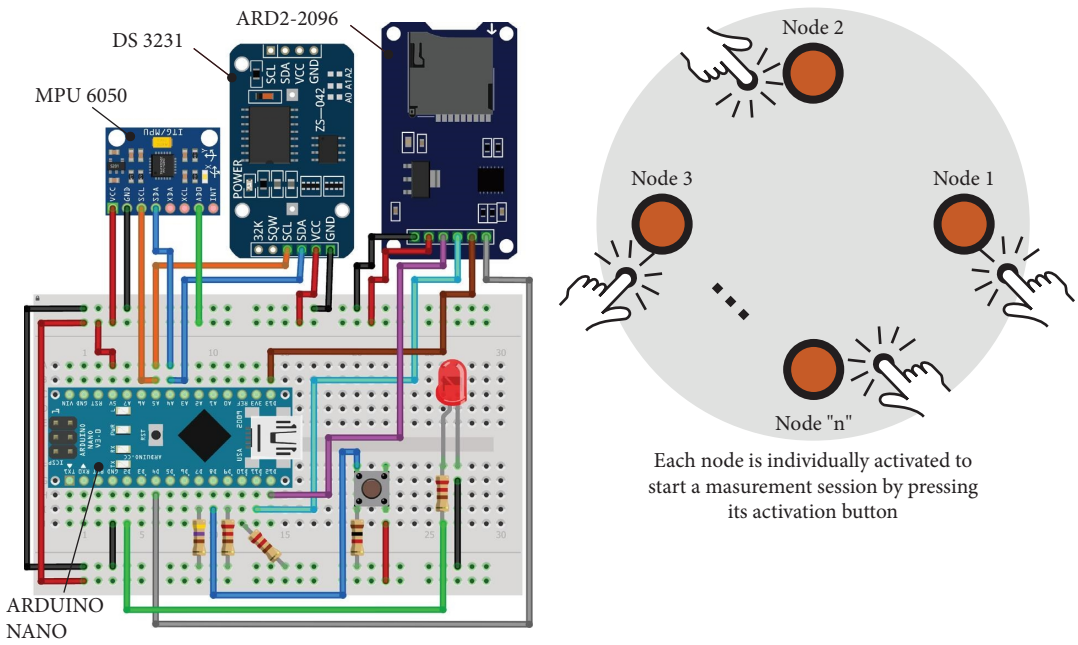

Figure 2: Version A node: (a) electronic schematic; (b) network functionality.

two main blocks of code: the setup block, whose instructions are only performed once, at start up; the loop block, whose instructions are repeated until the system is turned off. In the source code of Version A, after the initial setup block, a continuous loop checks whether the node's activation button was pressed by the user. If the pressing of a button is detected, the system performs a measurement cycle whose duration and sampling frequency are preconfigured in the source code (i.e., to change it, a new source code must be uploaded to the Arduino board). A red LED indicates to the user that the system is performing a measurement session. Once the measurement cycle is completed, the loop starts to check again for a button pressing.

2.2.3. Version B: Coordinated Wired Nodes. Figure 4 presents the electronic schematics and the network functionality of the second version of the low-cost system, named Version B. This version extends Version A by connecting all measurement nodes, named Measurers, to a central node, named Controller, whose role is to, once its activation button is pressed, simultaneously trigger all Measurers to start a measurement session, so as to provide synchronization at least at the start of measurement sessions. The electronic schematic shown in Figure 4(a) refers to a generic node which can be either a Controller or a Measurer, since the components required by both nodes are present. In a real implementation, some components may be removed, depending on the deployed node, for cost reduction (e.g., the push button may be removed in Measurers).

In Version B, the connection between the Controller and Measurers is done by cable using a generic commercial module built around the MAX485ESA chip, as illustrated in

Algorithm 1 Source code for Version A - isolated nodes

```
Initiate libraries and variables
start setup
    Turn on resistors to keep power bank on
    Initiate accelerometer, microSD card module, RTC module
end setup
start loop
    while button was not pressed do
        Check if button was pressed
    end while
    Open new measurement file in microSD
    Turn on LED to indicate start of sampling process
    Get time from RTC and save to file
    while elapsed time is less than sampling duration do
        Retrieve acceleration data from accelerometer
        Save to file
        Wait 1/(sampling frequency) seconds to get next sample
    end while
    Turn off LED to indicate end of sampling process
    Close measurement file
next loop
```

Figure 3: Pseudocode for Version A of the system.

Figure 4(a) [36], which allows bidirectional multipoint communication between systems through the RS-485 standard. The RS-485 standard is designed to be used over long distances, with the MAX485ESA chip rated to work in wire lengths up to 1200 m [36], and on high noise environments [37]. As illustrated in Figure 4(b), a single transmission line, composed by a pair of twisted cables, starts from the Controller node and is derived to each Measurer. In this work, only the Controller can send data over the line, with Measurers being able only to receive data. However, the source code may be further developed to allow more complex network capabilities, such as centralization of measurement data in the Controller node, without requiring hardware change.

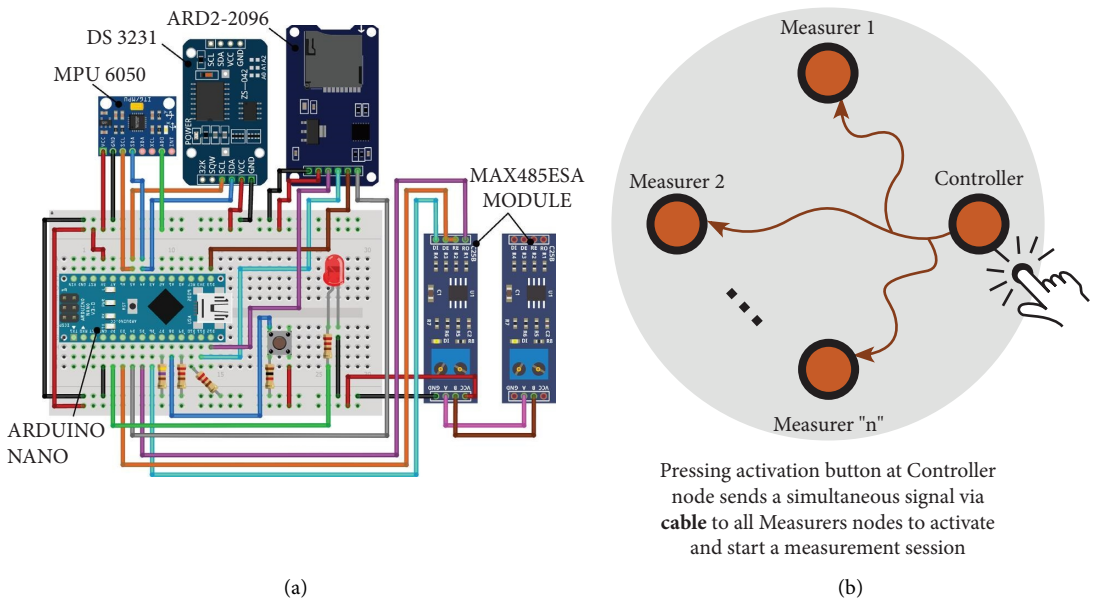

FIGURE 4: Version B node: (a) electronic schematic; (b) network functionality.

Figures 5 and 6 present the pseudocodes for the Measurers and Controller nodes, respectively, of the Version B system. The Controller node (Figure 6) continuously checks for the pressing of the button, which triggers the transmission of an ASCII encoded message containing the character "1" that signals simultaneously to all Measurers that they should start a measurement session. The Measurer nodes (Figure 5), on the other hand, continuously listen to the RS-485 line for incoming messages and, when a message containing the ASCII character "1" is received, a measurement session with predefined duration and sampling frequency is performed. A red LED is used in all nodes to indicate the system's status: a blinking LED indicates that the Controller has transmitted a message, and a lit LED indicates that a Measurer is performing a measurement session.

