Edge Processing for Frequency Identification on Drone-Deployed Structural Health Monitoring Sensor Nodes

Ryan Yount[a], Joud N. Satme[a], David P. Wamai[a,b], and Austin R.J. Downey[a,c]

[a]Department of Mechanical Engineering, University of South Carolina, Columbia, SC, USA, 29201

[b]Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, 29201

[c]Department of Civil and Environmental Engineering, University of South Carolina, Columbia, SC, USA, 29201

ABSTRACT

For rapid civil infrastructure assessment following natural and man-made emergencies, the utilization of minimally invasive and cost-effective drone deployable sensor packages has the potential to become a valuable tool. Although compact sensors with wireless data transfer capabilities have proven effective in monitoring the structural dynamics of infrastructure, these systems require data processing to occur externally, frequently off-site. These extra steps impede the high-speed assessment of a structure's state. Difficulties can arise when the transmission is unfeasible due to degraded communication links during natural or man-made emergencies. Additionally, off-site data processing can add unneeded interruptions to actions that can be taken by emergency personnel after infrastructure damage. To enhance the effectiveness of sensor packages in expediting infrastructure assessment, incorporating real-time data analysis through embedded edge computing techniques emerges as a promising solution. The objective of this work is to demonstrate on-device data processing for frequency-based structural health monitoring techniques using drone-deployable sensors. This approach advances the effectiveness of drone-deployable sensors in rapid infrastructure assessment by mitigating their susceptibility to errors or delays in data communications. The proposed approach computes the frequency components of vibration measurements taken from a structure of interest, for example, the monitoring of a bridge immediately following a damaging event such as a flood. This work presents contributions in terms of outlining a methodology that emphasizes the hardware-based implementation of edge computing algorithms and examines the required on-device performance and resource utilization for structural health monitoring at the edge. The execution time for the sensor's edge computing functions was profiled, resulting in an additional 9.77 seconds per test, an advancement over traditional transmit and analyze methods.

Keywords: edge computing, drones, sensors, structural health monitoring, damage detection

1. INTRODUCTION

The assessment of the structural health of civil infrastructure in the aftermath of natural disasters and man-made emergencies is a critical area for advancement.[1] Extreme weather conditions and environmental factors often render structures inaccessible or dangerous to inspect and maintain, posing significant risks to human operators. Moreover, accessibility issues can arise under normal circumstances, compounded further by the aftermath of emergencies. Traditionally, structural health monitoring (SHM) has predominantly been performed by on-site work crews, necessitating substantial investments in both time and equipment for a thorough examination of each structure. These conventional approaches typically involve the collection of data, which is then processed off-site, which inherently delays the availability of crucial real-time insights into a structure's dynamic behavior.

Further author information: (Send correspondence to Austin R.J. Downey)
Austin R.J. Downey: E-mail: austindowney@sc.edu

Unmanned Systems Technology XXVI, edited by Hoa G. Nguyen,
Paul L. Muench, Robert Diltz, Proc. of SPIE Vol. 13055, 130550L
 · 0277-786X · doi: 10.1117/12.3013712

The introduction of rapid, minimally invasive, nondestructive testing technologies presents a promising solution for overcoming these limitations. A future aspect of this methodology is the utilization of on-the-edge data processing, which significantly reduces the timeline for obtaining actionable insights into the structural health of infrastructure. The current landscape of drone technology within SHM primarily focuses on leveraging drones equipped with integrated sensors for tasks such as digital image correlation, crack detection, and thermal imaging.[2] These applications, while valuable, do not fully utilize the potential of drones in SHM. By using drones for deploying vibration sensors that can perform on-device data processing, there exists an opportunity to deepen our understanding of structural dynamics under challenging conditions.[3,4]

Current approaches in SHM have seen significant advancements through the integration of UAVs, edge computation, and SHM measurement techniques.[5] UAV-based remote sensing has emerged as a vital approach for bridge condition assessment, offering an efficient, cost-effective, and accessible means to inspect and monitor the structural health of bridges and other infrastructures.[6] The use of UAVs allows for rapid data collection even from areas deemed difficult to access, reducing the need for physical scaffolding and enhancing safety during inspections.

