

Development of a LabVIEW Application for Measurement and Analysis of Acceleration Signals from an External Reference

Naji Mordi Naji Al-Dosary

Department of Agricultural Engineering, College of Food and Agriculture Sciences, King Saud University, P.O. Box 2460, Riyadh 11451, Saudi Arabia

Abstract: LabVIEW software is a very programmable tool, which can be easily retrofitted and customized to build new aspects of control and database monitoring systems for working with electrical devices. In an instructional environment, the basic LabVIEW software processes provide simulation model of virtual sensors and their associated acceleration output from virtual motion input. This simulation can be used as part of an inclusive perception academic training system for engineering programs. Three virtual accelerometers were used to demonstrate the utility of simulating the acceleration force and vibration principles. A new LabVIEW front panel was developed to represent the sum of the simulation forces acting on each of the 3-axis semi-accelerometers. Thus, raw data strings of the 3-axis semi-accelerometers could be input as simulated real data and analyzed for time and the x , y , z -axis coordinate values. The toolkit of the LabVIEW platform is easy to implement, so engineers and scientists, who are very interested in knowing recent applications of the modern technology and the impact of its development processes, could be able to recognize the LabVIEW toolkits structure optimally, have a comprehensive practice with it and possess an extensive concept of the analysis techniques of this engineering language. Completely, the platform can also be quickly reconfigured to meet any specific laboratory technical requirements. So, this simulation study is underpinning of the accelerated coordinates understanding and has also been exclusive to create a virtual and vital acceleration application via the LabVIEW driver software.

Key words: LabVIEW platform, semi-accelerometer, coordinates, front panel, block diagram, waveform graph.

1. Introduction

Engineers, who are interested in using modern engineering applications to build or improve their engineering prototypes, have benefited greatly from both technology and software advances, and are applying new technical methods, such as the graphical programming (i.e., LabVIEW programming language). In contrast to the conventional methods of developing graphical simulation using long complex programs that are developed in languages (i.e., for its procedural language has required long structure codes), such as C, visual C++, Java or Visual Basic, the LabVIEW (national instruments) graphical interface provides a much simpler straightforward environment. By

designing and running simulations in LabVIEW, engineers can control or operate virtual equipment, set virtual variables appropriate to the project and achieve rapid design progress as effectively as possible [1]. Hinkson et al. [2] discussed one of the LabVIEW applications running in Microsoft Windows, and mentioned that for studying accelerator system control, LabVIEW was most efficient, providing highly functional virtual tools and an efficient system solution capability. Furthermore, current advances in agricultural practices involve the wider use of acceleration instrumentation, which contributes to the growing use of operational monitoring and control systems. Design and development of these new systems can be improved through the use of the LabVIEW platform.

Halvorsen [3, 4] reported the practical benefits of

Corresponding author: Naji Mordi Naji Al-Dosary, Ph.D., research fields: machines systems development and design.

LabVIEW application, modules and toolkits for system design, control and simulation, and noted that it can save an appropriate time and money, as well as can be deployed in various desktop and laptop platforms running windows. It is becoming more useful as the numbers of functions and the levels of control design involved in simulations and modeling increase, especially within the field of large scale technical applications that require more accurate data. Porter and Delinger [5] developed a sensor system using the simple low-cost USB interface of any computer or laptop running LabVIEW to communicate with any piezoresistive microcantilever systems.

For vibration force measurements on citrus tree canopy, Udumala Savary et al. [6] developed a sensing application to reveal the data associated with the tree canopy vibration that was required to detach citrus fruits. Depending on the accelerometer capabilities, the accelerometers' raw data (acceleration and the coordinates of x , y and z -axes) were compiled into a text file by using MathScript through the LabVIEW palettes, and the text file was then utilized to calculate the canopy shaker frequencies and vibration forces.

Figueiredo et al. [7] discussed a low-cost remote SMS system for a remote monitoring and control system that is adaptable and can be utilized in numerous practical applications. The monitoring system was comprised of LabVIEW software, a computer and a GSM modem which was accessible via an external cell phone, and by using the LabVIEW toolkits, the instrumentation applications could be remotely monitored and controlled.

For protection of a mechanical system designed for a chop saw machine, the LabVIEW program was utilized to estimate the vibration signals of the mechanical system. Subsequently, actual signals were obtained by using two accelerometer instruments connected to a data acquisition (DAQ) module and laptop. Vibration data were written to a Microsoft

Excel file and saved by using the LabVIEW palettes. Mainly, when compared to other computer languages, the LabVIEW program was simple, visual, accurate and adequate for current instrumentation systems. Desai et al. [8] emphasized that the main advantage of utilizing the LabVIEW toolkit was its flexibility.

In mechanical production processes, most manufacturing industries need to monitor their production tools very closely and carefully [9]. The industries need to design an accurate detection and control system that operate in real time (i.e., immediate status and reduced monitoring intervals) before their tools get to the point of breaking down. A practical new monitoring system was developed based on the available sensors which were used to send signals from the tools and on the machine conditions to a DAQ device, and then monitored by toolkits in the LabVIEW software. By utilizing the graphical program (LabVIEW) and the designed control system together, an emergency alarm would be activated before the breakdown point, as well as the LabVIEW system obtained data from the controller system in real time, which was a significant improvement over other available technology. In addition to the construction of the monitoring system that was necessary to control excess vibration of the industry's machines, the LabVIEW program was also applied to determine the actual vibration levels of the machines. Measured signals from the vibration sensors were acquired through wired connections to the vibration sensors, a DAQ device and a PC. By using the fast Fourier transform (FFT) method, vibration data were collected and stored in a text file which was read by the LabVIEW software and analyzed through the designed palettes in LabVIEW.

In general, a LabVIEW program was initially deployed for the vibration monitoring system in order to determine the accuracy of the mechanical machines maintenance. When compared to other conventional instruments, the advantages of LabVIEW include: flexibility of use, the capability of monitoring several

machines' vibration and recording their signals quickly, an online monitoring capability (remote control and operation), adaptability, a visual interface, simple customization, not complex, low cost and not requiring expensive accompaniment instruments to run it (save money). Shah et al. [10] and Lita et al. [11] concluded that the development of typical performance and monitoring system was accessible by the application of the LabVIEW graphical programming environment.

Currently, automatic accelerometer systems have permeated many precision engineering applications. Successful implementation and operation of accelerometer devices is based on accurate analysis of the acquired data. LabVIEW virtual instrumentation engineering software and associated platform workbench and palettes are an efficient method of instructing students on estimating the optimum operating variables of the data that affect any application performance. The foundation of the software is accessed by opening the LabVIEW workbench which provides basic information about functioning of the LabVIEW acceleration palettes and platforms. It has many different palettes that provide for efficient development of code applications, such as the underlying principles for acceleration forces and vibration. Accordingly, the objective of this study was to recognize the LabVIEW toolkits, develop a write text file as a reality of 3-axis semi-accelerometer data (time and three dimensions, the orthogonal x , y and z -axis), and then read data from the external text file and perform the analysis (full data columns and graphs).

2. Simulation Description

Essentially, the analysis of accelerometer simulation data has two assignment processes. The first, called "write to text file", collects a text file of the output quantities from the design palettes, which include the three dimensions data from the semi-accelerometers. The second process is called

"read from text file", in which output data can be viewed as the tri-axial accelerometer signal. The two simulation systems that were examined utilize the basic principle of simulation of acceleration measurement, accessed via a large and varied library of reconfigurable virtual accelerometer palettes. The best example to describe the tri-axial accelerometer signal can be shown during the basic LabVIEW 2016 software, version 16.0 [12], which are used to obtain all plots of the acceleration system response. The complex blocks diagrams that are symbolized in this object represent an application of the acceleration system, such as a practical application in the field of agricultural and biosystems engineering and communications development. In the absence of actual laboratories, these programs are easy to understand and help the investigator to build a virtual laboratory, which can be applied into scientific applications. As a final point, these accelerometer applications and simulations can lead to real instrument development.

3. Accelerometers' Simulation Contributions

Initially, a file dialog window and open/create new file or replace existing file icons are chosen within the LabVIEW program. For the imaginary acceleration signals, individual 3-axial semi-accelerometers would be quite chosen and operated simultaneously. The acquired x , y and z -axis coordinates of the semi-accelerometers would be shown as a visible text file using the LabVIEW 2016 software architectures [12]. In addition, if more calculations are required, the acceleration forces could be calculated from the known semi-accelerometers sense (3-axial of acceleration values as shown in the existed text-file) and some standard equations of the resultant acceleration and Newton's Second Law of Motion [3, 6]. Thus, diagrams of the semi-accelerometer analogue signal could be developed in the reconfigurable students' lab and then utilized for the oscillations in practical applications, such as live

functional activities studies.

4. Simulation Test

4.1 Flow Diagram of Physical Data Determination for the Semi-accelerometers Simulation

For this demonstration of the software’s potential, the file dialog window and open/create new file or replace existing file icons were chosen, the number of acquisitions was selected as 10, and the number of points/acquisition was chosen as 3, all through time duration set at 500 milliseconds (ms). Next, into a while loop, the three semi-accelerometers were hooked together by using some string and array palettes via several mutual wire connectors, and the *x*, *y*, *z*-axis coordinates of the three accelerometers with time duration were displayed in the front panel and named “data strings of all accelerometers”, “data strings of the three accelerometers in three rows for one time” and “data strings of the three accelerometers on one row for one time”, respectively (Figs. 1 and 2).

