A low-cost system for continuous dynamic monitoring and autonomous data analysis using the Internet of Things

Bernardo Lopes Poncetti[1], Dianelys Vega Ruiz[1], Marcos Massao Futai[1]

[1]*Dept. of Civil Engineering, GeoInfra USP, University of São Paulo*
Av. Prof. Almeida Prado, trav. 2, n83, 05508-900, São Paulo, Brazil
bernardoponcetti@gmail.com

Abstract. Structural health monitoring using the Internet of Things (IoT) is the latest technique employed in the field of structural damage detection. Although conventional systems based on commercial sensors, such as piezoelectric accelerometers, provide high accuracy, their high cost often limits their application. To address this issue, one possibility has been the development of strategies using low-cost sensors. However, there are several challenges to be addressed, such as the optimization of the low-cost devices to increase their reliability in reading and processing data, and the development of data storage and data transmission strategies, mainly for dynamic monitoring applications, which requires working with large amounts of data. In this regard, this paper presents the development of a low-cost SHM solution, which is able to collect, store, process and transmit vibration data to the cloud from an instrumented aluminum beam. For this purpose, a prototype is developed using an ESP32 board, an MPU6050 triaxial accelerometer and a microSD card. A completely low-cost system is adopted, where the data processing and availability of results to the final user is performed in the free version of ThingSpeak IoT platform from MathWorks. As a result, a fully automatic dynamic monitoring strategy able to collect, store and transmit raw vibration data to the cloud is developed. Then, the raw acceleration data is processed and analyzed in the cloud, where the Fast Fourier Transforms (FFT) are computed and visualized in quasi-real time.

Keywords: SHM, IoT, MEMS accelerometer, ESP 32, ThingSpeak.

1 Introduction

The technological advances in the last decades have enabled the creation of many devices and systems for Structural Health Monitoring (SHM). The recent creation of new devices with higher capacities and also lower prices, and the aging of the infrastructure have provided a growing interest in civil engineering professionals to implement SHM systems for monitoring the structural integrity of buildings and infrastructure.

The use of systems with connectivity to the internet becomes essential to assist in the collection, management, and storage of data for SHM. The Internet of Things (IoT) appears widely requested in all fields of engineering and is being widely developed for SHM systems in the field of civil engineering, as shown in Jie et al. [1], Muttillo et al. [2], Nuzzo et al. [3], Mohapatra et al. [4] and Dabbakuti [5].

In this context, understanding that the use of IoT a for an SHM system can significantly improve the infrastructure operational and maintenance decision, this article presents a routine of an IoT application for structural dynamic monitoring using low-cost sensors, exemplifying a strategy for collecting, sending, storing and visualizing data. A prototype of an electronic device with a routine for collecting, storing, and sending data to the cloud is developed in this work. The evaluation of the prototype is done using an aluminum beam in the laboratory, instrumented with commercial accelerometers. The automatic strategy to collect, send, analyze, and provide the results to the user is resumed explained in this paper.

It is important to say that this project intended to create a complete SHM system using IoT with low-cost solutions. Therefore, all the solutions adopted are made with low-cost sensors and free IoT platforms as the ThingSpeak adopted in this article.

The ThingSpeak is an IoT Cloud from Mathworks that has a free and paid version. Searching for a low-cost strategy this paper presents a routine that explores the free version of the Mathworks platform taking care of its limitation. Because of its restriction in the data receiving frequency and analysis, the ThingSpeak is mostly used for static measurements, as seen in Glisié and Inaudi [6]. Even though, this IoT platform has an advantage the possibility to process MatLab coding online in the Cloud allowing autonomous data preparation and analysis, and using some strategies may be used for dynamic monitoring as shown in this article.
.

2 Methodology

According to Moradi et al. [7], to implement a structural monitoring process is necessary to establish the monitoring aim and select an adequate strategy, identifying and selecting representative parameters to be monitored. It is necessary selecting an appropriate monitoring system, design a sensor network, establish the monitoring schedule, and plan data exploitation. Therefore, for the development of this project, the methodology applied followed the necessities to implement a monitoring process listed by Moradi et al. [7], solving some issues ordered in Fig. 1 the monitoring strategy methodology was implemented.

The starting point was determining an aim and the relevant parameter, after determining the monitored issue the monitoring strategy was defined, including the monitoring system, monitoring schedule and sensor network, and finally doing de data exploitation making the results available to the user.