With various approaches to structural health monitoring and numerous technologies currently in use, such as optical and thermal imaging, acoustic emissions, and vibrations, the work by Hassani et al.[7] sets a framework for SHM sensor evaluation. From monitoring various types of civil structures to damage prognosis algorithms, the authors offer a methodology for evaluating cases to assist in the selection of the most suitable sensing technology for a given application. The authors also present a detailed review method for evaluating state-of-the-art sensors and damage detection and prognostics algorithms. Furthermore, Hassani et al.[8] also define optimal sensor placement (OSP) as the placement of sensors that results in the least amount of monitoring cost while meeting predefined performance requirements. This emphasizes the role of aerially deployable wireless sensing systems due to their high mobility while still being a cost-effective solution with similar performance to their wired counterparts.

Edge computation is being increasingly integrated into SHM systems to address the delays in data processing from on-site crews. The use of edge computational devices enables real-time data analysis and decision-making directly at the data acquisition site. This approach reduces the need to travel off-site for real insights into a structure as well as decreasing the need to transmit large amounts of raw data to a centralized server. Despite its potential, the implementation of edge computing in SHM comes with challenges, including the development of robust algorithms capable of operating under the conditions of constrained computational resources available.[9]

Vibration-based SHM techniques, utilizing MEMS accelerometers, have become a standard for detecting anomalies and assessing the structural integrity of buildings, bridges, and other critical infrastructure. These techniques rely on analyzing the vibrational characteristics of structures to identify potential damage or changes in structural behavior over time. Advances in MEMS technology have improved the sensitivity and reliability of these sensors, making them more useful for SHM applications.[10]

This work presents a methodology for integrating edge computation directly onto drone-deployable sensor packages designed for the vibration monitoring of civil structure.[11] In this preliminary work, the algorithms specifically designed to detect and track changes in the first mode of a vibrating structure are deployed to an edge computing device.[12] By facilitating real-time analysis of vibration data directly on the device, the approach minimizes the delay or potential for lost data caused by data transmission. These advancements can improve emergency personnel's response to infrastructure damage. Drones can be leveraged to access hazardous or hard-to-reach areas. The proposed sensor package, designed to autonomously monitor a structure's first natural frequency, can function independently and alert first responders or technicians to any alterations in the structure's vibrational properties.

The contributions of this work are twofold. First, a previously proposed algorithm[12] for frequency-based damage detection is expanded to include an automated methodology to find the frequency associated with the first mode of the structure. Second, the appropriateness of the edge computing algorithm is demonstrated by deploying the algorithm to the same ARM Cortex-M7 microcontroller that is used on the open-source drone-deployable sensing node designed for SHM. The sensor design and code are available on GitHub.[13]