In addition, the data strings of the three accelerometers (*x*, *y*, *z*-axis coordinate values) were shown as diverse channels in along row for the first

simultaneous time-period of each accelerometer. The first three values of the data corresponded to the coordinate values of the first accelerometer, the second three values in the row indicated the *x*, *y*, *z*-axis coordinate values of the second accelerometer, and the third three values in the same row were the *x*, *y*, *z*-axis coordinate values of the third accelerometer. In the same manner, the next rows (up to the 10 acquisitions) revealed the *x*, *y*, *z* coordinate values of the three accelerometers. Finally, all three accelerometer data were shown as 10 time periods (10 rows), where the first three columns were the *x*, *y*, *z*-axis coordinate values of the first accelerometer, the second three columns were the three axes coordinate values of the second accelerometer, and the last three columns show the *x*, *y*, *z*-axis coordinate values of the third accelerometer.

For greater detail, *x*, *y*, *z*-axis coordinate values of each accelerometer were displayed individually by arranging new array and string palettes. The first accelerometer data were shown in an indicator called “data string of accelerometer 1”, the second as “data string of accelerometer 2”, and the third as “data string of accelerometer 3” (Fig. 2).

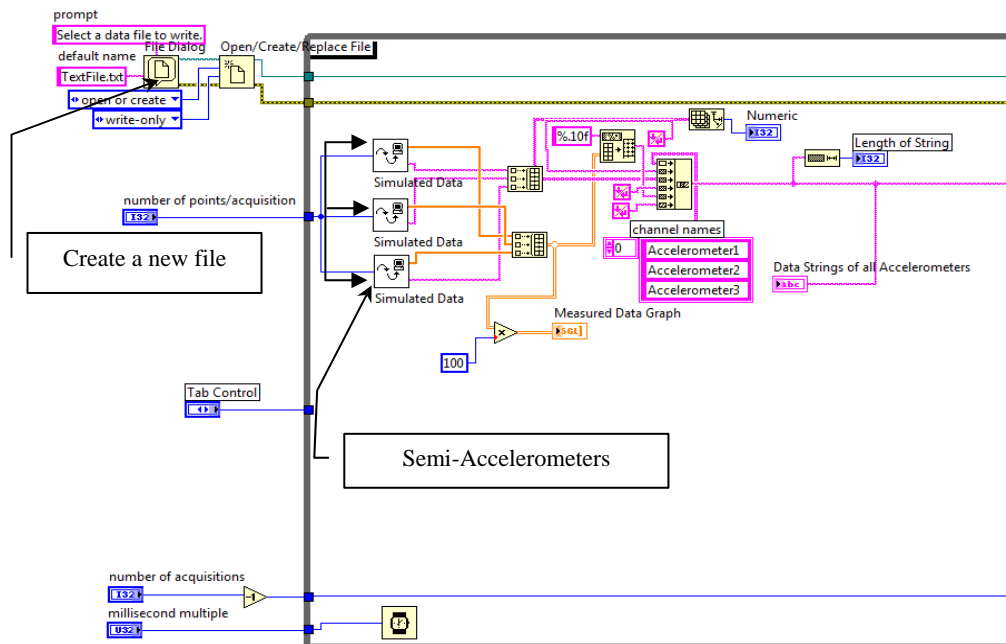


Fig. 1 Three accelerometers linked as a simulated data icon in the LabVIEW block diagram.

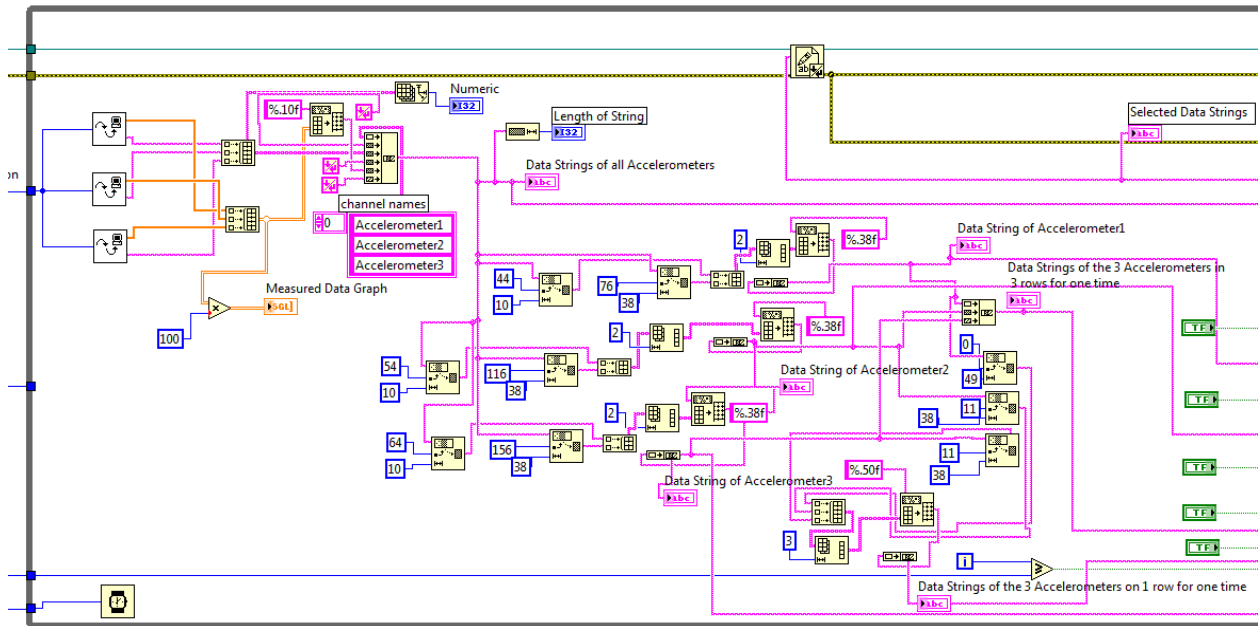


Fig. 2 The strings and arrays of the three accelerometers in the LabVIEW block diagram.

Furthermore, several comparison and Boolean palettes were utilized in the LabVIEW block diagram to arrange the accelerometers data stream, so they were shown individually in an external text file, and the close file and simple error palettes appeared on the end of the block diagram for any file closing and error checking dialogs, as shown in Fig. 3.

Finally, the overall LabVIEW block diagram of all the formerly discussed dialogs is shown in Fig. 4. In addition, the raw x , y , z -axis coordinates of the three accelerometers would be stored in an external notepad text file that can be easily called, such as, "Text_File_Acc_Dec_2016.txt", and the test result stored up as shown in Fig. 5. The "Tab control" could be used in the LabVIEW block diagram and the front panel in order to have less complicated data handling control in a smaller sized tab. Also, x , y , z -axis coordinates could be read in the LabVIEW front panel of each accelerometer waveform graph as shown in Fig. 6 and the supplementary figures.

4.2 Reading the Physical Data for Determination of the Semi-accelerometers Simulation

The first step, after accumulating the x , y , z -axis coordinate values of the three accelerometers in an

external text file, is to open the file and examine the individual accelerometer data (time and x , y and z coordinates axes). This is accomplished via the file dialog window, which contains: open/create/replace file, get file size, read from text file, close file and simple error palettes. The appropriate function palette is selected along with the file to be read, as shown in Fig. 7. At the same time, different types of individual string and array palettes can be linked together by using mutual wire connectors (Fig. 8), or by using the LabVIEW block diagram for the complete set of dialogs (Fig. 9). Either of these methods results in the display of the x , y , z -axis coordinates of the three accelerometers with associated time columns in the front panel list "data strings of all 3 accelerometers" (Fig. 10). The "Tab control" is used in the LabVIEW block diagram on the front panel to simplify data handling control. The test result of the streams of X value of the three accelerometers over time stamp (data by X channel vs. time) can also be shown in the LabVIEW front panel as a waveform graph for each individual accelerometer (Figs. 11-13). Also, as shown in Fig. 14, X values of the three accelerometers over time stamp (data by point value X vs. time) can be displayed simultaneously. Finally, all values from the

Development of a LabVIEW Application for Measurement and Analysis of Acceleration Signals from an External Reference

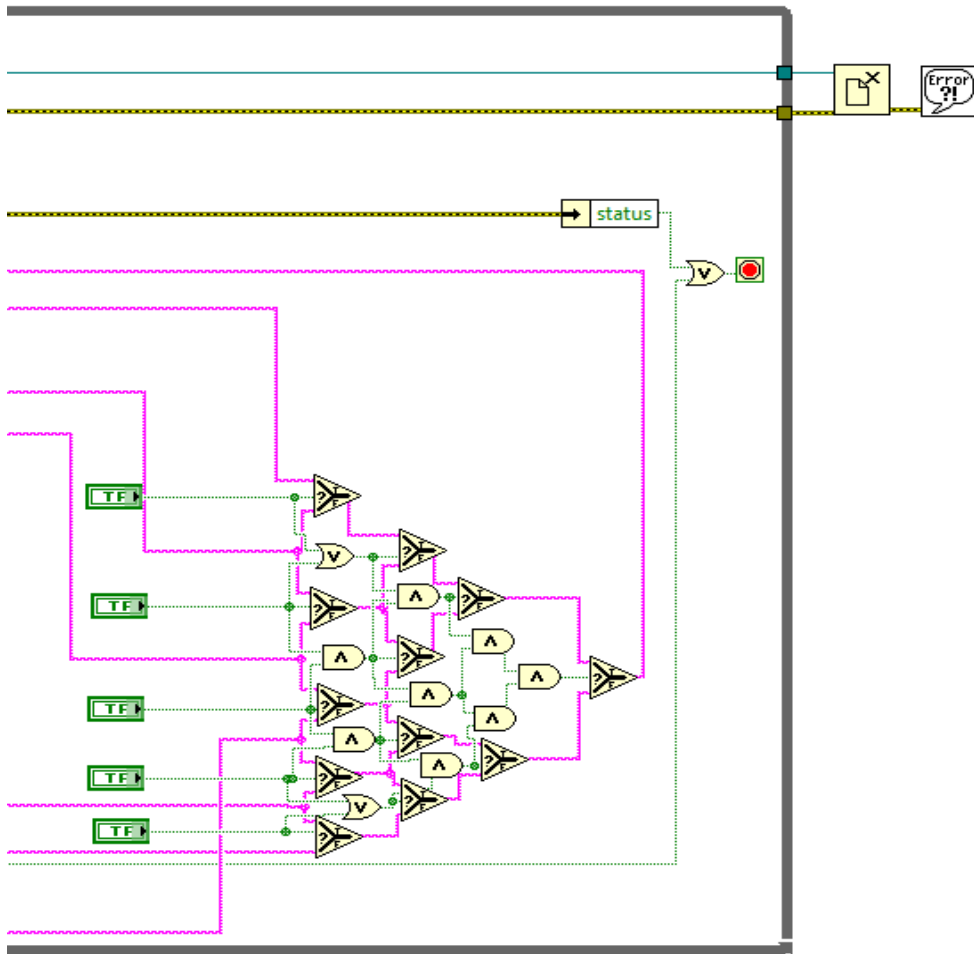


Fig. 3 Arranging the comparison and Boolean palettes for accelerometers data stream in the LabVIEW block diagram.