2.2.4. Version C: Final Version with Coordinated Wireless Nodes. Figure 7 presents the electronic schematics and the network functionality of the third and last version of the low-cost system, named Version C. This version extends Version B by replacing cabled connections by wireless communication provided by a generic module built around the nRF24L01+ chip, a radio-frequency transceiver chip (i.e., a single chip capable of transmitting and send messages) that works at a 2.400–2.4835 GHz ISM (industrial, scientific, and medical) radio spectrum [38]. This chip has been reported to reliably produce up to 1500 transmitted packets of 25 bytes per second in range of up to 100 m in a direct line of sight, with potentially reaching up to 1200 m in a power-amplified version [39]. Such a range and data rate match many wireless SHM applications, considering that the system will work in the same way as Version B, with communication between nodes serving only as a central trigger so measurements can start synchronously [28]. Also, this may lead to a latency that partially satisfies some SHM requirements regarding time

Algorithm 2 Source code for Version B - Measurer nodes

```
1:  Initiate libraries and variables
2:  measurerGo ← 0                          ▷ To control sampling sessions
3:  inputString ← 0                         ▷ To read RS485 messages
4:  start setup
5:      Turn on resistors to keep power bank on
6:      Initiate accelerometer, microSD card module, RTC module
7:  end setup
8:  start loop
9:      if there is data on RS485 line then
10:         inputString ← data from RS485 line
11:         if inputString = 1 then
12:             inputString ← 0
13:             measurerGo ← 1              ▷ So to start measurement session
14:             Open new measurement file in microSD
15:         end if
16:     end if
17:     if measurerGo = 1 then              ▷ Start measurement session
18:         Open new measurement file in microSD
19:         Turn on LED to indicate start of sampling process
20:         Get time from RTC and save to file
21:         while elapsed time is less than sampling duration do
22:             Retrieve acceleration data from accelerometer
23:             Save to file
24:             Wait 1/(sampling frequency) seconds to get next sample
25:         end while
26:         Turn off LED to indicate end of sampling process
27:         Close measurement file
28:         measurerGo ← 0
29:     end if
30: next loop
```

FIGURE 5: Pseudocode for Version B of the system: Measurer node.

synchronization between nodes, even without a synchronization scheme besides a simultaneous centralized activation of all Measurers by the Controller node [40]. Again, Figure 7(a) refers to a generic node with components required by both Measurers and Controller nodes, such that in a real implementation some components may be removed to deploy a particular node.

Figures 8 and 9 present the pseudocode for the source code of the Measurers and Controller nodes, respectively, for

Algorithm 3 Source code for Version B - Controller node

```
Initiate libraries and variables
start setup
    Turn on resistors to keep power bank on
end setup
start loop
    while button was not pressed do
        Check if button was pressed
    end while
    Transmit value "1" in RS485 line
    Blink LED to show successful transmission
next loop
```

Figure 6: Pseudocode for Version B of the system: Controller node.

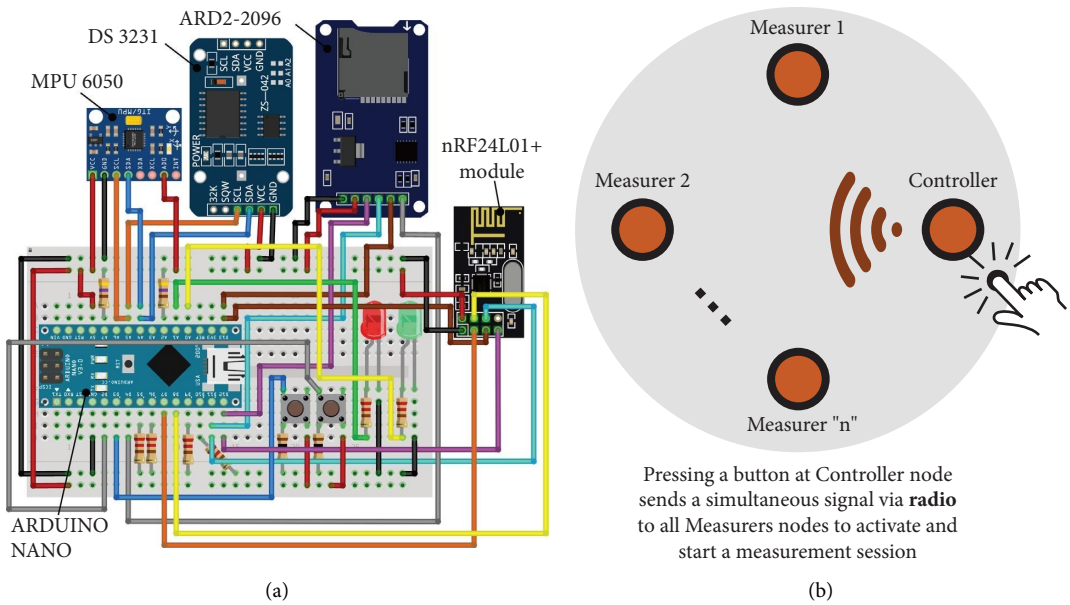

Figure 7: Version C node: (a) electronic schematic; (b) network functionality.

the Version C system. The algorithm is structured in a similar way to Version B, with the proper replacements of wired by the wireless communication protocol. In Version C, an additional button was included in the Controller node, so as to allow switching between different measurement cycle configurations without needing to upload a new source code. The Measurers' source code was accordingly updated to allow for such configuration. The red and green LEDs are used to show to the user which configuration cycle is currently selected on the system. For demonstration purposes, only two cycle configurations were implemented, but the code can be easily scaled up to include additional configurations. Figure 10 presents a photo of a Version C node as used in the characterization and validation tests.

This framework was considered adequate to comply with all the general requirements presented in Section 2.2.1. Besides the very choice of using the Arduino prototyping platform, the requirement of technical accessibility was met mainly by the choice of electronic modules, as they are all off-the-shelf components, easily found on the market, with easy-to-use and well-documented interfacing libraries available online. The simplicity requirement was met by the simplified routine executed by the system, which provides the minimum functionality necessary to perform wireless multinode vibration monitoring. Lastly, the scalability requirement was considered satisfied by the choice of the wireless solution, which may work in distances of up to 1200 m, and to the system architecture, which allows that a large-scale network be comprised of many small synchronized meshes such as the one evaluated in this work.

2.2.5. Postprocessing Interpolation Strategy. Foreseeing the necessity to deal with non-uniform sampled signals in time due to limitations intrinsic to the low-cost system (*e.g.*, low computing and memory power), a postprocessing scheme was devised, consisting of interpolating the non-uniform sampled acceleration time series into a uniformly sampled series. The adoption of a postprocessing strategy, performed in a computer with the data gathered by the low-cost system, was considered suitable by its null impact on the developed system, maintaining it as simple as possible according to the general requirements presented in Section 2.2.1.

Four different interpolation strategies were implemented. Three built-in MATLAB [41] interpolation methods (linear, spline using not-a-knot end condition, and Modified

Algorithm 4 Source code for Version C - Measurer nodes

```
Initiate libraries and variables
activateCycle1 ← false                ▷ To control sampling configuration
activateCycle2 ← false                ▷ To control sampling configuration
start setup
    Turn on resistors to keep power bank on
    Initiate accelerometer, microSD card module, RTC module
    Initiate RF module
end setup
start loop
    Listen RF line to update activateCycle1 and activateCycle2
    if a measurement cycle is to be started then
        if activateCycle1 = true then
            Set cycle 1 sampling duration and frequency
            Turn on green LED
        else
            Set cycle 2 sampling duration and frequency
            Turn on red LED
        end if
        Open new measurement file in microSD
        Get time from RTC and save to file
        Print header to file with sampling configuration
        while elapsed time is less than sampling duration do
            Retrieve acceleration data from accelerometer
            Save to file
            Wait 1/(sampling frequency) seconds to get next sample
        end while
        Close measurement file
        Turn off LED
        activateCycle1, activateCycle2 ← false
    end if
next loop
```

FIGURE 8: Pseudocode for Version C of the system: Measurers.