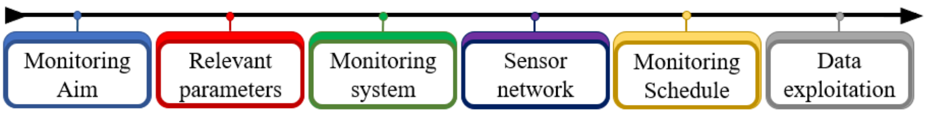

Fig. 1. Procedure followed for the monitoring system development.

3 Low-cost monitoring system

3.1 Monitoring Aim and Relevant Parameters

The project is designed to be used for dynamic monitoring of transportation infrastructure. Several situations of structural failure or where is need an intervention in the transportation infrastructure are reported in the literature, some examples can be seen in Xu et al. [8], Selvakumaran et al. [9], Cao et al. [10], Wang et al. [11], Khodabandehlou et al. [12]. Failure cases can be caused by extreme environmental situations, deterioration over time, special solicitations, or inadequate use of the infrastructure as seen in Wang et al. [11] with the passage of overloaded vehicles. Such situations can cause accidents and generate expressive losses. One way to prevent such occurrences is the use of continuous monitoring of infrastructures to predict catastrophes or assist in the management of inspection and maintenance. The monitoring system may also be useful during specific operations, such as rehabilitation, to guarantee its integrity, and not only during operational situations.

The final objective of this project is to create a fully autonomous low-cost monitoring system that can identify if there is any damage in the structure. To do it, the system analyzes the dynamic structural response. So, the collected parameters are the acceleration and temperature, both are relevant for damage identification and localization. As shown in Satpal et al. [13] the structural dynamic response may be used with Machine Learning (ML) to access the integrity of the element.

Therefore, this paper only shows a low-cost dynamic monitoring system for structural components of the transportation infrastructure, the developing system intends to use ML technics to classify the data into the ThingSpeak Cloud.

3.2 Monitoring System and Schedule

For the low-cost prototype development, it has been used an ESP-32 WROOM DevKit, a MPU 6050 3-Axis Gyroscope Accelerometer module, a microSD card adapter, and a 4 GB micro SDHC memory module, Fig 2.

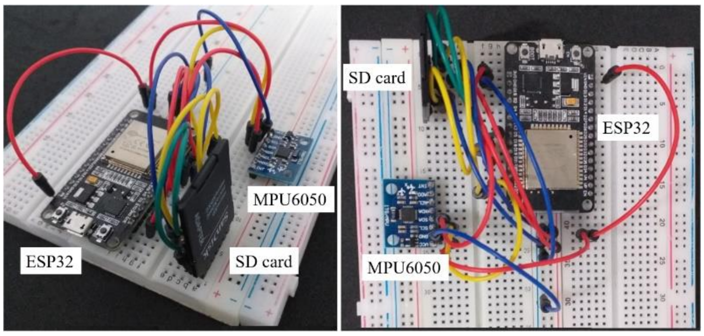

Fig. 2. Prototype used on the low-cost monitoring system proposal.

Developed to be used for dynamic readings of transportation infrastructure, the prototype has a routine that only starts the data storage when the accelerometer registers values (Ac_read) higher than the reference value (AC_ref), Fig 3, this trigger may be activated by the passage of a vehicle on the structure. An example of an application for the prototype is for railway structures where the sensor will activate only during the train passage. Fig. 3 shows the routine implemented for the prototype. When Ac_read is higher than the Ac_ref the device creates a new document in the SD Card where the sampling data will be stored then the routine of sampling and storage data is started. This routine has some time controls to guarantee the reading frequency rate and the sampling interval. The data collected may be saved in a .txt or .csv document type in the SD Card. For the prototype used in this demonstration, all the data is saved in a text document.

During the sampling stage the ESP 32 works as a Data Acquisition System, after finished the sampling interval it stops the accelerometer readings and starts the routine for sending data to ThingSpeak Cloud. In this phase, the ESP32 work as a gateway, sending information to the internet. The ThingSpeak accepts data sent by HTTP request or protocol MQTT.

The dynamic data may be larger than the memory of ESP32 or than the amount of data accepted by ThingSpeak for one single data sending-request. To solve this problem the prototype separates an amount of data adequate with the ESP32 and ThingSpeak capacity, sending several JSONs with the maximum amount of data possible. The routine that sends data repeats until all the data had been sent to the Cloud, after this, the prototype read the temperature collected by the MPU6050 and sends a last request to the ThingSpeak with the temperature and with a trigger to start the data verification routine automated in the Cloud.