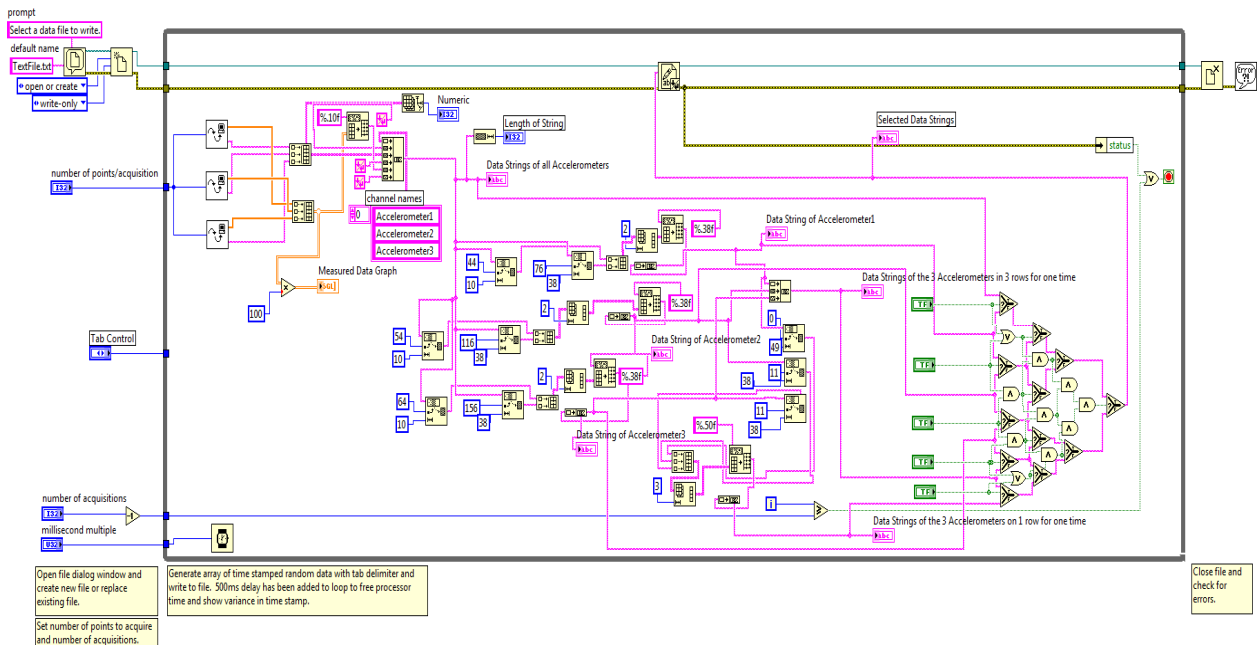


Fig. 4 Main dialog for writing a text file in the LabVIEW block diagram.

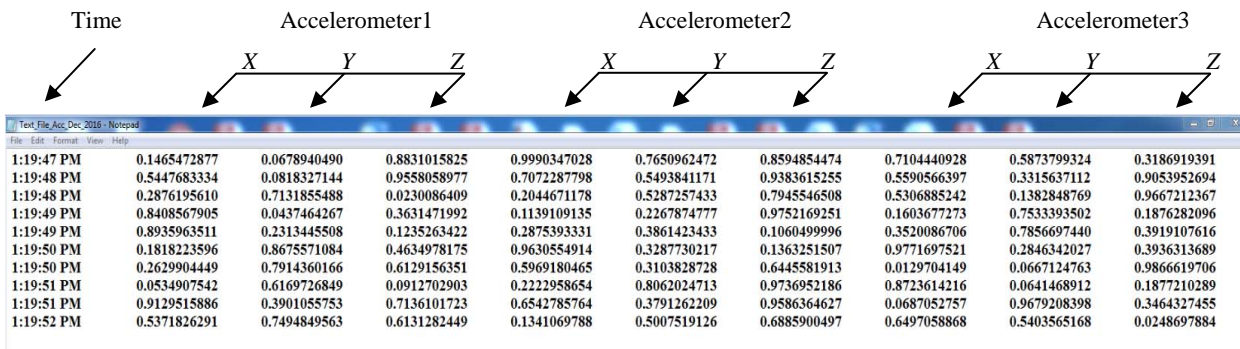


Fig. 5 The x, y, z-axis coordinate values of the three semi-accelerometers.

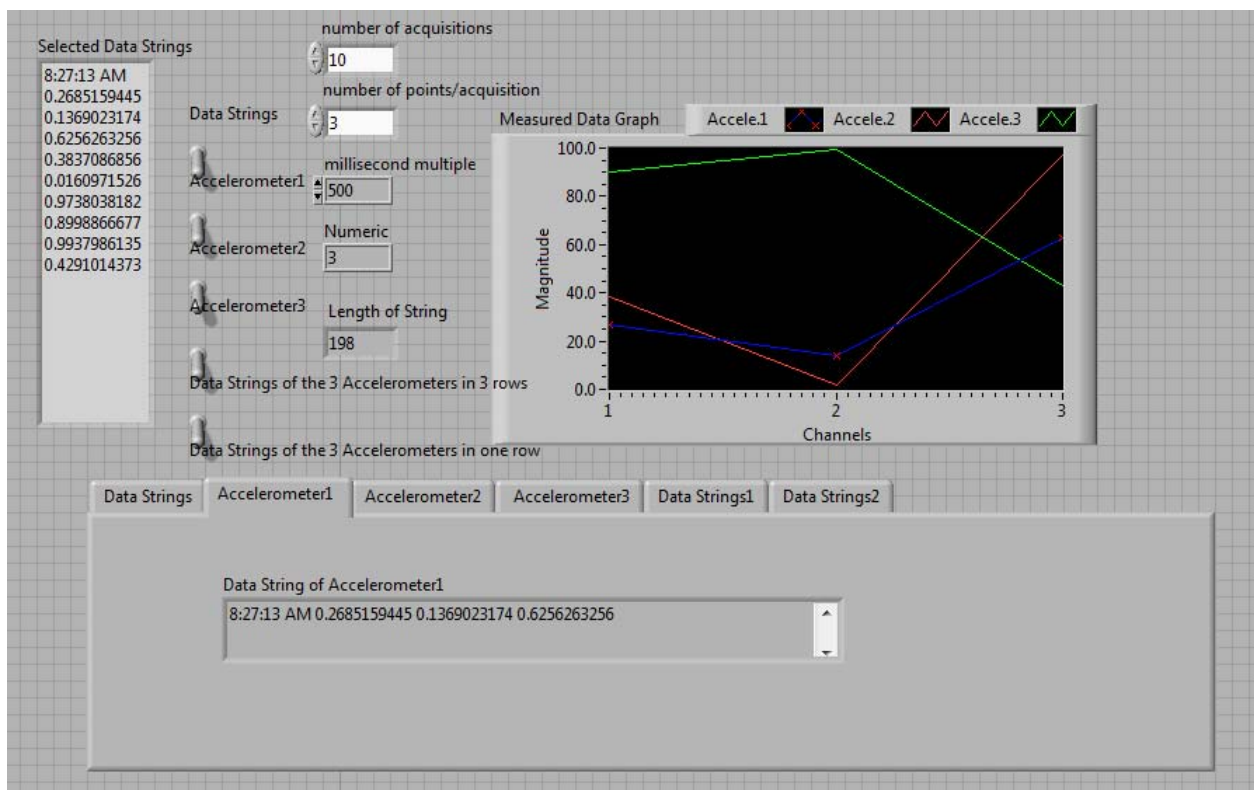


Fig. 6 The LabVIEW front panel of the writing a text file dialog.