Algorithm 5 Source code for Version C - Controller node

```
Initiate libraries and variables
sendPayload ← false                   ▷ To control transmissions
start setup
    Turn on resistors to keep power bank on
    Initiate RF module
end setup
start loop
    if sendPayload = true or unsuccessful transmission then
        Blink LED to indicate transmission attempt
        Transmit payload and get transmission status
        sendPayload ← false
    end if
    Read cycle selection button
    if cycle selection button was pressed then
        Switch selected measurement cycle in the payload
        Turn on the LED respective to indicate the selected cycle
    end if
    Read send payload button
    if send payload button was pressed then
        sendPayload ← true
    end if
next loop
```

FIGURE 9: Pseudocode for Version C of the system: Controller.

Akima (MAkima) cubic Hermite interpolation) were used to interpolate the data in the time domain. This was enabled by the time stamping of each acceleration sample. Additionally, a scheme to use a fast Fourier transform (FFT) algorithm for non-equispaced data [42] to interpolate data in the frequency domain was devised: the MATLAB built-in non-uniform fast Fourier transform algorithm (NUFFT) was used to estimate the FFT from the non-uniformly sampled signals, and then the inverse FFT algorithm was applied to the FFT estimates to obtain uniformly sampled signals. The

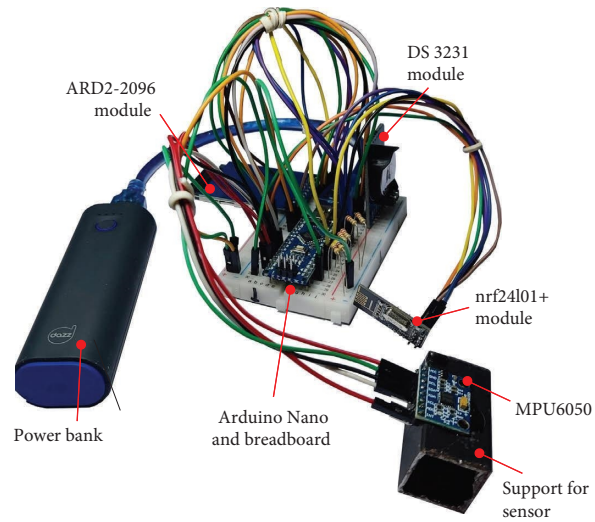

FIGURE 10: Version C node as used in the characterization and validation tests.

performance of these four interpolation strategies was assessed by comparing their derived estimates of natural frequency and mode shapes.

2.3. Characterization of the Sampling Frequency Stability. Sampling frequency stability is a major performance aspect of a data acquisition system for modal analysis, since usual modal identification methods assume uniformly synchronized sampled signals in time domain [28, 43–45]. Infringement of this hypothesis leads to introduction of numerical noises that can impair modal identification.

The sampling frequency stability of the low-cost system was characterized by constructing the histogram of time intervals between each acceleration sample collected by the system. This was possible because each acceleration sample was time stamped using the microprocessor's internal clock. The data used in this analysis were collected during the validation tests.

2.4. Impulse and Ambient Vibration Tests on a Real-Scale Slab. In order to validate and produce insights regarding the performance of the developed low-cost system, the modal identification of a real-size concrete slab was performed with a high-grade system and the developed low-cost system. The slab was subjected to impulse and ambient vibration conditions, and modal identification was performed with the experimental data to estimate natural frequencies and mode shapes. The tests were based on an operational modal analysis (OMA) framework, in which the modal analysis is performed with experimental data that do not contain information about the input forces that induced the vibration state [45].

The monitored slab was a rectangular dynamic testing platform located at the Structural Laboratory of the University of Brasilia, with dimensions of 100 mm in thickness, 6100 mm in length, and 4900 mm in width. This rectangular slab was simply supported in its two widest sides by steel profiles of W 200×19.3 type and the remaining sides were

free edges, as illustrated schematically in Figure 11. Carmona et al. [46], who also presented extensive numerical and experimental modal characterization of this slab, reported further details about the slab. Accelerations in the direction perpendicular to the slab plane were measured with both low-cost and high-grade systems at five points (A1–A5), as shown in Figure 11.

The impulse vibration condition was performed by impacting the heel of a person with approximately 75 kg standing in the centre of the slab. After impact, the person remained still until the end of the measurement session, to introduce additional inputs in the structure. Figure 12 illustrates the execution of impulse vibration test. The ambient vibration condition consisted of monitoring the slab without any external load applied to it except those normally associated with the environment, such as wind load.

The low-cost system was configured with a sampling frequency of 615 Hz, which was empirically found to be the highest sampling frequency achievable with the implemented source code. The high-grade system was composed of five piezoelectric annular seismic accelerometers model 8340 by Brüel and Kjaer (sensitivity of 10 MV/g and typical output noise of 25 $\mu g/\sqrt{\text{Hz}}$) connected to a data acquisition hardware model ADS2000. The data-acquisition system was configured to acquire data with a sampling frequency of 500 Hz. Both low-cost and high-grade systems were configured with a measurement duration of 30 seconds for the impulse test and 30 minutes for the ambient vibration test. After the interpolation algorithms were applied, the time series were downsampled to 250 Hz and the durations were reduced to 25 seconds and 18 minutes for impulse vibration and ambient vibration, respectively.

The frequency domain method (FDD) of the ARTeMIS Modal 7.2 software was used to estimate the natural frequencies and mode shapes from the experimental data. Preferably, the ARTeMIS automatic mode detection tool was used to identify the modes, so as to avoid to the best extent the subjectivity of the analyst. In some tests, however, manual mode selection was needed, as identified when discussing the results. The following preprocessing configuration was used in all analysis: detrending; decimation to a frequency range of 0 to 25 Hz, which matches the frequency range of interest in the slab; and a window resolution of 1024 points with overlapping of 66% for the estimation of power spectral density.

The low-cost system data were evaluated in their raw form (without any interpolation) and with each of the interpolation schemes implemented. The performance of the low-cost system to identify natural frequencies was evaluated by comparing the differences between the natural frequencies identified from data collected from the low-cost and high-grade systems. The performance to identify mode shapes was evaluated by computing the MAC (modal assurance criterion).

All frequency and mode shapes comparisons considered the data obtained from the high-grade system as the reference values, against which estimates from raw and postprocessed (interpolated) data obtained from the low-cost system were compared. Natural frequencies were compared.

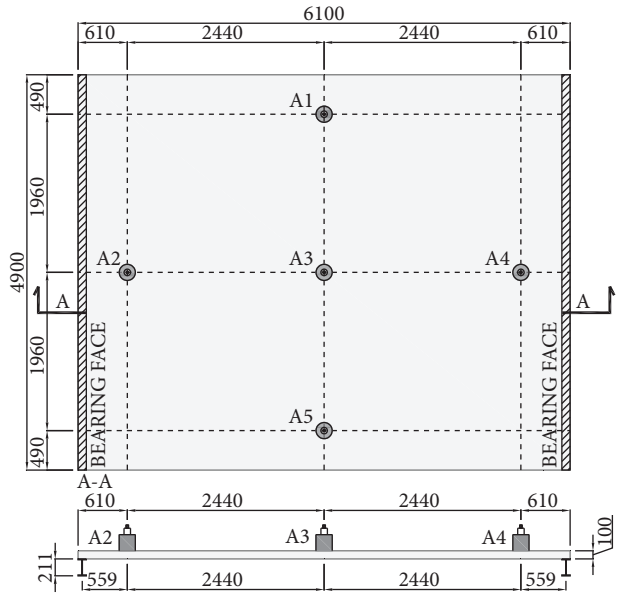

Figure 11: Schematic of monitored slab: geometry and monitored nodes (A1–A5) (units in mm).

3. Results

3.1. Sampling Frequency Characterization. Figure 13 presents the results from the sampling frequency stability characterization in the form of a histogram that indicates the relative occurrence of sampling frequencies based on data from all the measurement sessions performed in this work. The system was configured to sample data at 615 Hz. The sampling frequencies were computed as the inverse of the time intervals between two consecutive samples. The histogram is discretised in frequency bins of 1 Hz, arbitrarily chosen for better visualization.

