Smartphone-based app development for vibration-based structural inspection

Abstract

Motif, an iOS app that measures bridge vibration frequencies using accelerometers built-in mobile information terminals such as smartphones, and tablets without any other professional devices is developed. The app integrates functions of sensor data acquisition, data analysis and data upload. This paper investigates the use of smartphone sensors for implementing the drive-by bridge monitoring system in practice. Smartphone technology demonstrates its enormous potential in drive-by structural monitoring and is favored over professional devices in terms of logistics, cost, and ease of use.

Keywords

Bridge Monitoring; Smartphone; Sensing; Network

1. Introduction

Transport officials are challenged by the ageing and deteriorating infrastructure. A significant portion of the current infrastructure in use around the world is nearing the end of its useful life and needs to be replaced or structurally repaired. The evaluation of these structures' current health and the timing and scope of any necessary renovation are two important aspects of this asset management. In order to provide proper maintenance, give funding priority, and ensure acceptable levels of transportation safety, there is a need for increased bridge monitoring. The majority of existing bridge structural health monitoring (SHM) methods involve directly instrumenting the structure with sensors and tools to assess things like vibration frequencies. These methods are crucial because they can show when the condition of the bridge is deteriorating. However, because on-site installations are necessary, they can be labor- and money-intensive. Alternative low-cost indirect vibration-based SHM methods that use a vehicle's dynamic reaction to conduct "drive-by" pavement and/or bridge monitoring have recently been proposed.

2. Methodology

2.1 Design of the smartphone app

In this study, a measuring application has been created that interacts with the hardware and operating system elements of smartphones to enable built-in sensor components and to gather, examine, store, and upload data to the cloud. This measurement application is developed using Swift programming language in the XCode integrated development environment. The Swift source code is available at the following URL: https://github.com/HenryCZhang/Motif. This application makes use of built-in sensors to collect data, complete data analysis and obtain parameters needed for the seismic analysis. To access hardware-generated data, *CoreMotion*, an iOS framework that reports motion- and environment-related data from the onboard hardware of iOS devices, including from the accelerometers and gyroscopes is imported. The application also uses *Surge* which is a Swift library that applies the Accelerate framework to provide high-performance functions for matrix math to perform Fast Fourier Transform (FFT) calculation on the sensor data. *SwiftUICharts*, a charts / plotting library for *SwiftUI* is used to display the vibration vs time plots of the accelerometer readings. This real-time processing of data using smartphone computational capability allows a faster diagnosis of the bridge to be generated in the form of seismic intensity graphs.