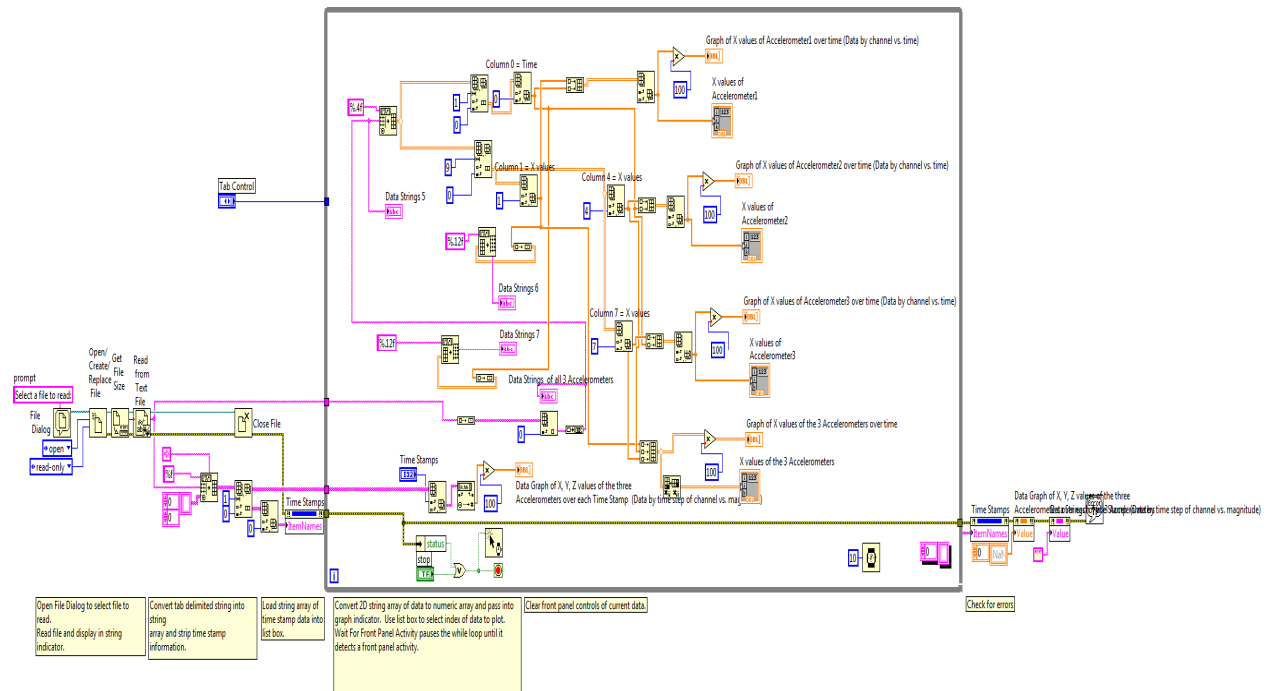


Fig. 9 Dialog of reading a text file in the LabVIEW block diagram.

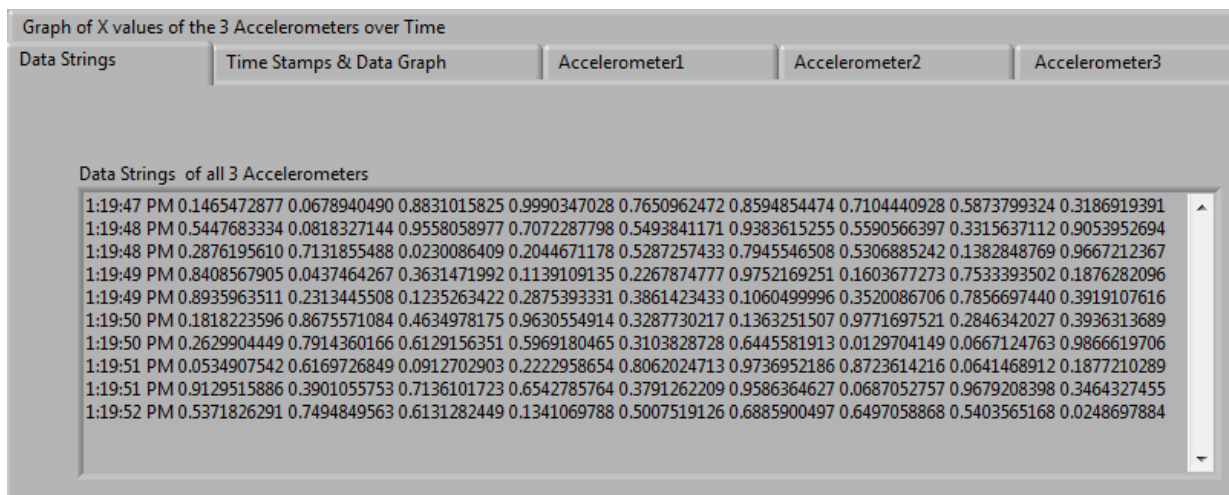


Fig. 10 Stream of the three accelerometers data are presented in the LabVIEW front panel.

Development of a LabVIEW Application for Measurement and Analysis of Acceleration Signals from an External Reference

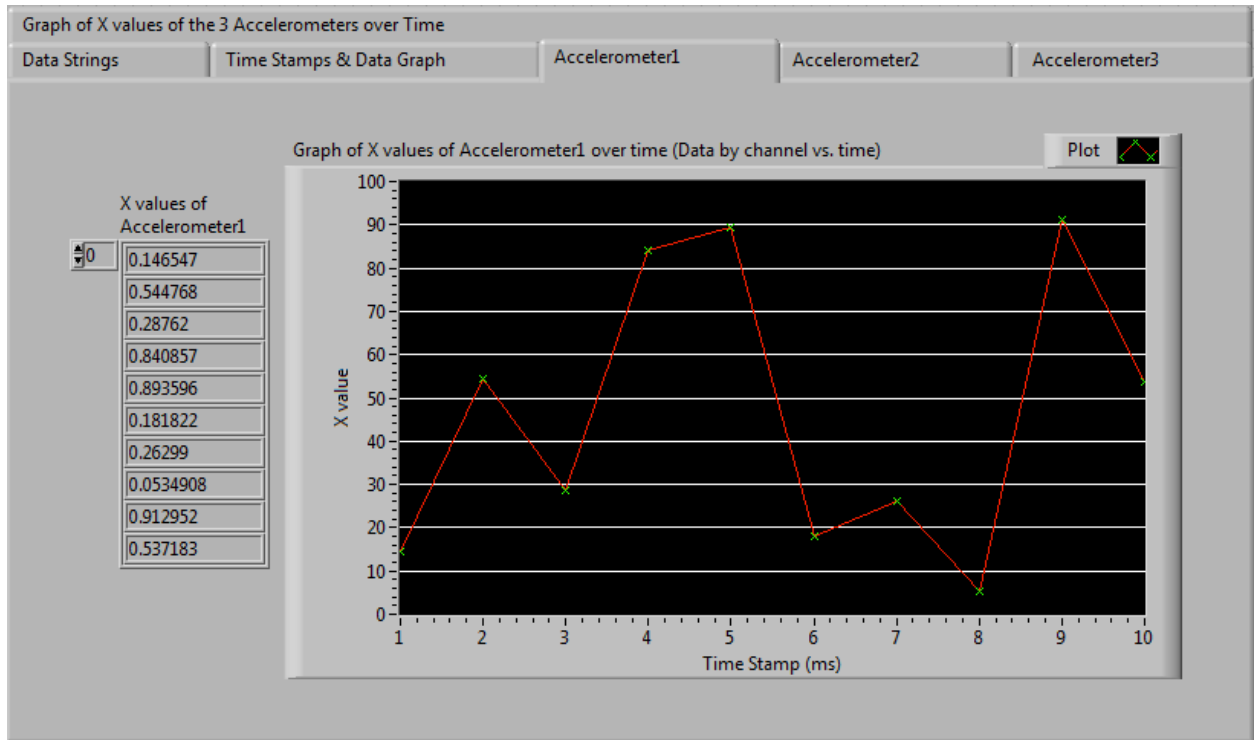


Fig. 11 Graph of X values of the first accelerometer over time presented in the LabVIEW front panel.

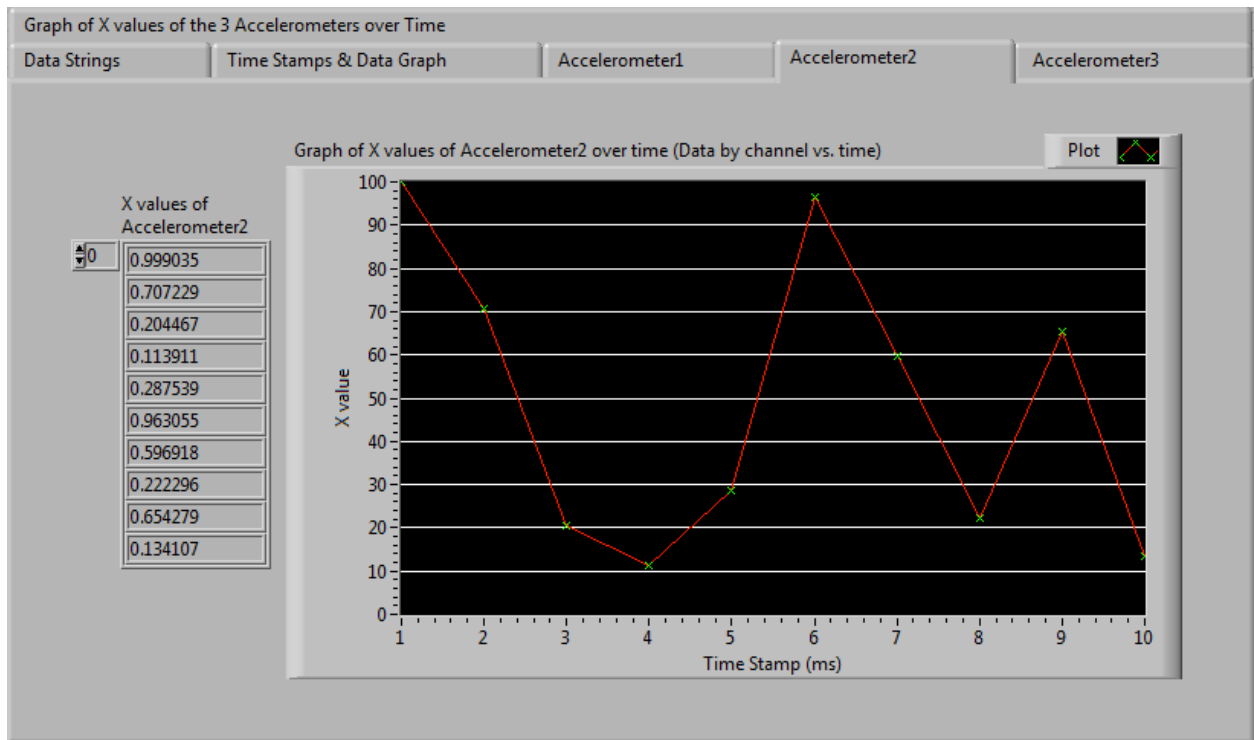


Fig. 12 Graph of X values of the second accelerometer over time presented in the LabVIEW front panel.