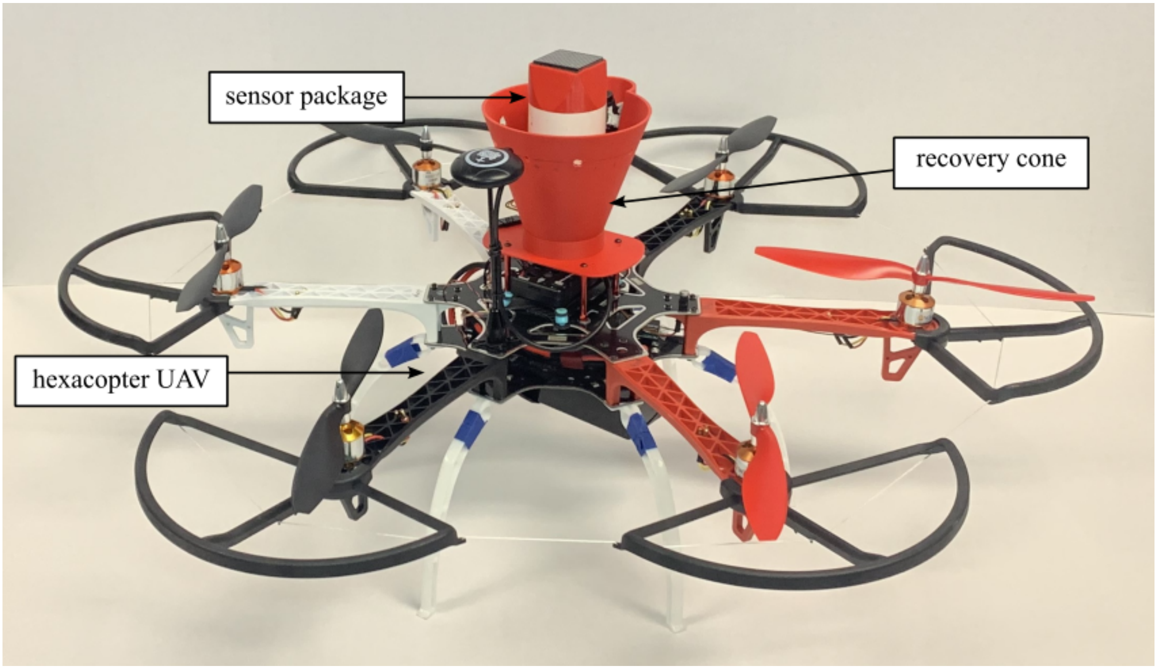

Figure 1. Labeled components for a typical drone sensor package deployment.

2. METHODOLOGY

This section describes the methodology employed to examine and validate the effectiveness of the sensor package for edge computational SHM. The methodology is divided into three subsections, each detailing a component of the study: the hardware specifics of the sensor package, the algorithm installed on the sensor package, and an experimental validation.

2.1 Sensor Package Hardware

The sensor package discussed in this paper is designed for the vibration monitoring of civil structures. For example, Satme et al. reported on a case study that used the sensor packages to perform experimental modal analysis on a pedestrian bridge in use.[11] For rapid assessment, the sensor packages can be deployed by hand or by leveraging UAVs.[3] Figure 1 illustrates a standard setup for deploying the sensor using a drone, while Figure 2 details the step-by-step process of UAV deployment. These sensor packages are considered "smart" in that they can be enhanced with onboard signal conditioning. Satme et al. developed a long short-term memory (LSTM) error-compensating network for the sensor package that demonstrated a 9.3% increase in signal-to-noise ratio (SNR_{dB}) of the collected signals, with the most improvement found at lower frequencies.[14] Hardware and software designs for the sensor packages[13] and deployment systems[15] are open-sourced and freely available.

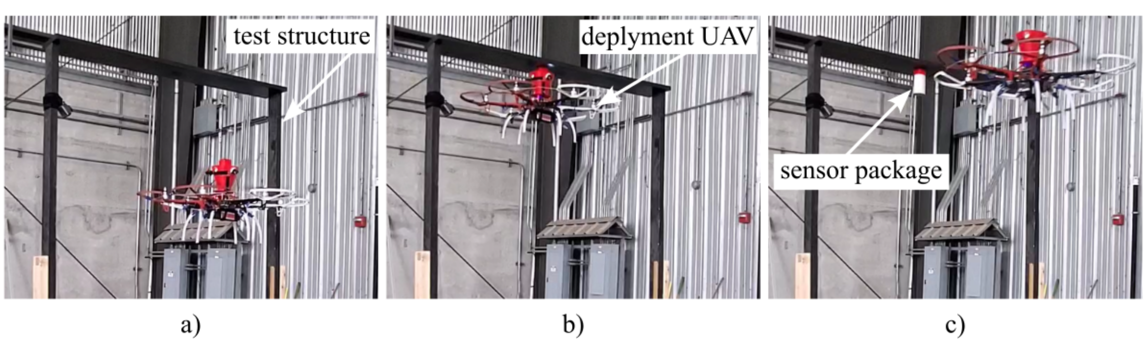

Figure 2. Deployment steps for the drone-deployable sensor package showing a) delivery, b) deployment, and c) departure.

The sensor package is shown in Figure 3 and is designed as an embedded system-based device for long-term data logging of structural vibrations. The core of this device is an ARM-Cortex-M7 processor housed in a Teensy 4.0 microcontroller. The sensor package receives power from a 1500 mAh 2-cell lithium polymer battery, with a power conditioning and regulating system, ensuring stable power distribution to the various systems on board.

The functionality of the sensor package is provided by a Murata SCA 3300-d01 MEMS accelerometer, which communicates via the Serial Peripheral Interface (SPI) protocol. For deployments with minimal intrusion, the sensor package incorporates an EPM V3R5C NicaDrone electropermanent magnet. This magnetic setup requires a 5W pulse to change states, which is typically done twice during deployment for low power consumption. IO commands and data transmission are handled by a Nordic Semiconductors NRF24L01+ module, which operates at 2.4 GHz using the ShockBurst protocol. This network enables multi-link communication with several sensor nodes in addition to wireless sensor activation. A real-time clock and a nonvolatile memory module are also incorporated to extend the device memory to conduct computation in addition to ensuring accurate data logging. The package is protected by a 3D-printed PLA frame and a PVC shell to shield against any environmental conditions during field use. The sensor package is designed with a suitable size and weight for UAV deployment.