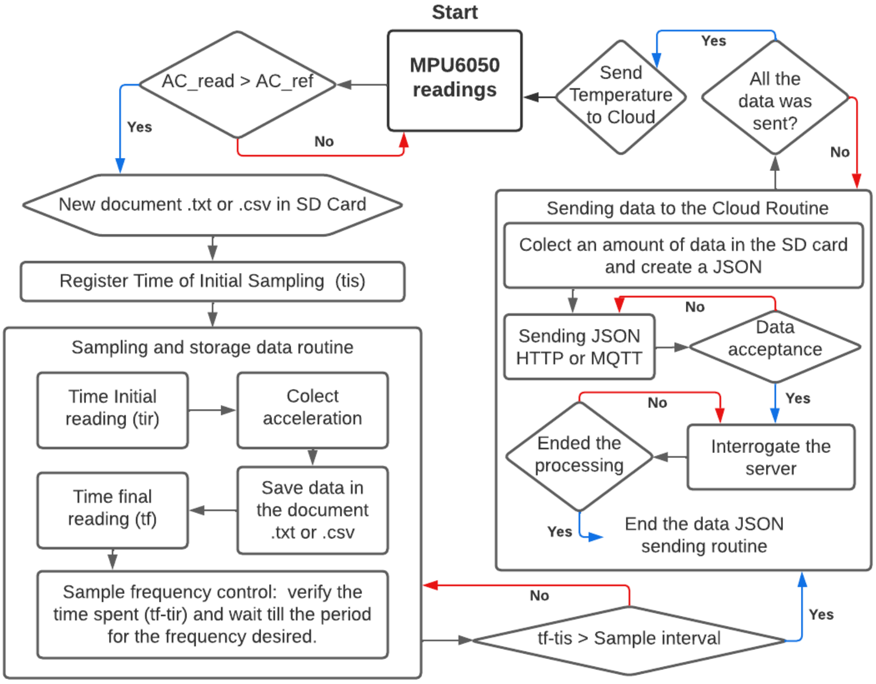

Fig. 3. Routine implemented in the prototype.

3.3 Sensor Network

Several strategies may be used to design a sensor network. Some possibilities are the use of a private local server to store, analyze and send the data of all sensors to the cloud, or even the sensor may provide the data already treated using board computation. Mostly the time sends only the specific value, the frequency obtained by the Fourie Transformation for example, to the cloud is the best strategy. However, the system developed in this work sends the raw data to the cloud. This strategy is adopted to have more possibilities for using ML for damage identification and localization.

The ThingSpeak allows a free account to have 4 channels, each channel may have 8 fields. The paid version possible to have more channels. These fields are the local ones where the sensors store data in the Cloud. However, it is important to observe that, the server has limitations for the amount of data acceptance of each channel, especially for a free account, the free version may send one data request every 15 seconds, and the paid version each second. The data request may be a JSON with a data bunker containing a group of values. It is important to observe that, when the data request has too many entries the server will take more than 15 seconds to process all data. It is important to wait for all the data processing of each channel before sending more information, the excess of requests may cause data loss. Because of this limitation, the routine for sending data to the Cloud always interrogates the server, Fig. 3, waiting for the ending of data processing.

The time necessary to finish the data processing of a free account influences in the time necessary to send all information to the server. To minimize this problem and to help with the data loss control, the system developed in the work uses 1 channel for each sensor installed and 1 channel only for data evaliation (Fig. 4). The number of channels can be higer with a paid account, as the ThingSpeak also allows to collect data from different accounts, which allows the use of more channels.

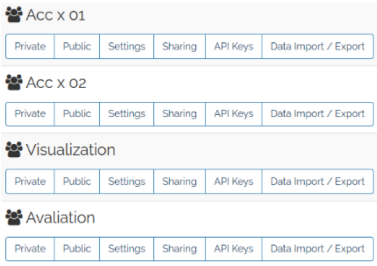

Fig. 4. ThingSpeak channels.

Fig. 5 shows how works the sensor network, and the communication with the ThingSpeak. The prototypes are directly connected to ThingSpeak. It sends the raw data to a specific channel. After all the acceleration data had been sent to the cloud, the sensors send the temperature and a trigger command to the Evaluation Channel where the Machine Learning algorithm for damage detection will be started. The prototypes aren't doing communication with each other, all of them work independently of each other, which is a problem for the sensor synchronization of this system.