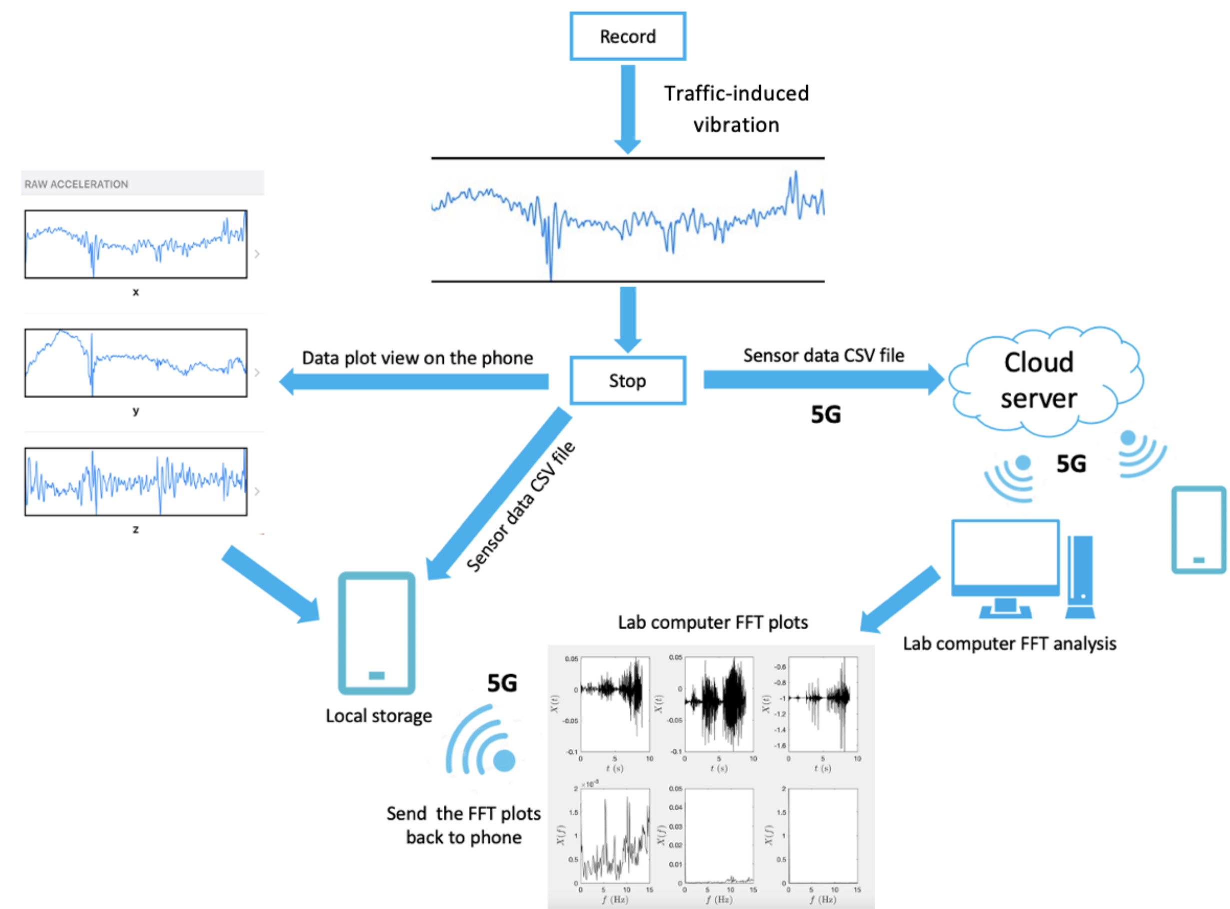

Fig. 1 the proposed smartphone app flow

The general workflow of motif is shown above in Fig. 1. Smartphone will start recording the sensor data according to the sensors and sampling rate setup on the main interface at the beginning after the *start recording button* is pressed. Once the *stop recording button* is pressed, the sensor readings will be saved on the phone and the FFT plots and the vibration vs time plots of the accelerometer readings will be generated in that sensor log detailed view within the app. The imported Swift library, *Surge*, performs the FFT calculation on the sensor data using the Accelerate framework to provide high-performance functions for matrix math. *SwiftUICharts*, the charts / plotting library for *SwiftUI* is tasked with the FFT plot generation. This FFT visualization is an additional feature developed in this study to the original skeleton Swift code. Besides smartphone local data visualization, sensor logs can be uploaded to a database as CSV files and proceed to in-depth analysis on a laboratory computer and the results can be likewise shared to the smartphone for viewing. This data exporting functionality is accomplished by *UIActivityViewController*, a built-in Swift view controller that enables applications to post content to social media sites and send items via email or SMS, and more.

2.2 Functions of the prosed app

Essential interfaces of Motif are shown as Fig. 2, (a) is the main interface, and it includes several functions as follows.

1) The calibration of sampling rate. It is measured in samples per Hz.
2) The selection of built-in sensors including accelerometer, gyroscope, and Magnetic Field. Since this research mainly focuses on acceleration data, the angle and orientation values won’t be recorded by the app. Receiving readings from all three sensors can be accommodated by modifying the Swift code.
3) The real-time collection of acceleration data including X, Y, Z orientation axes, which is shown in Fig. 2(b). The app measures the acceleration data which is used to configure the dynamic properties of the bridge and the vehicle. An accelerometer measures changes in velocity along one axis. The three-axis accelerometer found in all iOS devices provides acceleration values along each of the three axes. The data from the accelerometers is measured in increments of the gravitational acceleration, with 1.0 denoting an acceleration of 9.8 metres per second in the specified direction. Depending on the direction of the acceleration, acceleration numbers might be either positive or negative. Once the *stop recording button* (shown in Fig. 2(b)) is pressed, the sensor log will be saved locally on the phone. The sensor data entries interface is shown in Fig. 2(c). All sensor logs are displayed on the samples page and each log detail can be viewed when it’s clicked on, which is shown in Fig. 2(d).