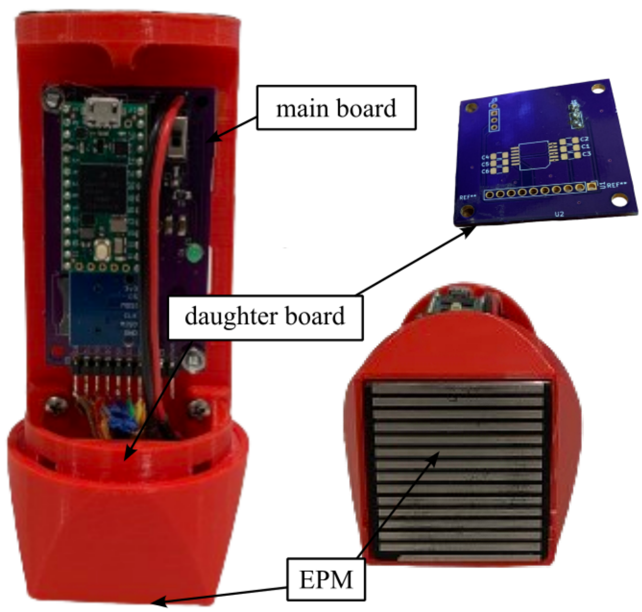

Figure 3. The hardware of the sensor package with key components annotated.

2.2 Sensing Algorithm

The methodology implemented on the sensor package includes an application of edge computing for SHM to leverage real-time data acquisition and processing directly on the device. A flow chart containing the basic run sequence of the methodology is shown in Figure 4. The methodology starts by initializing the libraries for the components of the sensor package as well as setting up variables and functions for later use. Then, data is collected from the z-axis of the accelerometer with recorded timestamps. Once a full vibration test is completed, the collected data and timestamps are then saved to an SD card in a CSV file format.

After data collection, the file containing the accelerometer data is read and each value is used to perform a Fast Fourier Transform (FFT) analysis. The FFT run on the sensor package is a variant of the Cooley-Tukey FFT algorithm.[16] This algorithm is meant to recursively break down a Discrete Fourier Transform (DFT) with a size that is a power of 2. The dataset is split into two sequences of even-indexed and odd-indexed points. The algorithm then computes the DFTs of these two sequences and combines them back into one sequence to produce the DFT of the original dataset.

The full FFT data is written to a new file on the SD card to provide a record of the frequency magnitudes. A peak detection algorithm is run to identify peak frequencies and their magnitudes, which are saved onto another file on the SD card.

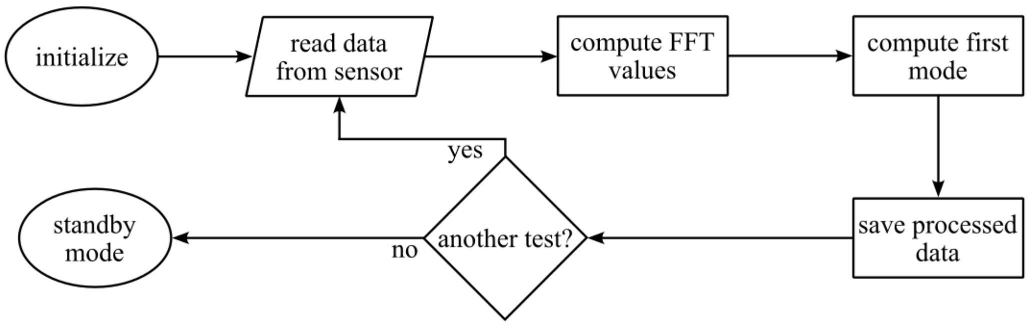

Figure 4. Flowchart indicating the sequence of operations deployed on the sensor package for the automated frequency-tracking algorithm.

2.3 Experimental Validation

To evaluate the sensor package's performance and the effectiveness of its algorithm, an experimental setup was devised involving a square stock beam positioned with a pinned support and a roller support. The experiment, shown in Figure 5, aimed to evaluate the sensor's precision in capturing data under diverse structural scenarios.