It can be observed that more than 90% of the samples lie within 5% of the target sampling frequency of 615 Hz (i.e., approximately 584 Hz to 646 Hz). For around 5% of the samples, a sampling frequency of approximately 200 Hz or less occurs. While such performance may be sufficient for some SHM purposes, such as natural frequency estimation, the occasional decrease in sampling frequency may lead to considerable distortions in mode shape estimation, which is very sensitive to phase variability between sensor nodes.

The occasional decrease to 200 Hz may be related to long writing cycles associated to writing to the SD card, which is performed after every single sampling, as shown in Figure 8, and involves time-consuming instructions such as writing data blocks and memory allocation. This problem arises by the lack of RAM memory on the Arduino Nano microprocessor, which makes necessary that samples be constantly written to SD card instead of being held in fast access, but volatile, RAM memory and only saved on the slow, but permanent, SD card memory in the end of the measurement session.

Various solutions can be implemented to mitigate this phenomenon: hardware upgrade, source-code optimization, or data postprocessing. Examples of hardware upgrades may be the use of GPS timestamping [26], use of volatile (RAM)

Figure 12: Impulse vibration test and the monitored slab.

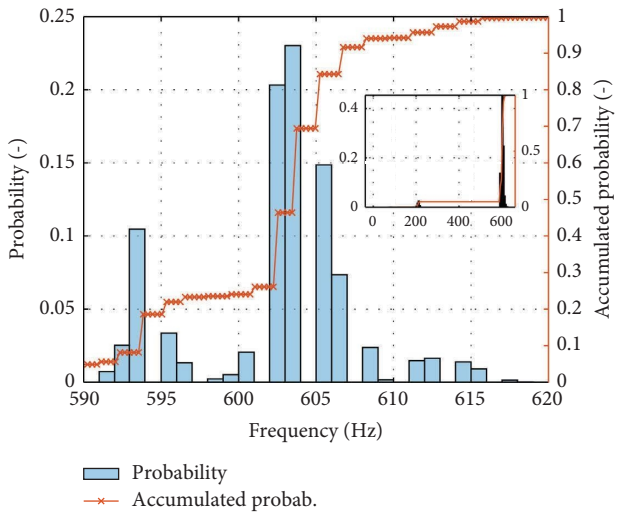

Figure 13: Histogram of sampling frequency throughout a measurement session for a given measurement node (frequency bins of 1 Hz).

memory extensions, or use of higher-performance microprocessors.Source-code optimization may involve reducing SD writing code to a minimal required, using data encoding schemes that leads to smaller files and, thus, faster to write in the SD card (e.g., binary instead of ASCII), or use of coding features like interrupts or asynchronous processing, if the microprocessor allows it. Lastly, data postprocessing may take many forms, such as interpolating the non-uniformly sampled data. Considering the general requirements of technical accessibility and simplicity intended for the developed system, the postprocessing strategy was identified as the most suitable strategy, in the context of this work, to circumvent the non-stable sampling frequency. Implementation of hardware or source-code optimization solutions would add undesired complexities, related to advanced electronics and programming topics, that could reduce the technical accessibility of the developed system.

3.2. Impulse Vibration Test. Figure 14 presents the PSD estimates obtained from the impulse vibration experiments. Figure 14(a) presents the result obtained with the high-grade system, which is presented in grey line for reference in Figures 14(b)–14(f), which present the low-cost results with raw and interpolated data. The results reported in [46] on frequencies identified by modal tests on the real-scale slab are also indicated as red vertical lines as reference, marking the values of 3.32, 15.33, and 23.05 Hz for the first three vibration frequencies. Higher modes were not captured during the impulse tests by neither of the two systems, which may be explained by the test slab being specifically designed to be highly excitable in the frequency range typical of human-induced vibrations (around 3 Hz) [46]. Higher modes could certainly be excited if synchronized loads were properly applied to the DOFs associated to their highest mode shape displacements, but, for the purpose of validating the developed system, analysis of the first three modes was deemed sufficient.

Visually, it can be observed that the results from the high-grade system clearly contain the frequencies obtained in [46], as shown by the sharp peaks coincident with the red vertical lines, indicating the validity of the modal tests performed in this work. The PSDs from the low-cost system are notably noisier than those of the high-grade system, as indicated by both the oscillating amplitudes in all frequencies and the higher overall baseline energy among all frequencies (i.e., the PSD of the low-cost system is in general higher than the PSD of the high-grade system in all frequencies). Nevertheless, even the low-cost results with no

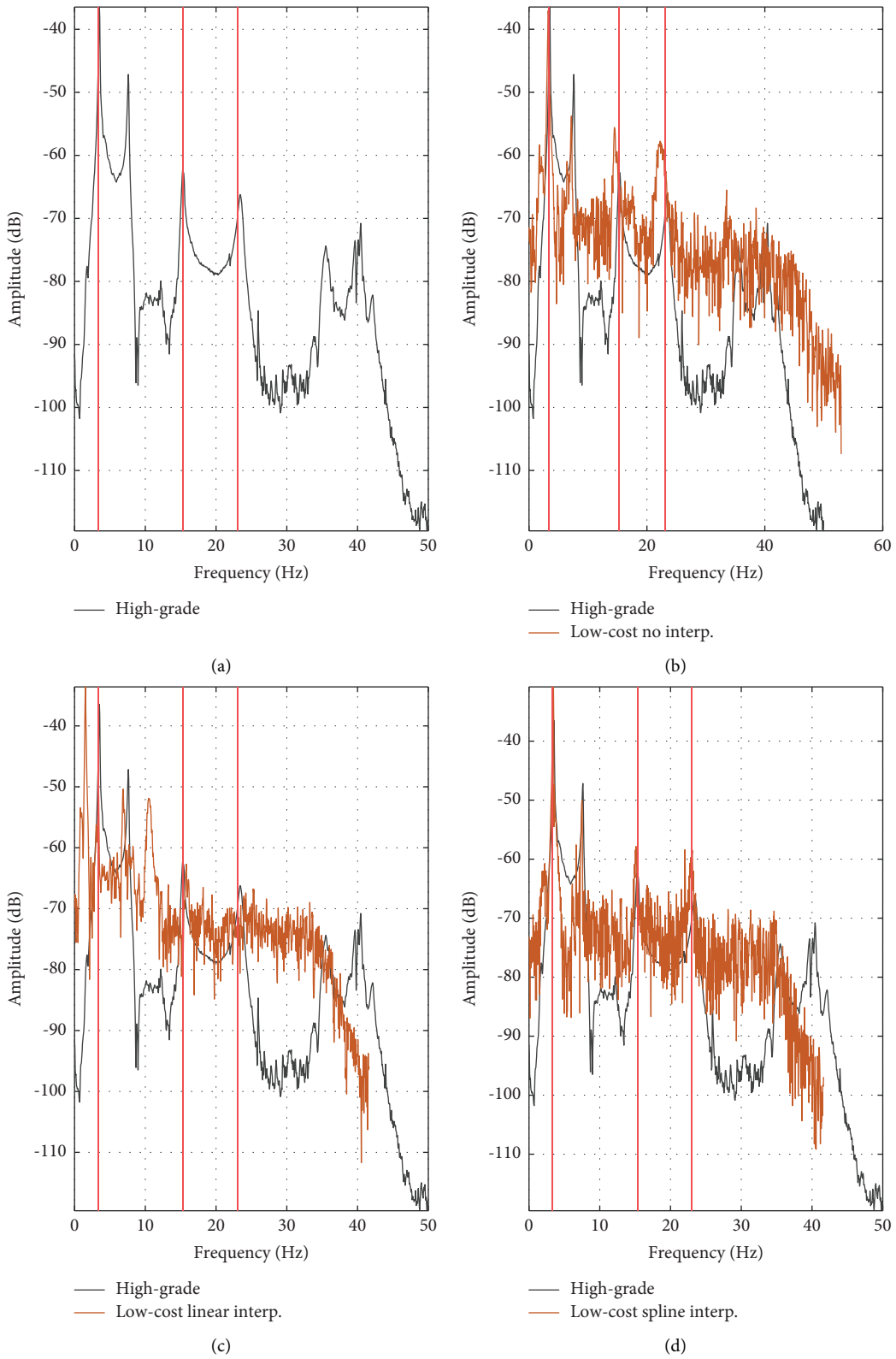

Figure 14: Continued.