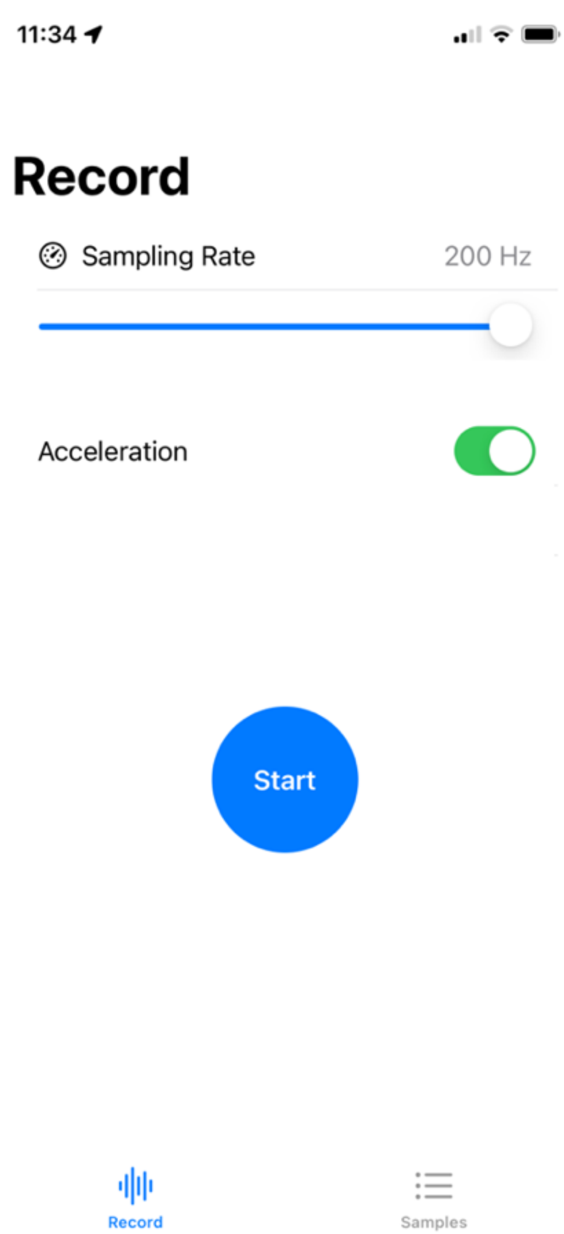

Fig. 2(a) Home page

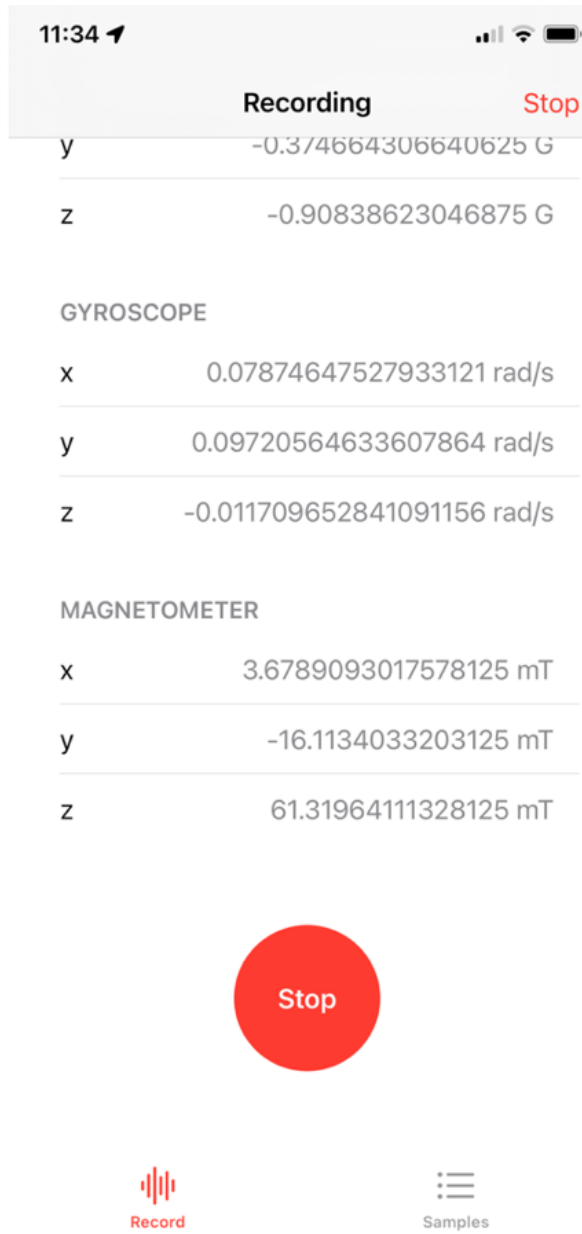

Fig. 2 (b) Data collection