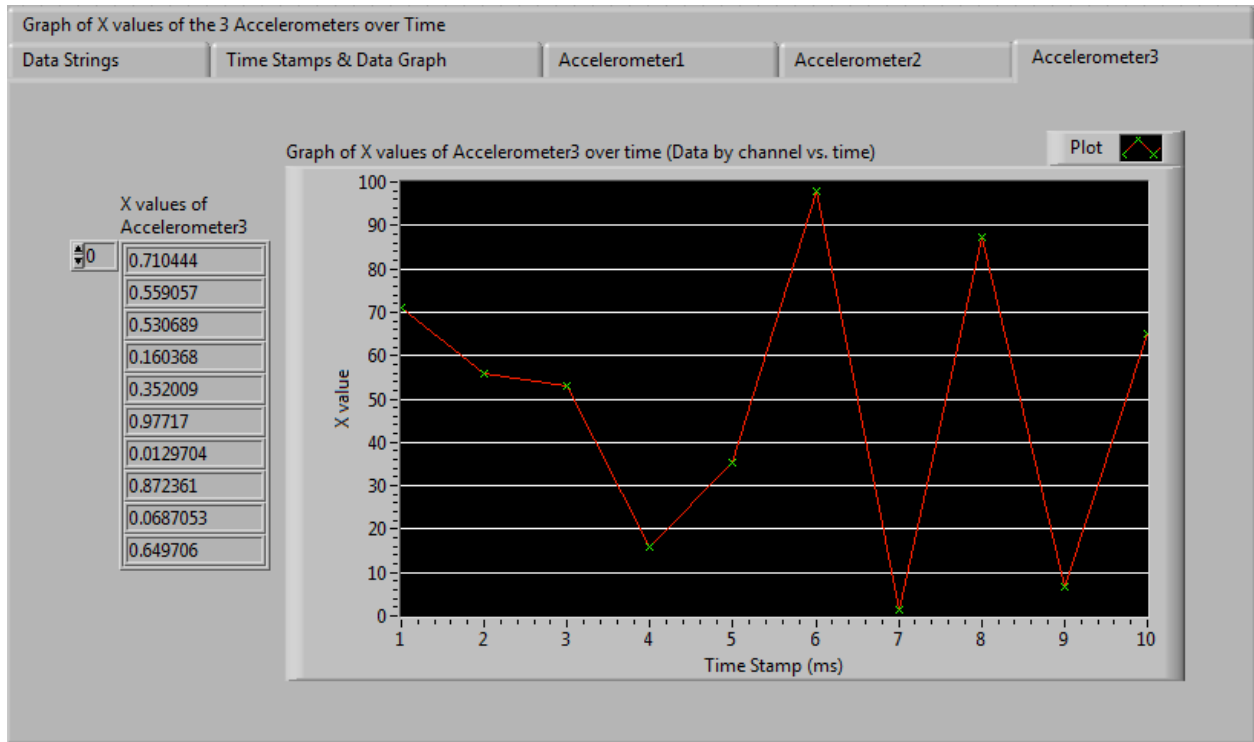


Fig. 13 Graph of X values of the third accelerometer over time presented in the LabVIEW front panel.

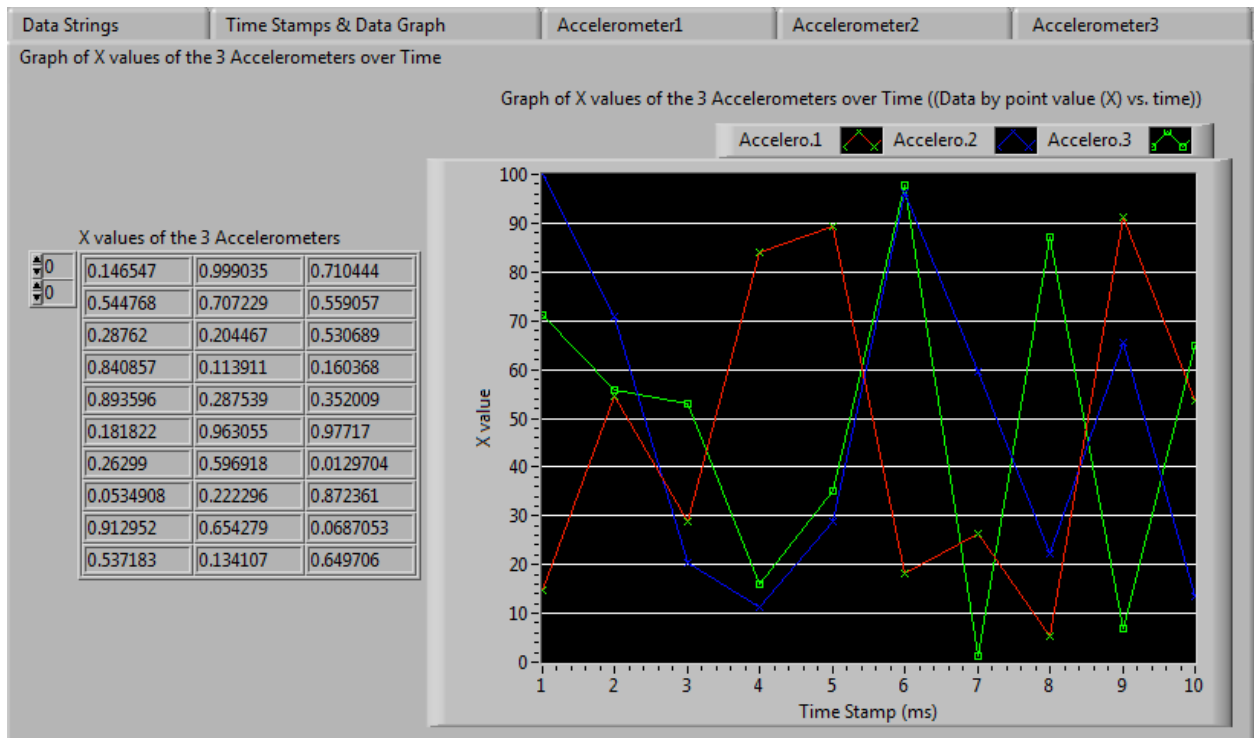


Fig. 14 Graph of X values of all three accelerometers over time presented in the LabVIEW front panel.

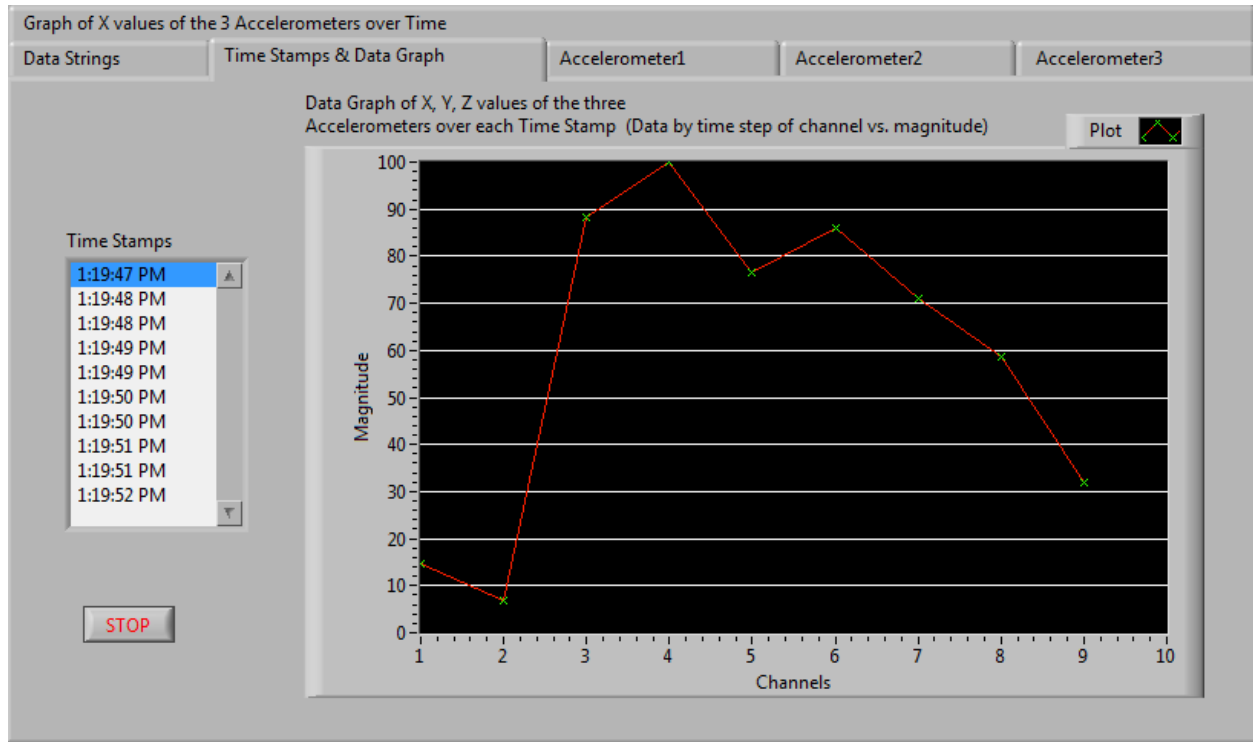


Fig. 15 Graph of all three accelerometers channels values vs. magnitude (x , y and z -axis values) over each separate time period presented in the LabVIEW front panel.

three accelerometers channels (nine numbers of x , y and z) versus the magnitudes (the values of x , y and z) over each separate time period (10 time stamps) are displayed in different format as shown previously in Fig. 15.