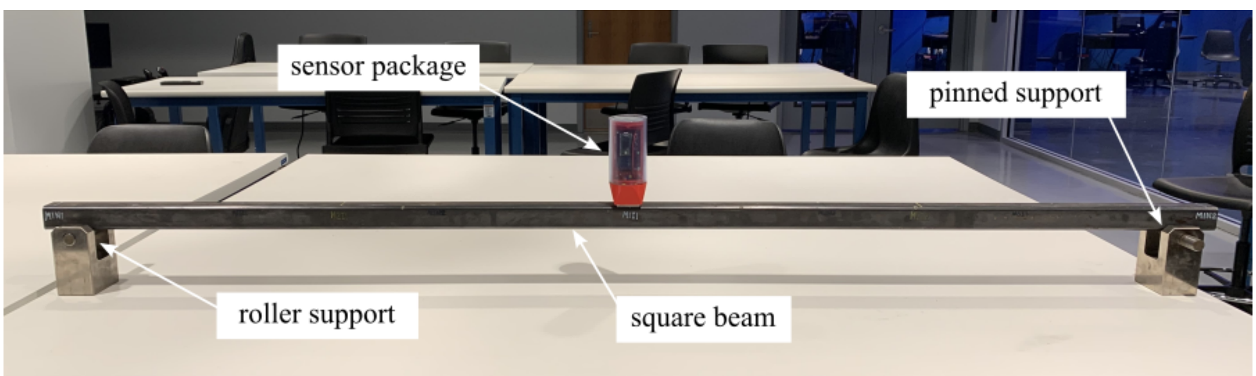

Figure 5. Experimental setup of a beam with pinned and roller boundary conditions on each end with key components annotated.

Positioned at the midpoint of the beam, the sensor package underwent three tests, each performed to emulate a different structural condition. Following the initial placement, the left roller support was sequentially repositioned closer to the beam's center for subsequent tests. A visualization of the repositioning can be seen in Figure 6.

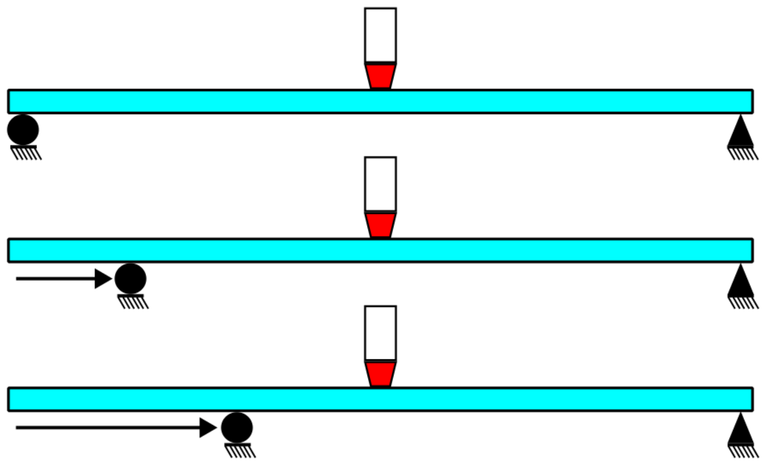

Figure 6. Pined-roller beam test setup showing the shifted roller boundary condition to simulate an altered structural state.

To induce vibrations, a modal impact tool was used, generating an impulse response within the beam, shown in Figure 7. The observed impulse response enabled the evaluation of the sensor package's vibration-sensing ability, FFT analysis, and the rapid identification of the beam's first flexural mode through a frequency domain peak detection algorithm.