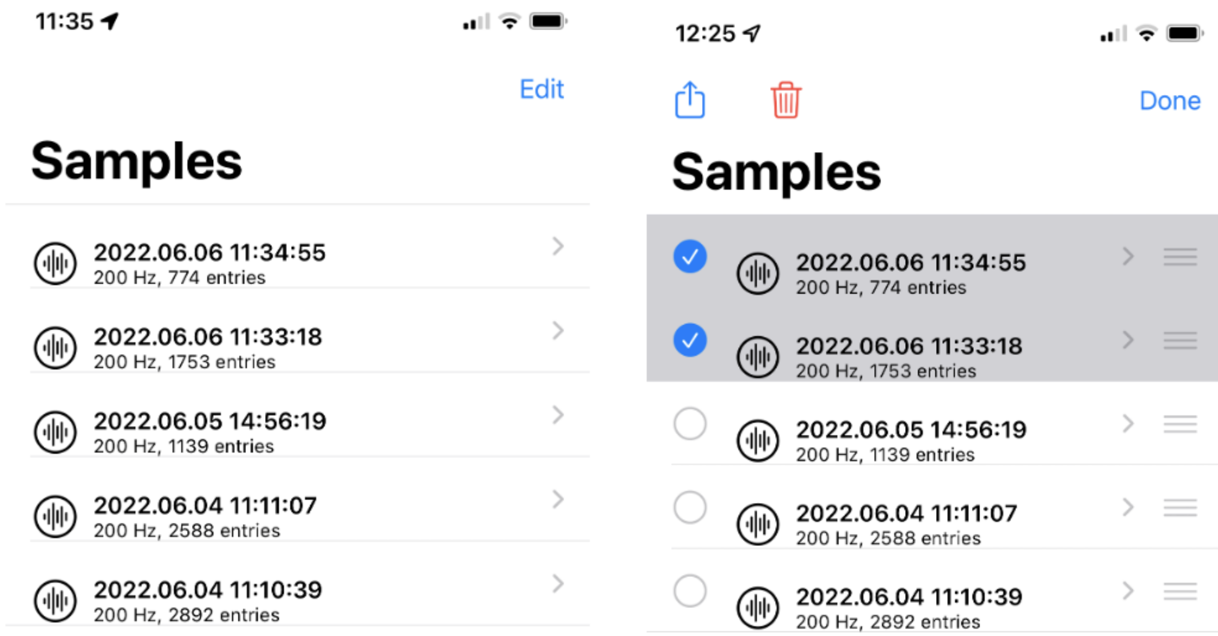

Fig. 2 (c) Data storage

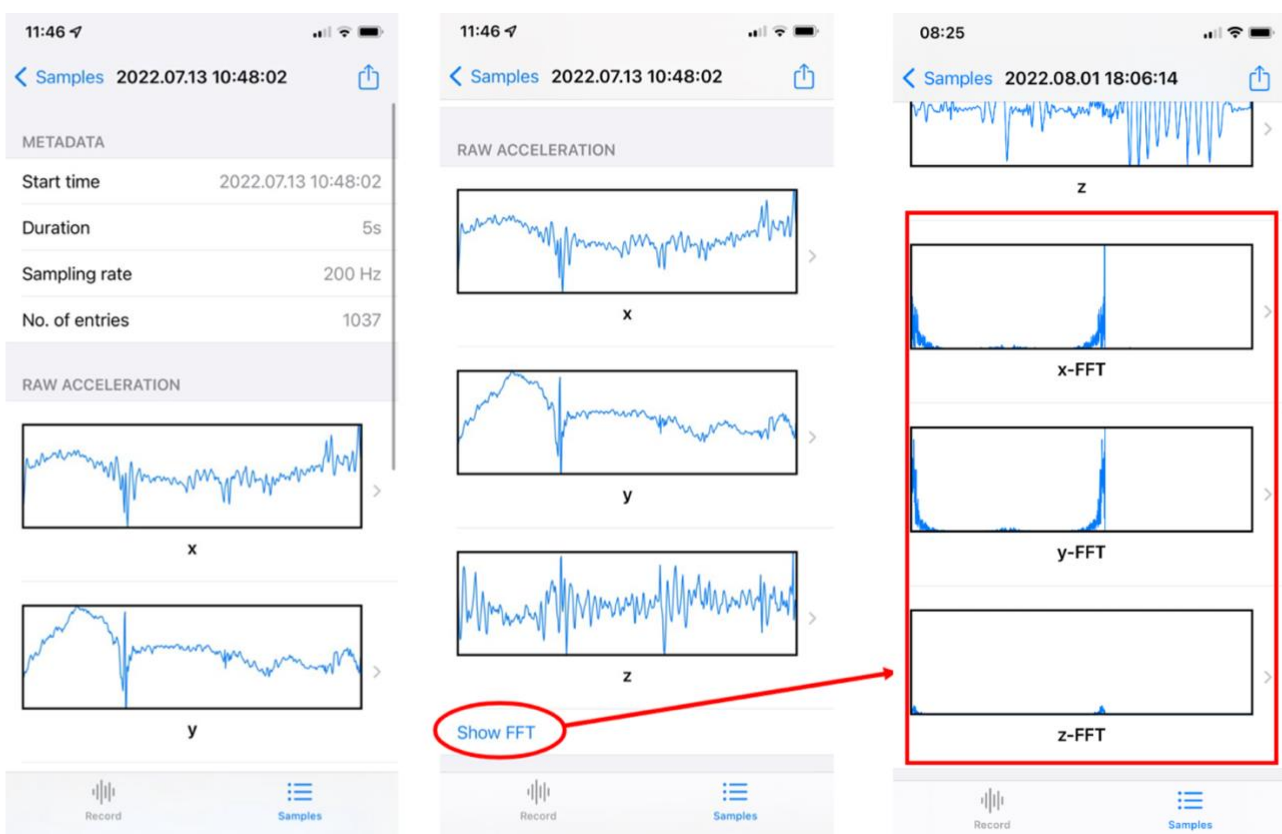

Fig. 2 (d) Data visualization

4) Data analysis, from which the acceleration time-history curve and the FFT of acceleration curve can be obtained, which is shown in Fig. 2(e).