Finally and effectively, the results of the writing and reading of the text-file as a result of the simulation of accelerometer sensors can be easily set in diagrams of the LabVIEW programming language, as well as the output data of the simulation signals that are given by the written program (i.e., in block diagrams) and stored in an external provenance as a text-file are applied to the reading script code, which is coded by the block diagrams in the LabVIEW and shown in the front panels. Besides, the results of the LabVIEW front panels shown in appendices Figs. 1-7 show the simulation data of the semi-accelerometer signals that have been adapted by the LabVIEW program. Roughly, the LabVIEW programming that has been adapted to write the script codes could be easily understood and handled, modifiable, and getting better results than any other codes in programming

languages.

5. Conclusions and Recommendation

Presently, the development in agricultural practices through applying diverse application techniques has been increasing, such as using the modern electrical circuits, electronic instruments and the software programming languages (e.g., LabVIEW program) in the field of precision agriculture which is one of the much important factors that contribute to change in the agriculture operation practices. The LabVIEW driver software (version 16.0) is very convenient application that is easy to use and adapts well to any simulation situation, or can be used to perform an assessment of any system response, as well as to help identify the special requirements for the design of a unique system at certain conditions. When used properly, the unique LabVIEW tools and architectures are an effective method to demonstrate real time, real life applications in a rather simple way.

In summary, the first phase to develop an operable semi-accelerometers simulation can be accomplished

by writing a readable text file from the LabVIEW block diagram and front panel code (national instruments circuit dialog), and the output of this simulation circuit as x , y , z -axis coordinate values. Then, by plugging the x , y , z -axis coordinate values of that simulation into an easily read text file dialog through a new LabVIEW block diagram, the original coordinate values (time and three orthogonal axes x , y , and z) can be revealed in a new LabVIEW front panel as digital data or graphs. The LabVIEW simulations are simple to deployed, accurate in detail, and the processes are easily taught to students and interested in applying modern technology. To conclude, in an instructional situation, practice by simulation with the LabVIEW applications can promote greater understanding of the principles of acceleration and their application in real time situations that would satisfy certain conditions. Indeed, the LabVIEW software is a group of palettes prepared and accommodated well for any simulation environment or any practical application and have been recommended by the LabVIEW users.

References

- [1] Icon-Technologies. 2017. "FAQ: The LabVIEW Programming Language." Accessed June 28, 2017. <http://www.icon-tech.com.au/s55/faq-the-labview-programming-language/>.
- [2] Hinkson, J. A., Chin, M., Kim, C. H., and Nishimura, H. 1994. "Automated Tune Measurements in the Advanced Light Source Storage Ring Using a LabVIEW Application." In *Proceedings of the 4th European Particle Accelerator Conference*, 1737-9.
- [3] Halvorsen, H. P. 2016. "Control and Simulation in LabVIEW." University College of Southeast Norway, Norway. Accessed October 28, 2016. <http://home.hit.no/~hansha/documents/labview/training/Control%20and%20Simulation%20in%20LabVIEW/Control%20and%20Simulation%20in%20LabVIEW.pdf>.
- [4] Halvorsen, H. P. 2016. "Tutorial: Control and Simulation in LabVIEW." University College of Southeast Norway, Norway. June 27, 2017. <http://home.hit.no/~hansha/?tutorial=control>.
- [5] Porter, T. L., and Delinger, W. 2006. "Electronics for LabVIEW Based Piezoresistive Microcantilever Sensor System." *Sensors & Transducers* 68 (6): 568-74.
- [6] Udumala Savary, S. K. J., Ehsani, R., Salyani, M., Hebel, M. A., and Bora, G. C. 2011. "Study of Force Distribution in the Citrus Tree Canopy during Harvest Using a Continuous Canopy Shaker." *Computers and Electronics in Agriculture* 76 (1): 51-8.
- [7] Figueiredo, R. C., Ribeiro, A. M. O., Arthur, R., and Conforti, E. 2011. "Remote SMS Instrumentation Supervision and Control Using LabVIEW." In *Practical Applications and Solutions Using LabVIEW™ Software*, edited by Folea, S. Croatia: Intech, 318-42.
- [8] Desai, A. K., Bharate, A. G., Rane, V. P., Gholap, S. V., and Mitra, A. C. 2017. "LabVIEW Programming for Vibration Analysis." *IOSR Journal of Mechanical and Civil Engineering* 1: 1-5.
- [9] Bhagat, S. N., and Nalbalwar, S. L. 2016. "LabVIEW Based Tool Condition Monitoring and Control for CNC Lathe Based on Parameter Analysis." In *Proceedings of 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 1386-8.
- [10] Shah, H. S., Patel, P. N., Shah, S. P., and Thakker, M. T. 2013. "8 Channel Vibration Monitoring and Analyzing System Using LabVIEW." In *Proceedings of 2013 Nirma University International Conference on Engineering (NUICONE)*, 1-4.
- [11] Lita, I., Visan, D. A., Mujea, G., and Ghita, D. 2005. "LabVIEW Application for Analysis of Mechanical Vibrations from Industrial Environment." In *Proceedings of the 28th International Spring Seminar on the Electronic Technology: Meeting the Challenges of Electronics Technology Progress*, 464-8.
- [12] National Instruments Corporation. 2016. *National Instruments LabVIEW 2016 Software*. Version 16.0. Austin, Texas: National Instruments Corporation.

Development of a LabVIEW Application for Measurement and Analysis of Acceleration Signals from an External Reference

Appendices

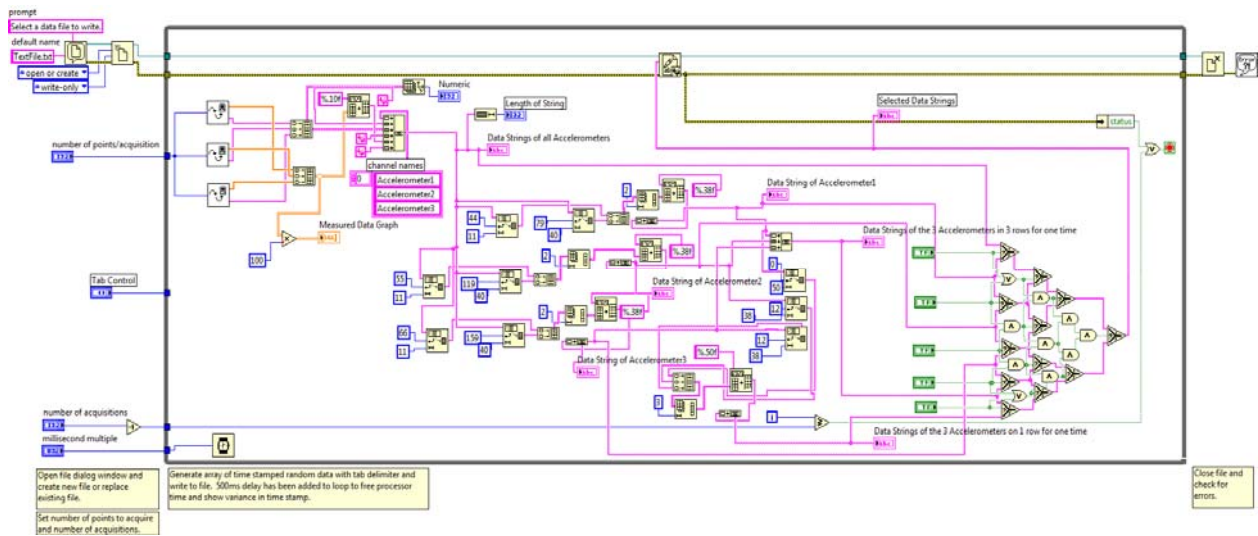


Fig. 1 Block diagram code for selecting data file to write.