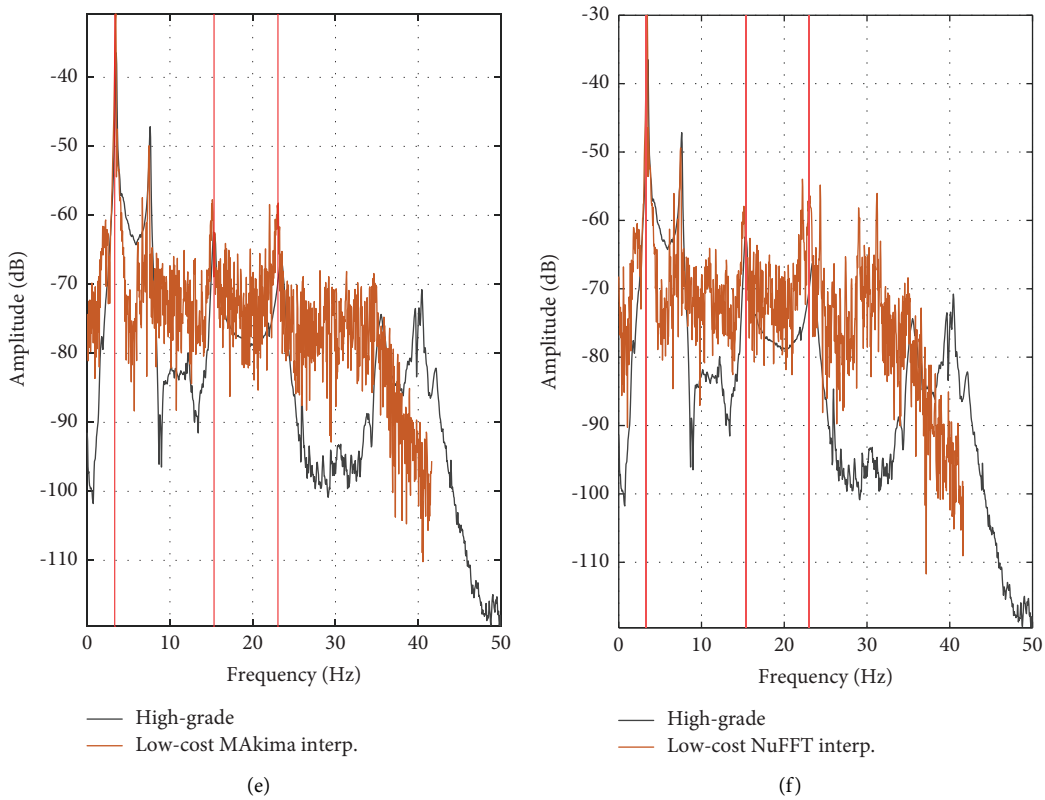

Figure 14: Power spectral density estimates from impulse vibration tests of the high-grade system and the developed system with various postprocessing interpolation strategies: (a) high-grade system; (b) low-cost system without interpolation; (c) low-cost system with linear interpolation; (d) low-cost system with spline interpolation; (e) low-cost system with MAkima interpolation; (f) low-cost system with NUFFT interpolation. The frequencies identified in [46] are marked in red vertical lines.

interpolation presented peaks that may be associated with the frequencies identified with the high-grade system, even though peaks associated with the second and third modes (i.e., 15.33 Hz and 23.05 Hz) are slightly misaligned to the high-grade system PSD. Interpolation did not visually reduce the noise but improved the alignment of the peaks associated to the vibration modes, which is expected to increase the modal identification precision.

It is also worth noting that an additional peak, at approximately 7.60 Hz, was consistently found in all experiments performed in this work, with both systems. Since such frequency was not previously reported in either the numerical or experimental results in [46], it is hypothesized that it may have been a vibration mode originated from changes occurred in the structural system of the slab since the first experimental campaign reported in [46] (i.e., due to cracking, modification in the support conditions). Therefore, this frequency was ignored in all subsequent analysis.

Table 1 shows the estimated values of the natural frequencies of the first three modes of the slab obtained in the impulse vibration test as well as the difference between the high-grade system and the low-cost system results, i.e., data with no postprocessing interpolation and with each of the four interpolation strategies. In the table, frequencies marked with " $*$ " are associated with modes identified after a manual peak selection, i.e., modes that could not be identified with the automatic mode detection tool of ARTeMIS software.

The results obtained from the high-grade system are less than 6% different from the frequencies identified in [46], which further strengthens the confidence in the modal tests performed in this work. It is worth noting that this magnitude of difference may be explained by various factors, such as differences in temperature and humidity conditions [47], resolution of the sensing network (since [46] monitored 81 nodes and this work only monitored five), and eventual changes in mass or stiffness of the slab occurred between these two tests.

The low-cost system may be considered fully validated, as even the low-cost system data without interpolation were remarkably close to the high-grade system results, with differences of a magnitude of 5%. Without interpolation, the modes could not be identified with the automatic mode identification tool of the software, suggesting that the raw data of the system may not be adequate for an automatic SHM system. Postprocessing in the form of interpolation was successful in reducing the differences to the high-grade system, which, except for linear interpolation, presented differences as low as 0.96% in relation to the high-grade system, as well as allowing automatic mode identification.

Specifically, linear interpolation of data presented poor performance in improving precision of the first mode and did not allow even a manual peak selection for the second and third modes, as can be observed in Figure 14(c), besides introducing a spurious frequency peak at around 10 Hz.

Table 1: Natural frequencies identified from the impulse vibration test.

System		Natural frequency (Hz)			Relative difference to high-grade system (%)		
		1st mode	2nd mode	3rd mode	1st mode	2nd mode	3rd mode
High-grade		3.516	15.283*	23.486		—	—
Low-cost	No interpolation	3.313*	14.544*	22.256*	5.77%	4.84%	5.24%
	Linear	3.394*	Not found	Not found	3.47%	—	—
	Spline	3.459	15.137	23.071	1.62%	0.96%	1.77%
	MAkima	3.459	15.137	23.071	1.62%	0.96%	1.77%
	NUFFT	3.459	15.137	23.031	1.62%	0.96%	1.94%

Frequencies marked with * are associated to modes identified with a manual peak selection.

Higher-order time-domain (spline and MAkima) and frequency-domain (NUFFT) interpolation methods presented identical performances for the first and second modes and marginal differences in the third mode. Differences regarding the third mode might be related to the difficulty of exciting it in the modal test, as, being a higher mode, its contribution on the final response is reduced when compared to the other two modes, and therefore its estimation is associated with higher uncertainty and variability throughout all the assessed data.

Figure 15 summarizes the mode shapes of the first three modes for the high-grade setup, low-cost system with no interpolation, and low-cost system with spline interpolation method, chosen to be illustrative of the best interpolation methods. Visually, it can be observed that interpolation improved the similarity between all mode shapes of the high-grade and low-cost system. In order to numerically compare the mode shapes, the MAC coefficients of the modes, considering that the degrees of freedom of all both systems are coincident, were computed taking the mode shapes of the high-grade system as reference. Table 2 summarizes these results and highlights in bold the MAC of the equivalent modes in each set of data (i.e., the MAC between the first modes of two datasets). MAC values of the high-grade system (in relation to itself) are indicative of the good quality of the modal test performed, as off-diagonal values are much lower than on-diagonal values, indicating low cross-correlation and high self-correlation between identified modes. Interpolation also greatly improved the correlation between all modes of the low-cost system and the high-grade system. Again, higher-order interpolation presented better results than linear interpolation, which, nevertheless, improved upon raw data that were practically not correlated to the mode shapes identified with the high-grade system data.