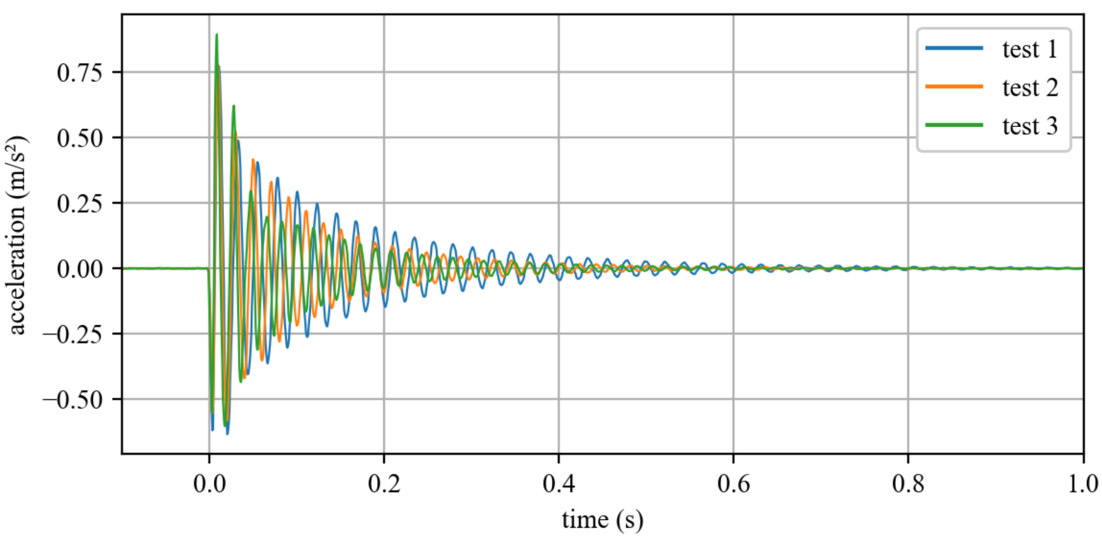

Figure 7. Impulse response of three cases of boundary conditions of a beam with roller supports.

3. RESULTS

This section presents the findings from the analysis of the sensor package's performance, focusing on its algorithm and hardware operation. The sensor's capability to process temporal vibration data and identify the first linear modal frequency of structures is examined. Additionally, an investigation into the hardware's computational resource utilization is also reported.

3.1 Algorithm Performance

The sensor demonstrated efficiency in processing vibration data from the beam, accurately identifying the system's first flexural modal frequencies. The frequencies identified for the first modes in each successive test were

45.1, 51.0, and 56.0, respectively. These variations in modal frequencies can be directly attributed to the adjustments in the beam's structural configuration for each experiment, evidencing the sensor's sensitivity to changes in structural conditions. To validate the sensor's onboard algorithm's accuracy, its FFT output was compared to an off-edge FFT calculated using the Numpy Library in Python[17] in Figure 8. The root mean square error values quantifying the comparison between the onboard and off-edge FFT computations were 0.0032, 0.0028, and 0.0031 in units of the normalized gain for tests 1, 2, and 3 respectively. This indicates a high degree of accuracy across all tests. Note that results here are normalized for the magnitude to account for variations resulting from differences in the Cooley-Tukey FFT algorithms as implemented in the two software packages.

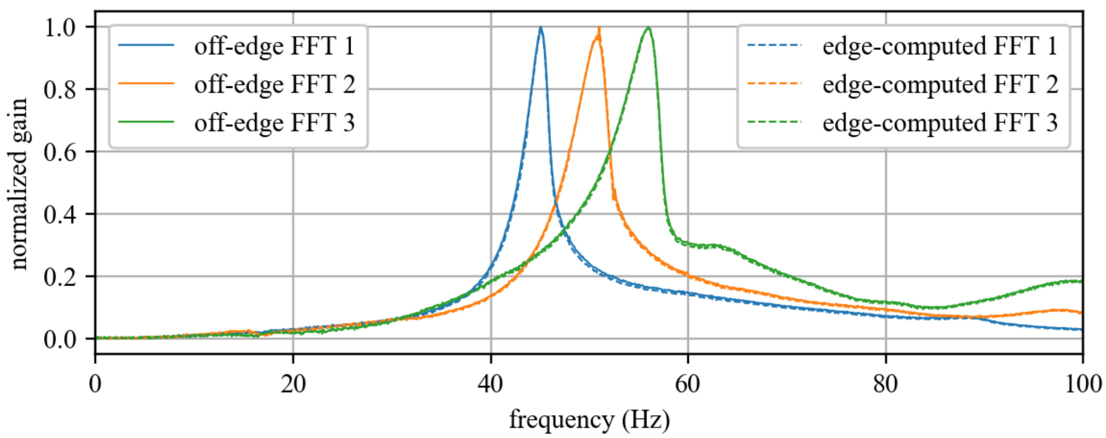

Figure 8. FFT comparison showing the offsite FFT alongside the onboard sensor FFT.

A frequency response function (FRF) analysis was conducted to quantitatively measure the similarity between the onboard FFT and the external FFT computation as shown in Figure 9. Ideally, a flat FRF, hovering around 1 indicates a perfect correlation between the onsite sensor FFT and the offsite analysis. This would suggest that the sensor's algorithm can match the accuracy of external processing. Although there is a slight deviation between the two FFT analyses, the greatest discrepancy near the modes was approximately 9.4%, seen in test 1 at 46 Hz. This deviation is in an acceptable range, as the magnitude was still enough to recognize what frequency the first mode was located at. The comparison showed a remarkable alignment between the two sets of Fourier transforms.