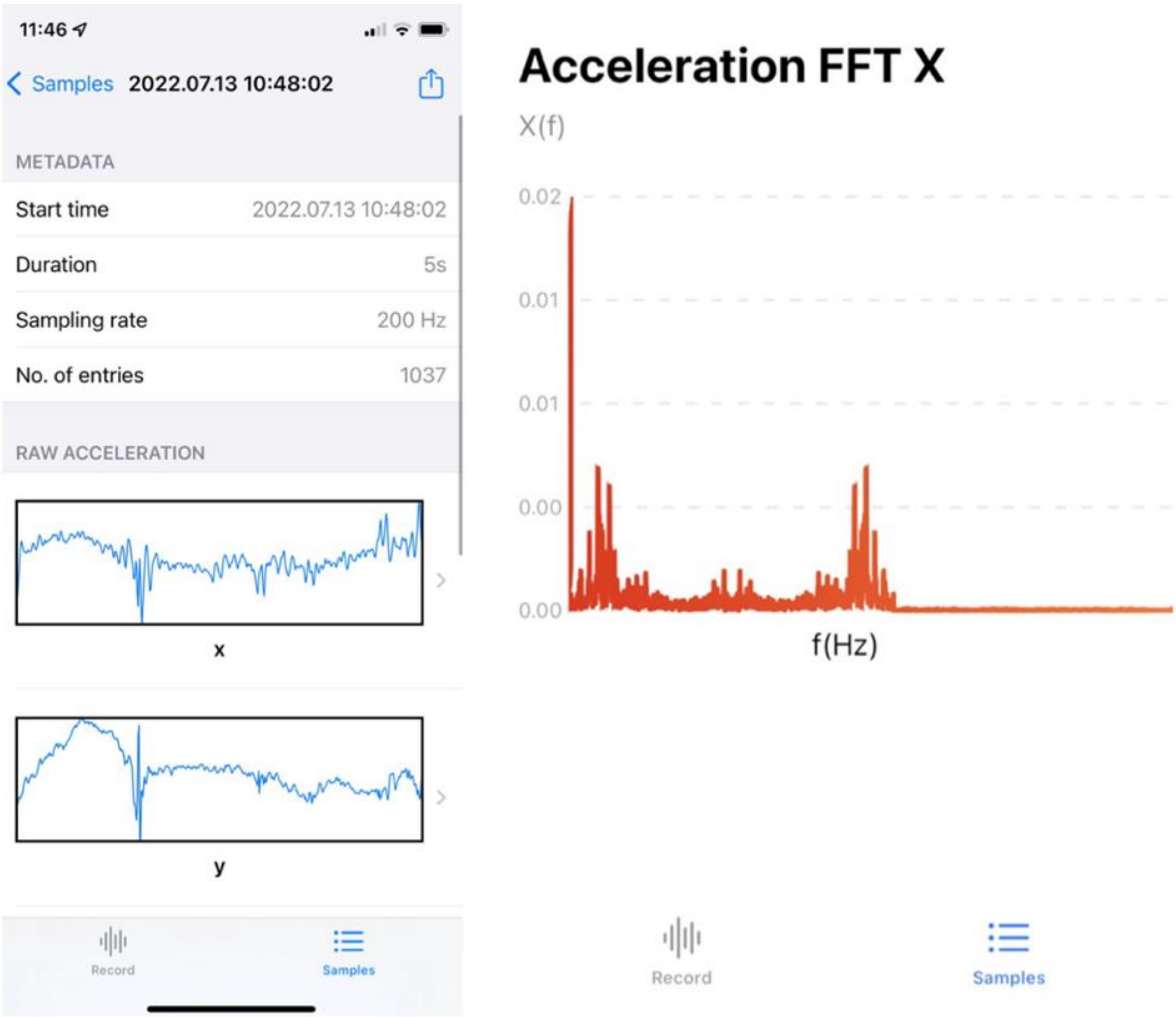

Fig. 2 (e) acceleration time-history curve & acceleration FFT curve

5) The sensor data can be shared to be the cloud or other devices through a wide variety of media sharing platforms using Bluetooth or Wi-Fi and is ready to be shared to the cloud or other devices through a wide variety of media sharing platforms using Wi-Fi or Bluetooth. In this study, Google Drive and MATLAB Drive are used as cloud servers (shown in Fig. 2(f)). Once the data has been transmitted to the cloud, it can be accessed by the laboratory computers and is ready for more complicated analysis. The evaluated results can be also shared to the cloud and later be accessed by smartphones or tablets, which is shown as Fig. 2(g, h).