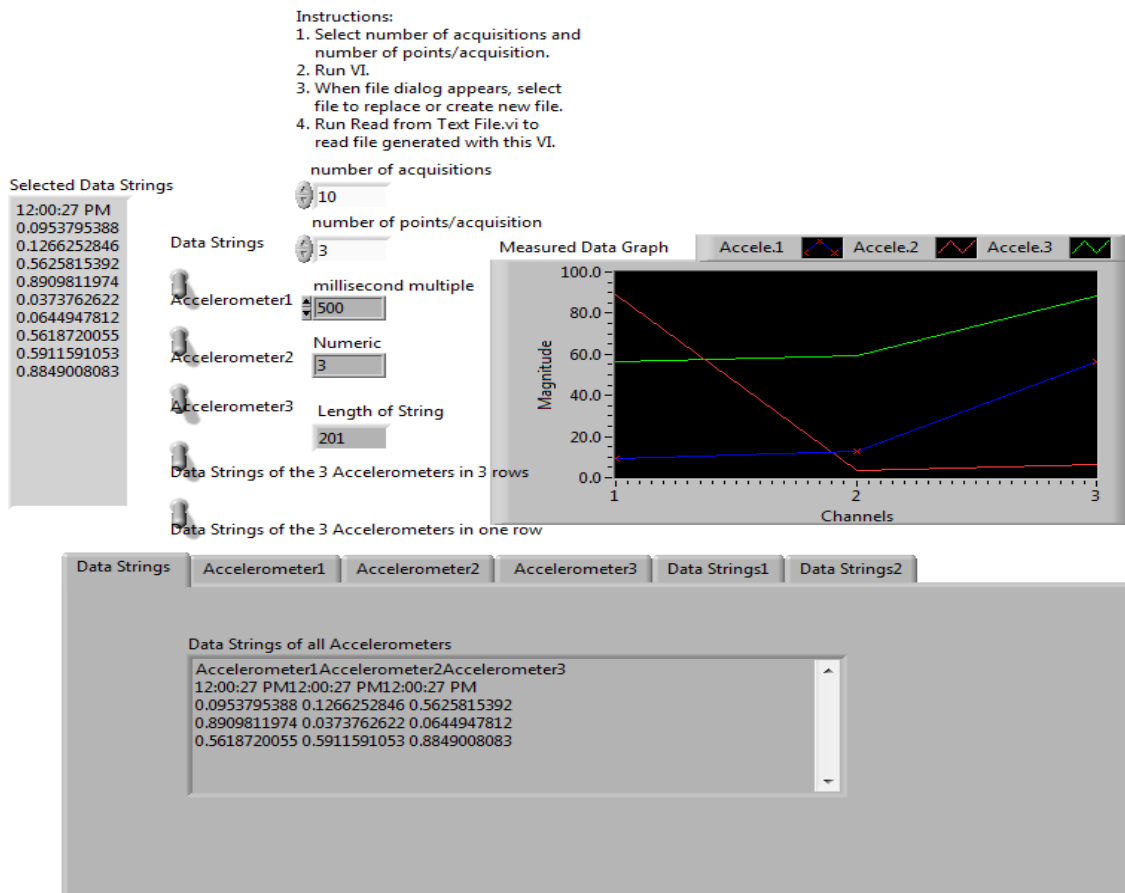


Fig. 2 The LabVIEW front panel shows all three accelerometers data and graph at the last time period.

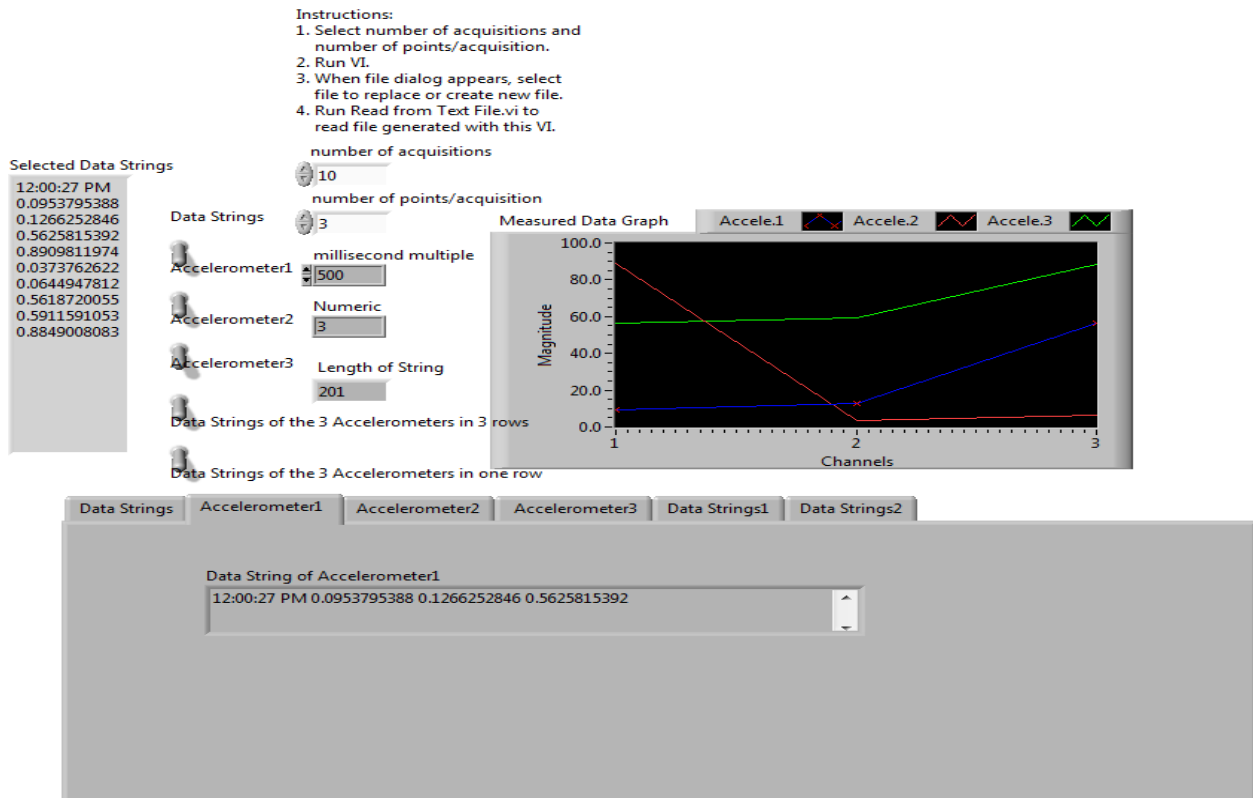


Fig. 3 The LabVIEW front panel shows the first accelerometer data at the last time period.

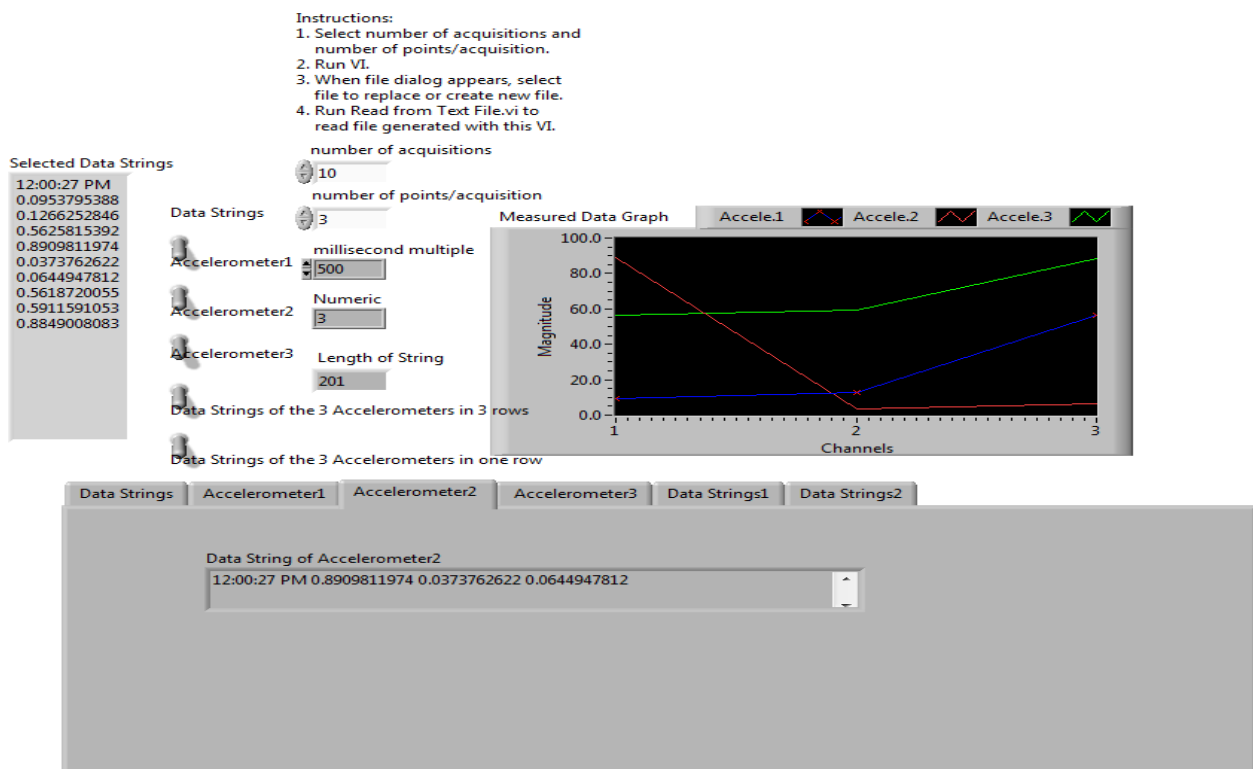


Fig. 4 The LabVIEW front panel shows the second accelerometer data at the last time period.