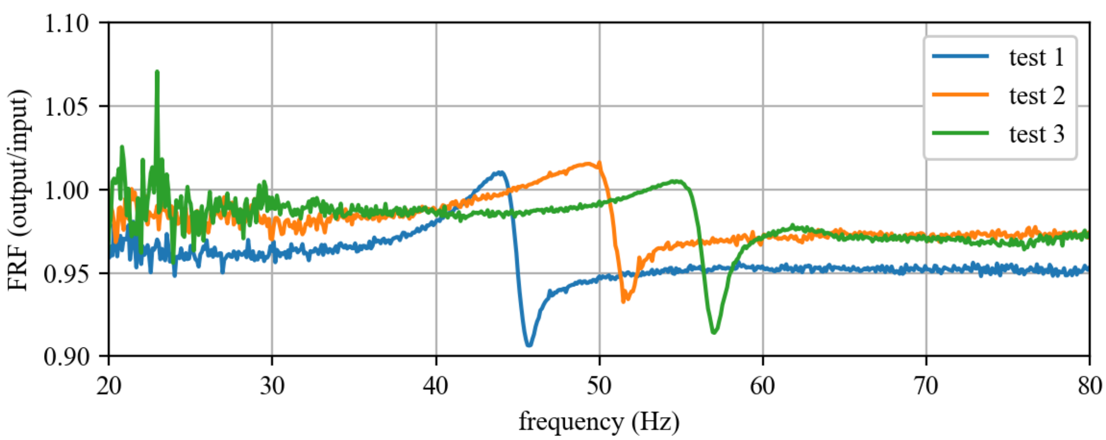

Figure 9. FRF comparison showing the difference between the offsite and onboard FFTs.

3.2 Hardware Performance

The Teensy 4.0 Development Board is an ARM Cortex-M7 microcontroller with a 600MHz primary oscillator. The microcontroller is well suited for on-the-edge computing, featuring 2MB of OCM flash, 1MB of RAM, and 8MB of QSPI flash memory. Its modest demand for operational power at 100mA makes it an ideal platform for sensor packages designed for long-term, continuous SHM.

The overall performance of the hardware driving the Fourier analysis and peak-finding algorithms is examined through GNU gprof,[18] an open-source profiling software from the GNU's Not Unix software collection. The algorithm itself was written using the Arduino toolchain sourced from the Arduino.h header, allowing access to quality-of-life features such as SD card and Serial communication support. This particular workflow is ideal for ensuring a stable algorithmic implementation capable of running continuously without the need for recalibration or similar intervention. Time efficiency is examined by probing profile data correlating to time, cumulative time, and percent of total time.

Table 1. Timing profile describing execution time of processes in the sensor package's algorithm.

process	time (s)	cumulative time (s)	percent of total time (%)
data collection	10.24	10.24	51.07
read sensor	4.43	14.67	22.07
SD card read	0.05	19.54	0.27
SD card write	0.01	19.88	0.07
FFT computation	0.01	19.90	0.07
numeric conversions	0.00	20.01	0.01

A collection of 30 profiling runs was conducted and compiled into a mean aggregate. The configuration of the profiling runs was controlled and consistent, with the sensor package arranged to collect 16384 samples at a 1.6 kHz sampling rate. The data was then processed using the FFT algorithm, which was then followed by a peak-finding algorithm. The sensor package was bench-bound for all runs. This does not affect the quality of the profiling runs. An equal number of floating-point operations occur regardless of data composition or testing environment. The timing profile includes only functions imperative to the algorithm's operation and is shown in Table 1.

4. CONCLUSION

The testing and analyses of the sensor package have successfully demonstrated its capability to effectively process and analyze vibration data on the edge. The sensor's detection of flexural modal frequencies under varying structural conditions proves its adaptability to dynamic environments. The congruence between the sensor's onboard FFT analysis and an offsite FFT calculation further validates the accuracy and reliability of the embedded algorithm. The findings from this study underscore the algorithm's ability to be deployed on resource-constrained devices for SHM. Future work will focus on refining the sensor algorithm to read multiple modal frequencies from a structure as well as learning modal frequencies to recognize discrepancies. This study makes contributions in the form of a methodology of edge computing algorithms.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation through grant numbers 2152896, 2237696, and 2344357 with additional support from the Air Force Office of Scientific Research (AFOSR) through award no. FA9550-21-1-0083. Any opinions, findings, conclusions, or recommendations expressed in the material are those of the authors and do not necessarily reflect the views of the National Science Foundation, or the United States Air Force.