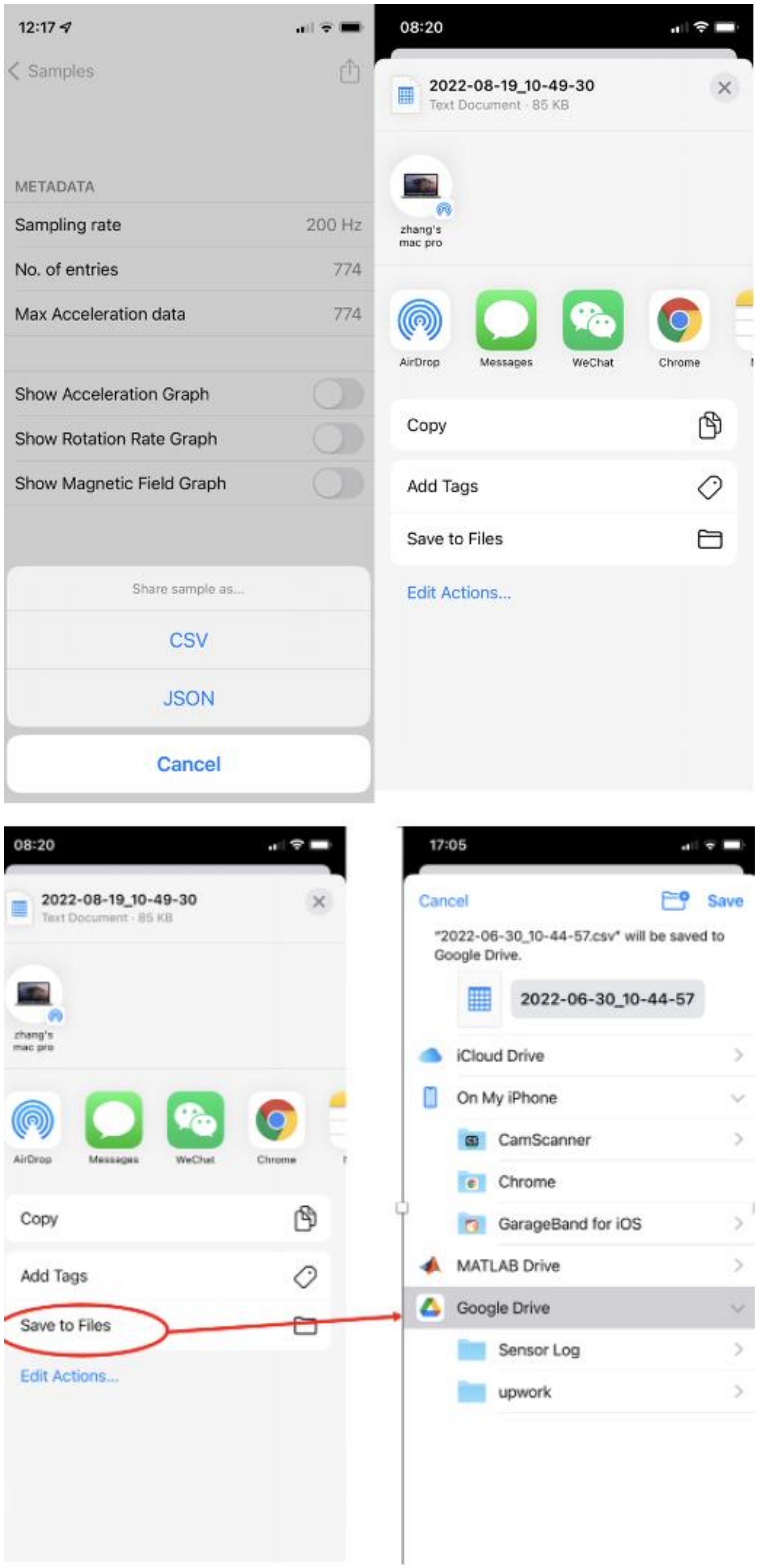

Fig. 2 (f) Data sharing

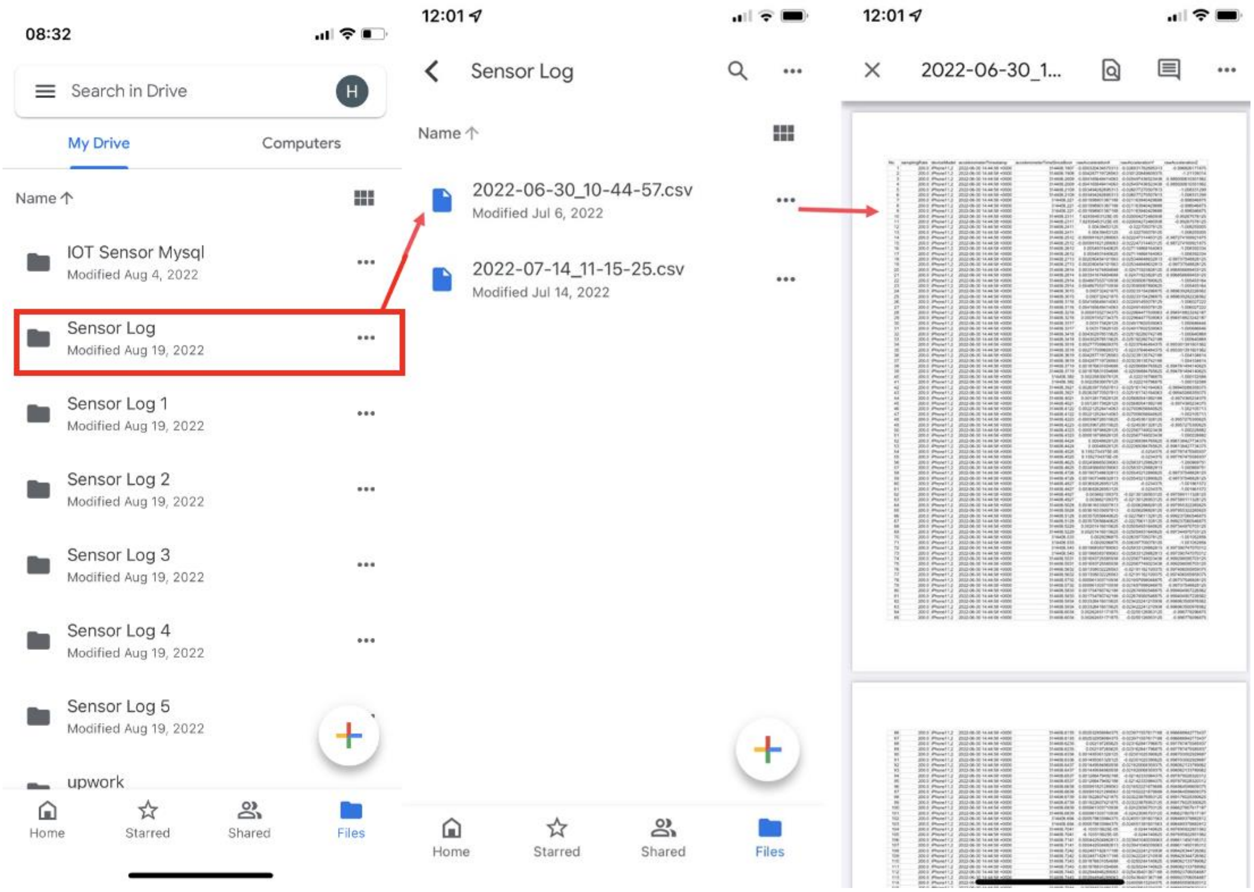

Fig. 2(g) Uploaded