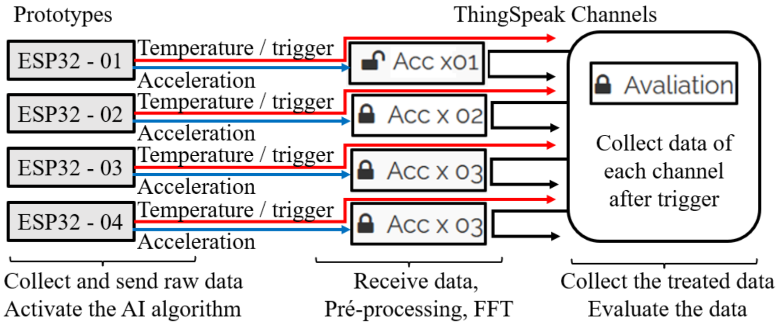

Fig. 5. Sensors connection with ThingSpeak.

3.4 Data exploitation

The ThingSpeak has as default the registering of data at the time of its entrance into the server. When sent a data bunker, the information will be clustered together as Fig. 6. (a). Each amount of data represents the values registered in each JSON sent to the cloud. This basic data presentation of the ThingSpeak isn't useful for dynamic readings. However, the data may be treated and re-organized online by the platform obtaining the Fig. 6 (b) that is the raw data collect by the ESP32. The ThingSpeak has his own programming environment that allows the user to rum all MathLab code online.

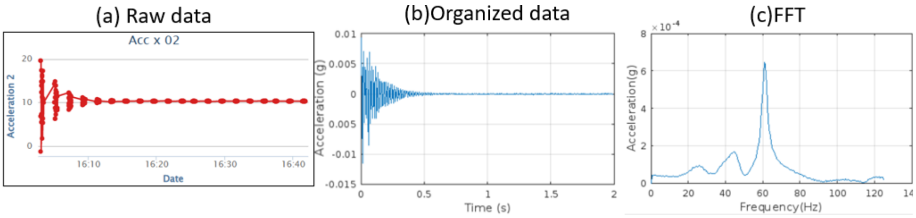

Fig. 6. Data exploitation

Using the online tool, the system is set to do all the data preparation and analysis. The Receive Data channels (Fig. 5) organize all the raw information and do the Fast Fourier Transform (as shown in Fig. 6. (c)) and the result is shown to the user. After the trigger is received by the Evaliation Channel, the data is verified by a Machine Learning algorithm trained to verify the occurrence of damage.

The ThingSpeak allows the users to configure their dashboard, making it easy for the final user to make visible the important data. Fig. 7 shows an example where is presented 2 fields and 2 graphs automatically created by the MathWorks cloud. The Field1 Chart is the raw data received, and Field 8 will be used to show the result of the ML algorithm trained to damage identification. The Acceleration X and the Fourier chart show the results of the automatic data treatment and analysis made by the routine implemented in the ThingSpeak.

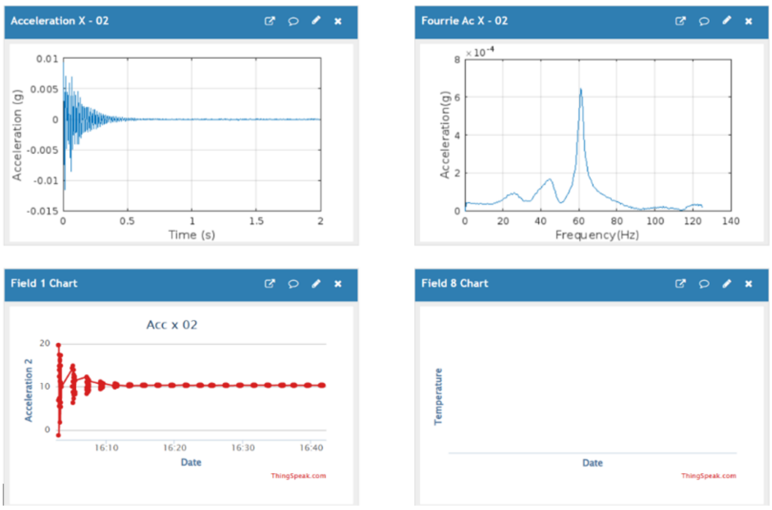

Fig. 7. Example of Dashboard available in the ThingSpeak.