Development of a LabVIEW Application for Measurement and Analysis of Acceleration Signals from an External Reference

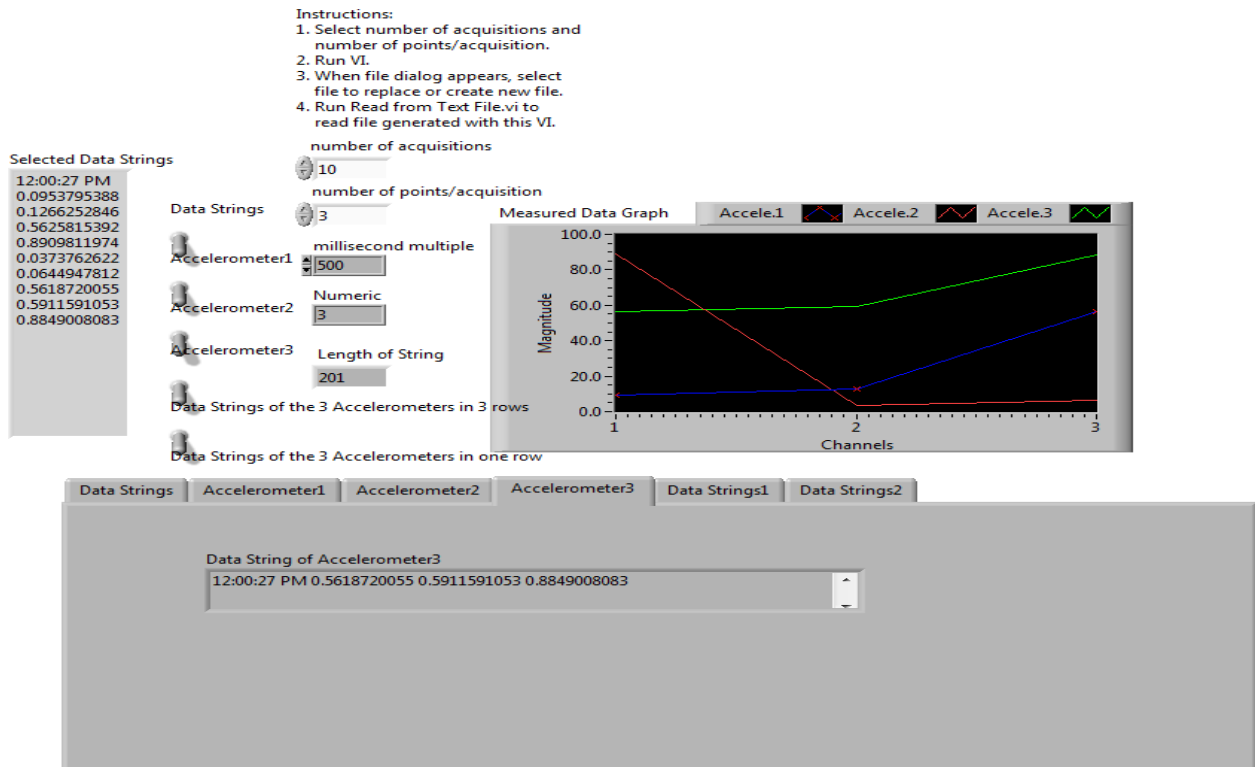


Fig. 5 The LabVIEW front panel shows the third accelerometer data at the last time period.

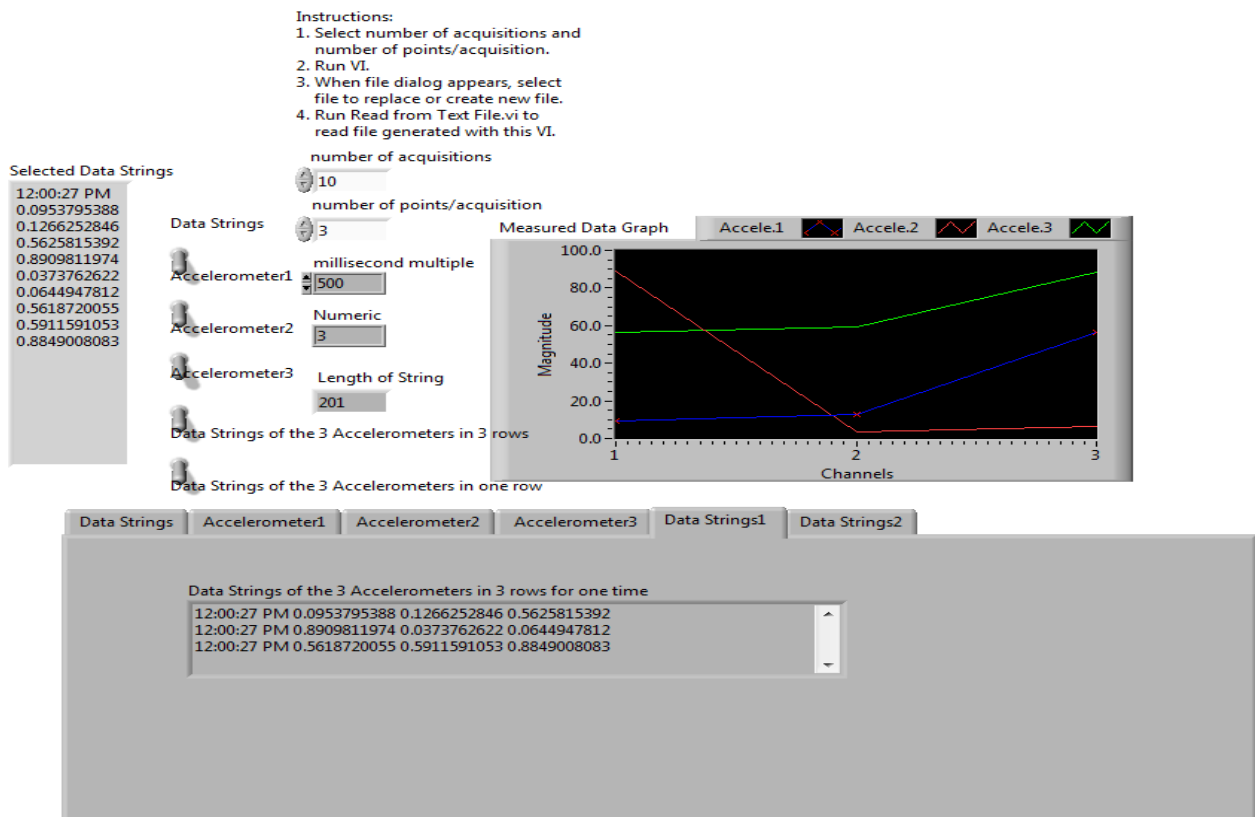


Fig. 6 The LabVIEW front panel shows all three accelerometers data in three rows and graph at the last time period.

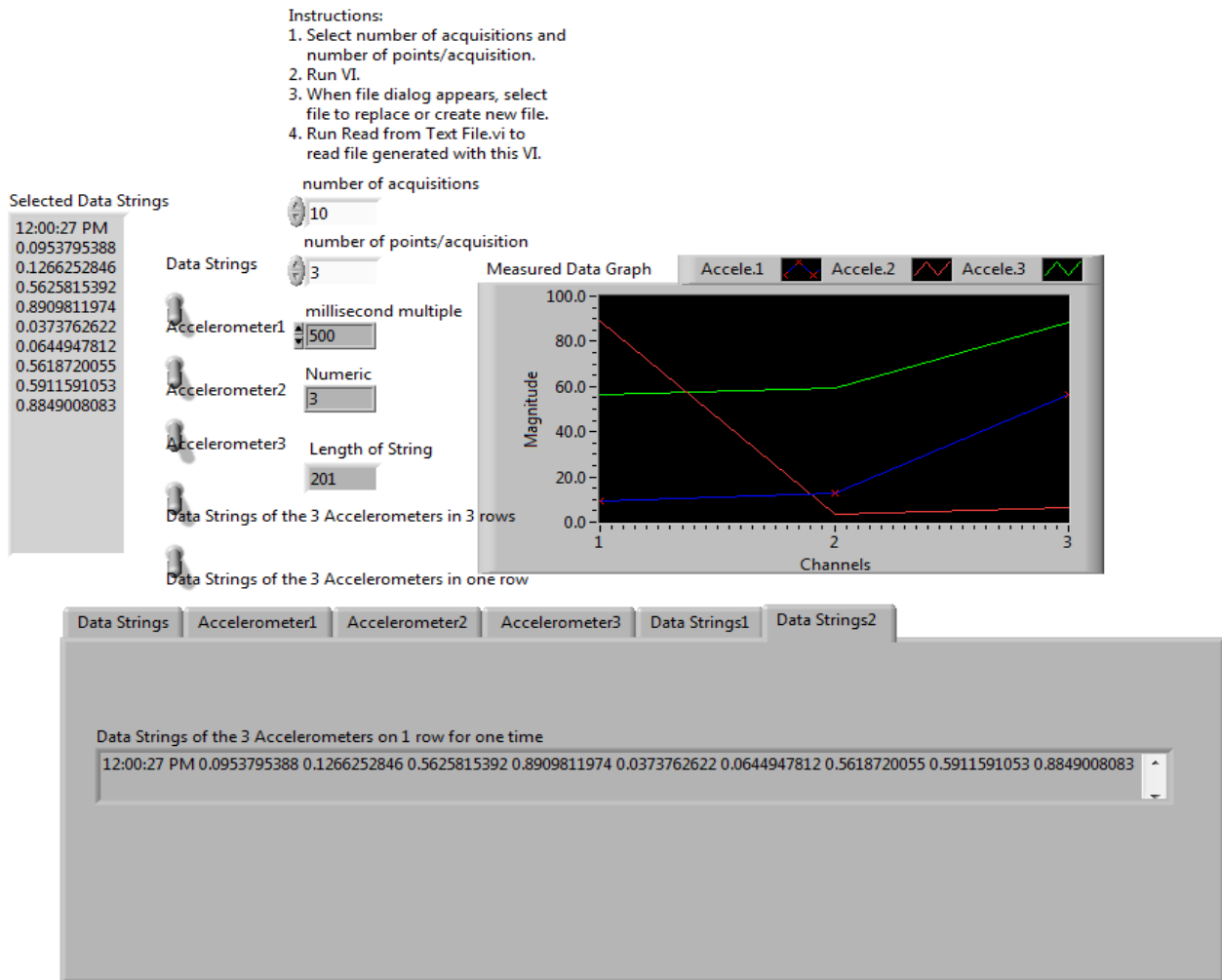


Fig. 7 The LabVIEW front panel shows all the three accelerometers data in one row and graph at the last time period.