3.3. Ambient Vibration Test. Figure 16 presents the PSD estimates obtained from the ambient vibration experiments as well as red vertical lines indicating the three vibration frequencies reported by Carmona et al. [46]. Figure 16(a) presents the result obtained from the high-grade system, which shows clear peaks associated with the first and second modes, but not with the third mode, which might be justified by the low amount of ambient vibration available around the slab to excite this high mode. Additional peaks shown in the plot (e.g., around 12 Hz, 20 Hz, and 40 Hz) may be related to electrical noise of the system, which could not be grounded during the test.

Table 2: MAC values between the mode shapes identified by each system (high-grade and low-cost systems with and without interpolation) and the mode shapes identified by the high-grade system.

System		Mode	MAC		
			1st	2nd	3rd
High-grade		1st	**1.0000**	0.0369	0.0184
		2nd	0.0369	**1.0000**	0.0202
		3rd	0.0184	0.0202	**1.0000**
Low-cost	Raw	1st	**0.0597**	0.1534	0.1169
		2nd	0.1278	**0.0702**	0.1475
		3rd	0.2509	0.3890	**0.0424**
	Linear	1st	**0.1548**	0.1315	0.0046
		2nd	0.7768	**0.2393**	0.0117
		3rd	0.0222	0.0623	**0.6800**
	Spline	1st	**0.4339**	0.0529	0.0462
		2nd	0.0769	**0.3399**	0.0658
		3rd	0.0414	0.0411	**0.9838**
	MAkima	1st	**0.4335**	0.0530	0.0463
		2nd	0.0776	**0.3305**	0.0653
		3rd	0.0393	0.0428	**0.9848**
	NUFFT	1st	**0.4356**	0.0523	0.0476
		2nd	0.0555	**0.3660**	0.0607
		3rd	0.0976	0.0083	**0.9206**

Figures 16(b)–16(d) show examples of results obtained from the low-cost system, which do not present peaks that can be visually associated with vibration modes. In these plots, the high-grade system is not shown in reference due to the difference in scale of the energy in each result: while the results of the high-grade system have a baseline energy (average level among frequencies of interest) of approximately 100 dB, the low-cost system remains between 75 and 80 dB in all plots. These results again indicate that the low-cost system is considerably noisier than the high-grade system, which may jeopardise its application in ambient vibration tests with low excitation levels. As observed in the vibration ambient tests, interpolation did not seem to considerably change the noise level, as all low-cost system plots remained with a baseline value around 75–80 dB. The spline interpolation, shown in Figure 16(c), appears to increase the energy contribution of the first mode in the plot, but this may be just a numerical bias introduced by the interpolation, as all low frequencies appear to present progressive increase in energy. The NUFFT interpolation appeared to slightly reduce noise in comparison to raw data

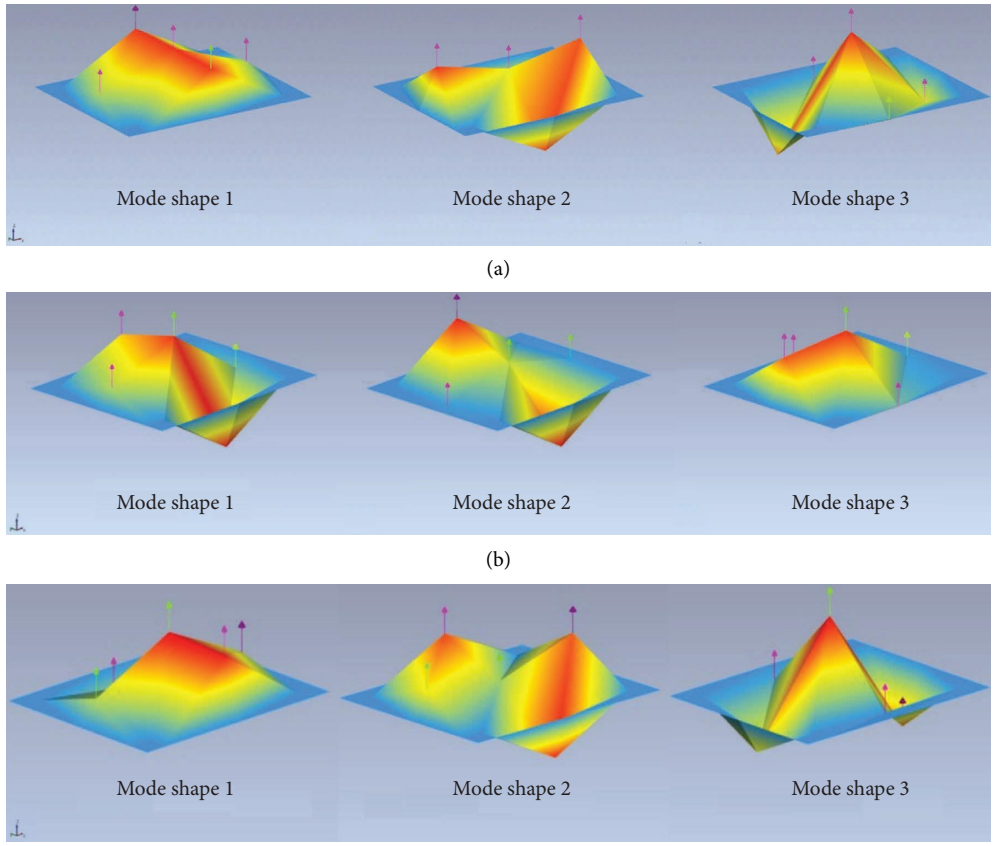

Figure 15: Mode shapes of the first three modes of impulse vibration test: (a) high-grade system; (b) raw data; (c) spline interpolation.

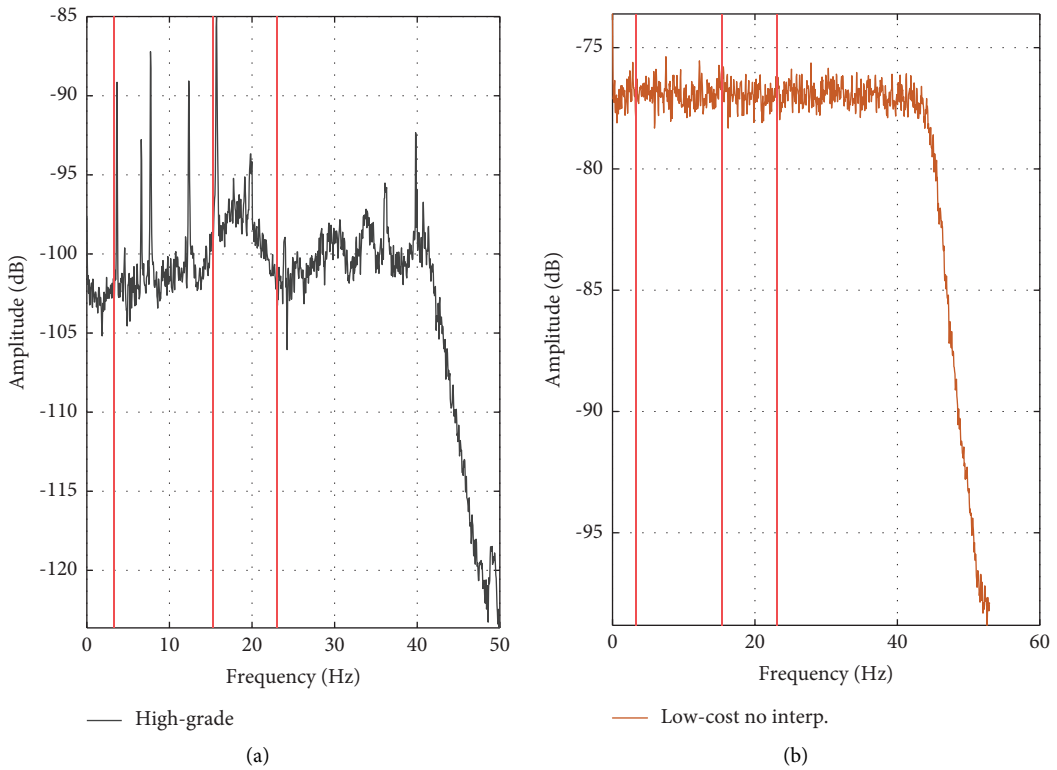

Figure 16: Continued.