4 Conclusion

The developed dynamic monitoring strategy maintained the proposal of being a low-cost IoT system. Its components can be purchased for amounts that together would cost less than 50 dollars for each prototype.

Although it has some limitations, the ESP32 has potential for IoT monitoring systems. One limitation is its internal memory which can be compensated with an external memory module such as an SD card as done in this article. However, even with the SD card helping to store data, the internal memory of the ESP32 was the limiting factor in the amount of data sent in each JSON.

Regarding the sending and processing of data, the developed routine was sufficient to guarantee the sending without the loss of the values. However, the submission process proved to be slow, one of the reasons being the speed of the free version of ThingSpeak. However, even with the limitation of the ThingSpeak free version it is proved that with the correct strategy, the platform may use for SHM porpoise.

The MathWorks IoT platform offers the possibility of applying MatLab routines in an online environment, enabling the processing of received data. Using ThingSpeak tools, it is possible to automate data analysis and generate an iterative dashboard that is easy for the user to operate. Data can be presented on the MathWorks platform itself, as well as exported to other platforms. Therefore, for a low-cost IoT solution, as proposed in this article, the platform is a valid alternative for SHM, mainly because it has powerful tools even in the free version allowing automatic data analysis.

Acknowledgements. The authors would like to thank the VALE Cathedra Under Rail for financing the project Tunnel 4.0. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] L. Jie, Z. Hui, L. Tong, Y. Shao-Wei. Application of Internet of Things in Tunnel Engineering – A Review.EJGE. v.20. 2015.
[2] M. Muttillo, V. Stornelli, R. Alaggio, R. Paolucci, L. Battista, T. Rubeis, G. Ferri. Strucutral Health Monitoring: An IoT Sensor system for Structural Damage Indicator Evaluation. *Sensors*. 2020.
[3] F. Nuzzo, D. Brunelli, T. Polonelli, L. Benini. Structural Health Monitoring System with Narrowband IoT and MEMs Sensors. *IEE Sensors Journal*. 2021.
[4] A. G. Mohapatra, J. Talukdar, T. C. Mishra, S. Anand, A. Jaiswal, A. Khanna, D. Gupta. Fiber Bragg grating sensors driven structural health monitoring by using multimedia-enabled iot big data technology. *Multimidia Tools and Applications*. 2022.
[5] J. R. K. Dabbakuti, A. Jacob, V. R. Veeravalli, R. K. Kallakunta. Implementation of IoT analytics ionospheric forecasting system based on machine learning and ThingSpeak. *IET Radar Sonar Navigation*. 2020.
[6] B. Glisié, D. Inaudi. Fibre optic methods for structural health monitoring. Wiley, England, 2007.
[7] P. Moradi, M. J. Asadi, N. Ebrahimzadeh, B. Yarahmadi. Ilam tunnels inspection, maintenance, and rehabilitation: A case stud. *Tunneling and underground Space Technology*, 2021.
[8] S. Xu, E. Ma, J. Lai, Y. Yang, H. Liu, C. Yang, Q. Hu. Diseases failures characteristics and countermeasures of expressway tunnel of water rich strata: A case study. *Engineering Failure Analysis*. 2022.
[9] S. Selvakumaran, S. Plank, C. Gieb, C. Rossi, C. Middleton. Remote monitoring to predict bridge scour failure using Interferometric Synthetic Aperture Radar (InSAR) stacking techniques. *International Journal of Applied Earth Observation and Geoinformation*. 2018.
[10] R. Cao, A. K. Agrawai, S. El-Tawil. Overheight impact on bridges: A computational case study of the Skagit River bridge collapse. *Engineering Structures*. 2021.
[11] L. Wang, H. J. Hwang, Z. Luo, C. H. Juang, J. Xiao. Probabilistic back analysis of slope failure – A case study in Taiwan. *Computers and Geotechnics*. 2013.
[12] H. Khodabandehlou, G. Pekcan, M. S. Fadali. Vibration-based structural condition assessment using convolution neural networks. Structural Control and Health Monitoring. 2018.
[13] S. B. Satpal, A. Guha, S. Banerjee. Damage identification in aluminum beams using support vector machine: Numerical and experimental studies. *Structural Control and Health Monitoring*. 2016.