acceleration data in Google Drive mobile App

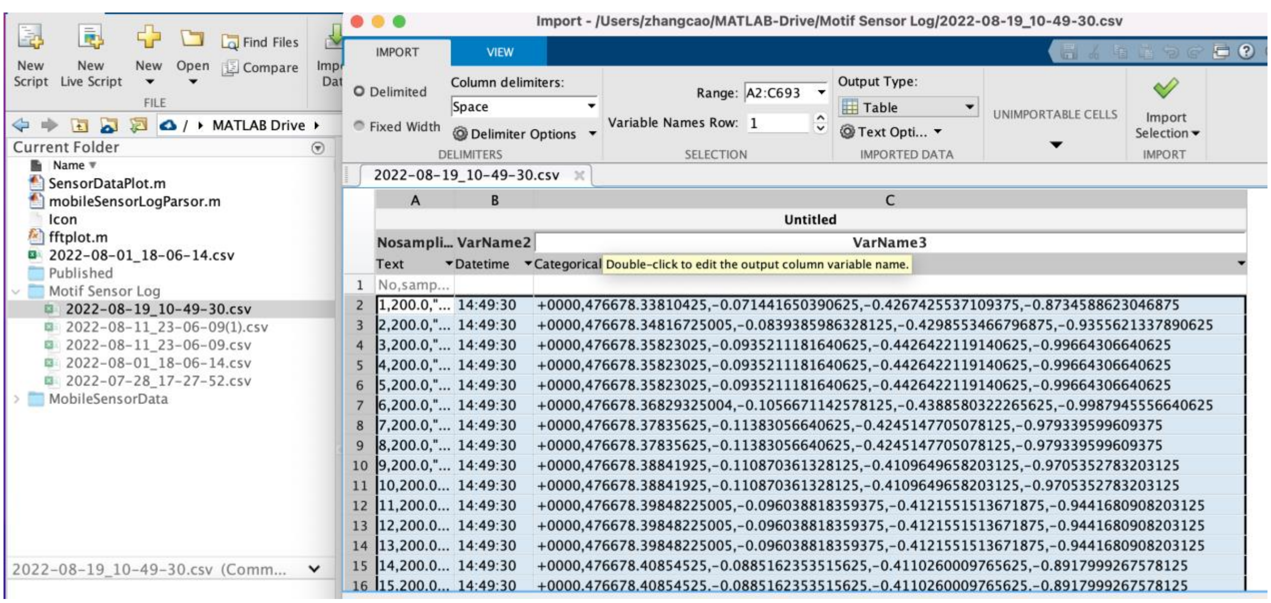

Fig. 2(h) Uploaded acceleration data in MATLAB Drive