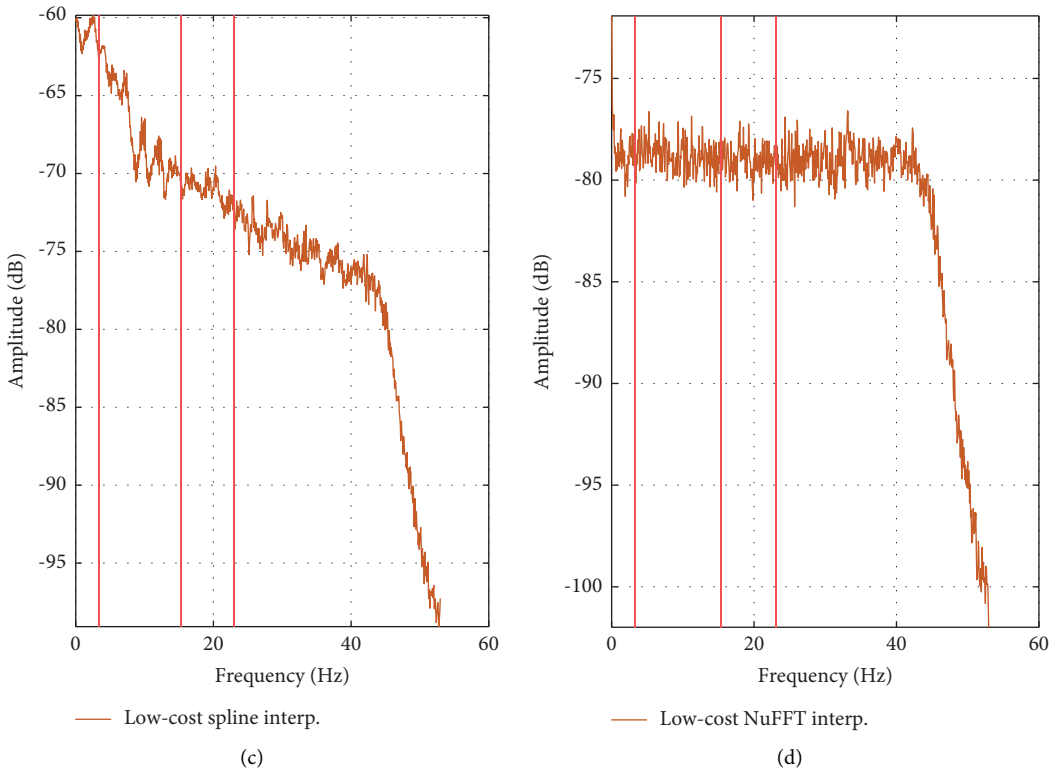

FIGURE 16: Power spectral density estimates from ambient vibration tests of the high-grade system and the developed system with various postprocessing interpolation strategies: (a) high-grade system; (b) low-cost system without interpolation; (c) low-cost system with spline interpolation; (d) low-cost system with NUFFT interpolation. The frequencies identified in [46] are marked in red vertical lines.

but did not introduce significant changes in the PSD in terms of frequency peaks. Linear interpolation and spline interpolation were not presented for the sake of brevity, as they produced similar results to the already presented methods.

Due to the poor performance of the low-cost system in ambient vibration tests, no further discussions are held regarding the mode shapes of the ambient vibration test. The automatic mode identification tool of ARTeMIS software was only able to detect the second mode in the high-grade system results, and manual selection allowed it to identify the first mode. No modes, either automatically or manually, could be identified in the ambient vibration data of the low-cost system.

4. Conclusions

This work presented the development and validation of a low-cost vibration-based SHM multinode wireless system, based on the Arduino platform, for identification of modal parameters in civil infrastructures. The stability of the system sampling frequency was characterized, and postprocessing interpolation strategies were proposed to handle identified limitations. The system was also validated by comparison to a high-grade system in modal identification tests of a real-sized slab. The main conclusions of the work were as follows:

(i) The proposed system was conceived as a minimal working example of a low-cost wireless SHM system with five sensor nodes. Effort was made to conceive it as simple and technically accessible as possible, and full hardware and software details were disclosed to allow its reproduction even by non-specialists in electronics. The system does not have evident limitations in terms of scalability and incorporation to other systems, such as IoT networks.

(ii) The sampling frequency stability test indicated that 90% of the samples were sampled with a frequency within 5% of the target sampling frequency of 615 Hz. However, around 5% of the samples are sampled with 200 Hz or lower, possibly explained by long writing cycles to the SD card memory, which introduces harsh distortions in the data that hinder modal identification with raw data. Use of interpolation, however, appeared to partially solve this issue.

(iii) All interpolation methods analysed were able to improve the quality of the low-cost system data for identification of frequency and mode shapes. Identified frequencies lied within 2% of those obtained with a high-grade system, and MAC values ranged from 0.3 to 0.9.

(iv) The proposed system was revealed to be considerably noisier than the high-grade system, with baseline power in PSD estimates being around 75–80 dB, as opposed to the high-grade system that had a 100 dB energy baseline. Because of this, no

modal identification could be performed with data obtained from the proposed system in ambient vibration tests.

Aspects of rate of data transmission or in-network processing, energy harvesting, long-term functionality and stability, or continuous time synchronization between nodes, associated to the development of wireless SHM networks and IoT, were not addressed in this work and will be so in forthcoming studies [8, 21, 26, 28]. Although undoubtedly necessary to real-world SHM of civil infrastructures, addressing these aspects would lead to an overly arid implementation for non-experts in electronics in terms of hardware and software, which were one of the targets of this work.

Data Availability

The experimental data and relevant software are available in [29] (https://github.com/renr3/lowCostWirelessVibrationSHM).

Additional Points

Highlights. (i) A low-cost vibration-based SHM multinode wireless system is fully disclosed. (ii) Sampling frequency stability of the proposed system is characterized. (iii) Interpolation methods to improve data quality in post-processing are presented. (iv) Frequency estimates are 2% of those obtained with a high-grade system. (v) MAC values ranged from 0.3 to 0.9 in relation to a high-grade system.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Renan Rocha Ribeiro, Rafael de Almeida Sobral, and Ian Barreto Cavalcante. The first draft of the manuscript was written by Renan Rocha Ribeiro and Rafael de Almeida Sobral, and all authors commented on previous versions of the manuscript. All authors have read and approved the final manuscript.

References

[1] Y. An, E. Chatzi, S. Sim, S. Laflamme, B. Blachowski, and J. Ou, "Recent progress and future trends on damage identification methods for bridge structures," *Structural Control and Health Monitoring*, vol. 26, no. 10, pp. 1–30, 2019.

[2] K. Chintalapudi, T. Fu, J. Paek et al., "Monitoring civil structures with a wireless sensor network," *IEEE Internet Computing*, vol. 10, no. 2, pp. 26–34, 2006.

[3] D. Montalvão, N. M. M. Maia, and A. M. R. Ribeiro, "A review on vibration-based structural health monitoring with speacial emphasis on composite materials," *The Shock and Vibration Digest*, vol. 38, no. 4, p. 49, 2006.

[4] P. Cawley and R. D. Adams, "A vibration technique for non-destructive testing of fibre composite structures," *Journal of Composite Materials*, vol. 13, no. 2, pp. 161–175, 1979.

[5] M. Palacz and M. Krawczuk, "Vibration parameters for damage detection in structures," *Journal of Sound and Vibration*, vol. 249, no. 5, pp. 999–1010, 2002.

[6] S. Das and P. Saha, "A review of some advanced sensors used for health diagnosis of civil engineering structures," *Measurement*, vol. 129, pp. 68–90, 2018.

[7] P. Cawley, "Structural health monitoring: closing the gap between research and industrial deployment," *Structural Health Monitoring*, vol. 17, no. 5, pp. 1225–1244, 2018.

[8] A. B. Noel, A. Abdaoui, T. Elfouly, M. H. Ahmed, A. Badawy, and M. S. Shehata, "Structural health monitoring using wireless sensor networks: a comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1403–1423, 2017.