REFERENCES

[1] Niusha Navabian, S. B. and Matulich, J., "Development of wireless smart sensor network for vibration-based structural health monitoring of civil structures," *Structure and Infrastructure Engineering* **18**(3), 345–361 (2022).

[2] Sankarasrinivasan, S., Balasubramanian, E., Karthik, K., Chandrasekar, U., and Gupta, R., "Health monitoring of civil structures with integrated uav and image processing system," *Procedia Computer Science* **54**, 508–515 (2015). Eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India.

[3] Carroll, S., Satme, J., Alkharusi, S., Vitzilaios, N., Downey, A., and Rizos, D., "Drone-based vibration monitoring and assessment of structures," *Applied Sciences* **11**(18), 8560 (2021).

[4] Nasimi, R., Moreu, F., and Fricke, G. M., "Sensor equipped uas for non-contact bridge inspections: Field application," *Sensors* **23**(1) (2023).

[5] Laflamme, S., Ubertini, F., Matteo, A. D., Pirrotta, A., Perry, M., Fu, Y., Li, J., Wang, H., Hoang, T., Glisic, B., Bond, L. J., Pereira, M., Shu, Y., Loh, K. J., Wang, Y., Ding, S., Wang, X., Yu, X., Han, B., Goldfeld, Y., Ryu, D., Napolitano, R., Moreu, F., Giardina, G., and Milillo, P., "Roadmap on measurement technologies for next generation structural health monitoring systems," *Measurement Science and Technology* **34**, 093001 (jun 2023).

[6] Feroz, S. and Abu Dabous, S., "Uav-based remote sensing applications for bridge condition assessment," *Remote Sensing* **13**(9) (2021).

[7] Hassani, S. and Dackermann, U., "A systematic review of advanced sensor technologies for non-destructive testing and structural health monitoring," *Sensors* **23**(4) (2023).

[8] Hassani, S. and Dackermann, U., "A systematic review of optimization algorithms for structural health monitoring and optimal sensor placement," *Sensors* **23**(6) (2023).

[9] Martín, C., Garrido, D., Llopis, L., Rubio, B., and Díaz, M., "Facilitating the monitoring and management of structural health in civil infrastructures with an edge/fog/cloud architecture," *Computer Standards & Interfaces* **81**, 103600 (2022).

[10] Sofi, A., Jane Regita, J., Rane, B., and Lau, H. H., "Structural health monitoring using wireless smart sensor network – an overview," *Mechanical Systems and Signal Processing* **163**, 108113 (2022).

[11] Satme, J. N., Yount, R., Smith, J., and Downey, A. R., "Case study for using open-source UAV-deployable wireless sensor nodes for modal-based monitoring of civil infrastructure," in [*Structural Health Monitoring 2023*], *shm2023*, Destech Publications (2023).

[12] Yount, R., Satme, J. N., and Downey, A. R., [*Frequency-Based Damage Detection using Drone-deployable Sensor Package with Edge Computing*], Springer Nature Switzerland (2024).

[13] Satme, J., Yount, R., and Downey, A., "Drone delivered vibration sensor." https://github.com/ARTS-Laboratory/Drone-Delivered-Vibration-Sensor.

[14] Satme, J. N., Coble, D., Huang, H.-T., Downey, A. R. J., and Bakos, J. D., "Non-linear vibration signal compensation technique for UAV-deployable sensor packages with edge computing," in [*Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2023*], Su, Z., Limongelli, M. P., and Glisic, B., eds., SPIE (apr 2023).

[15] Satme, J., Yount, R., and Downey, A., "UAV sensor package delivery system." Aug. GitHub repository, ARTS-Lab. https://github.com/ARTS-Laboratory/UAV-Package-Delivery-System.

[16] Condes, E., "arduinofft: A library for implementing floating point fast fourier transform calculations on arduino." https://github.com/kosme/arduinoFFT (2014). Version 2.0.

[17] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E., "Array programming with NumPy," (2020).

[18] Fenlason, J. and Stallman, R., "Gnu gprof," 1 (jan 1993).