[9] A. Girolami, F. Zonzini, L. De Marchi, D. Brunelli, and L. Benini, "Modal analysis of structures with low-cost embedded systems," in *Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, Italy, May 2018.

[10] J. J. Villacorta, L. Del-Val, R. D. Martínez et al., "Design and validation of a scalable, reconfigurable and low-cost structural health monitoring system," *Sensors*, vol. 21, no. 2, p. 648, 2021.

[11] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, and S. Glaser, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 2007 6th International Symposium on Information Processing in Sensor Networks*, IEEE, Cambridge, MA, USA, April 2007.

[12] S. Komarizadehasl, B. Mobaraki, H. Ma, J. A. Lozano-Galant, and J. Turmo, "Development of a low-cost system for the accurate measurement of structural vibrations," *Sensors*, vol. 21, no. 18, p. 6191, 2021.

[13] S. Hou and G. Wu, "A low-cost IoT-based wireless sensor system for bridge displacement monitoring," *Smart Materials and Structures*, vol. 28, no. 8, Article ID 085047, 2019.

[14] F. Aswin, I. Dwisaputra, and R. Afriansyah, "Online vibration monitoring system for rotating machinery based on 3-axis MEMS accelerometer," *Journal of Physics: Conference Series*, vol. 1450, no. 1, Article ID 012109, 2020.

[15] V. T. hoa, I. Akacha-helal, J. David, and M. Pietrzak-david, "Wireless motor vibration monitoring with MEMS accelerometers ADXL203 sensor micro controller ARDUINO board emitter wi-fi link," in *Proceedings of the 2019 21st European Conference on Power Electronics and Applications (EPE '19 ECCE Europe)*, pp. 1–9, Genova, Italy, September 2019.

[16] A. I. Ozdagli, B. Liu, and F. Moreu, "Low-cost, efficient wireless intelligent sensors (LEWIS) measuring real-time reference-free dynamic displacements," *Mechanical Systems and Signal Processing*, vol. 107, pp. 343–356, 2018.

[17] S. Jeong, S. Cho, and S. H. Sim, "Integrated cable vibration control system using wireless sensors," *SPIE Proceedings*, vol. 10168, 2017.

[18] Z. Sun, S. J. Dyke, F. Pena, and A. Wilbee, "Development of Arduino based wireless control system," *SPIE Proceedings*, vol. 9435, 2015.

[19] M. Susanto, R. Saputra, S. S. Herlinawati, and S. Alam, "Prototype of sensor node for low-cost machine vibration monitoring system using accelerometer sensor," in *Proceedings of the 2020 IEEE International Conference on Communication, Networks and Satellite, Comnetsat*, pp. 222–226, Batam, Indonesia, December 2020.

[20] R. D. De La Torre, G. A. E. Pasobillo, M. F. Rebueno, D. P. Sunga, B. J. J. Esguerra, and R. Concepcion, "Vibration-based structural health monitoring system for bridges using ADXL345 accelerometer with MATLAB standalone application," in *Proceedings of the 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM*, New York, NY, USA, September 2020.

[21] M. Mishra, P. B. Lourenço, and G. V. Ramana, "Structural health monitoring of civil engineering structures by using the internet of things: a review," *Journal of Building Engineering*, vol. 48, Article ID 103954, 2022.

[22] A. Abdelgawad and K. Yelamarthi, "Internet of things (IoT) platform for structure health monitoring," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 6560797, 10 pages, 2017.

[23] C. Scuro, P. F. Sciammarella, F. Lamonaca, R. S. Olivito, and D. L. Carni, "IoT for structural health monitoring," *IEEE Instrumentation and Measurement Magazine*, vol. 21, no. 6, pp. 4–14, 2018.

[24] F. Di Nuzzo, D. Brunelli, T. Polonelli, and L. Benini, "Structural health monitoring system with narrowband IoT and MEMS sensors," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 16371–16380, 2021.

[25] E. Hidalgo Fort, P. Blanco-Carmona, J. R. Garcia-Oya et al., "Wireless and low-power system for synchronous and real-time structural-damage assessment," *IEEE Sensors Journal*, vol. 23, no. 12, pp. 13648–13658, 2023.

[26] K. Y. Koo, D. Hester, and S. Kim, "Time synchronization for wireless sensors using low-cost gps module and arduino," *Frontiers in Built Environment*, vol. 4, no. 1, pp. 1–16, 2019.

[27] G. Zhang, C. Moutinho, and F. Magalhães, "Continuous dynamic monitoring of a large-span arch bridge with wireless nodes based on MEMS accelerometers," *Structural Control and Health Monitoring*, vol. 29, no. 7, pp. 1–21, 2022.

[28] J. Cao and X. Liu, "Structural health monitoring using wireless sensor networks," in *Mobile and Pervasive Computing in Construction*, C. J. Anumba and X. Wang, Eds., Wiley-Blackwell, Oxford, UK, 1st edition, 2012.

[29] R. Ribeiro, "Low cost wireless vibrations," *GitHub Repository*, 2023.

[30] Arduino, "What is arduino?/why arduino?" 2015, https://www.arduino.cc/en/Guide/Introduction.

[31] Arduino, "Nano," 2022, https://docs.arduino.cc/hardware/nano.

[32] R. R. Ribeiro and R. D M. Lameiras, "Evaluation of low-cost MEMS accelerometers for SHM: frequency and damping identification of civil structures," *Latin American Journal of Solids and Structures*, vol. 16, no. 7, pp. 1–26, 2019.

[33] InvenSense, *M. P. U.-6000 and MPU-6050-Product Specification*, InvenSense, Sunnyvale, CA, USA, 2013.

[34] Wiltronics, "ARD2-2096 microSD card reader module," 2018, https://www.wiltronics.com.au/wp-content/uploads/datasheets/ARD2-2096.Data.Sheet.pdf.

[35] Maxim Integrated, *DS 3231 RTC General Description*, Maxim Integrated, San Jose, CA, USA, 2015.

[36] Maxim Integrated, *Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers*, Maxim Integrated, San Jose, CA, USA, 2003.

[37] Maxim Integrated, *Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems-TIA-485-A*, Maxim Integrated, San Jose, CA, USA, 1998.

[38] Maxim Integrated, *Semicondutor nRF24L01+Single Chip 2.4GHz Transceiver-Preliminary Product Specification v1.0*, Maxim Integrated, San Jose, CA, USA, 2008.

[39] V. Kulasekara, S. Balasooriya, J. Chandran, and I. Kavalchuk, "Novel low-power NRF24L01 based wireless network design for autonomous robots," in *Proceedings of the 2019 25th Asia-Pacific Conference on Communications, APCC*, pp. 342–346, Ho Chi Minh City, Vietnam, December 2019.

[40] V. Krishnamurthy, K. Fowler, and E. Sazonov, "The effect of time synchronization of wireless sensors on the modal analysis of structures," *Smart Materials and Structures*, vol. 17, no. 5, Article ID 055018, 2008.

[41] MathWorks, "Documentation," 2017, https://www.mathworks.com/help/matlab/.

[42] A. Dutt and V. Rokhlin, "Fast fourier transforms for nonequispaced data," *SIAM Journal on Scientific Computing*, vol. 14, no. 6, pp. 1368–1393, 1993.

[43] P. D. Welch, "The use of fast fourier Transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[44] B. Peeters and G. De Roeck, "Stochastic system identification for operational modal analysis: a Review," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 4, pp. 659–667, 2001.

[45] E. Reynders, "System identification methods for (operational) modal analysis: review and comparison," *Archives of Computational Methods in Engineering*, vol. 19, no. 1, pp. 51–124, 2012.

[46] J. E. C. Carmona, S. M. Avila, and G. Doz, "Proposal of a tuned mass damper with friction damping to control excessive floor vibrations," *Engineering Structures*, vol. 148, pp. 81–100, 2017.

[47] B. Peeters, "System identification and damage detection in civil engineering," Doctor Thesis, Katholieke Universiteit te Leuven, Leuven, Belgium, 2000.