Journal of
Engineering Research

MULTICHANNEL VIBRATION ANALYZER FOR DYNAMIC SYSTEMS

Vicente Miguel Diaz de León Santiago
TNM-Veracruz Technological Institute, Miguel Ángel de Quevedo 2779, Veracruz, Ver., CP 91897, Mexico

Diego Alejandro Castro Ramón
TNM-Veracruz Technological Institute, Miguel Ángel de Quevedo 2779, Veracruz, Ver., CP 91897, Mexico

Luis Alberto Ortega Moguel
TNM-Veracruz Technological Institute, Miguel Ángel de Quevedo 2779, Veracruz, Ver., CP 91897, Mexico

1

Abstract: This article presents the design of a multichannel vibration analyzer that, together with several tests, for teaching the effects of vibrations on dynamic systems in the subject of robotics, mechanical vibrations, maintenance, among others. Using Simulink and Arduino microcontrollers, it was possible to build a device for visualizing vibrations in dynamic devices using low-cost sensors. Its performance and the ease in the use of Simulink with the Arduino support libraries have been evaluated to be used in a prototype of a didactic dynamic balancing and alignment machine.

Keywords: microcontroller (microcontroller), robotics (robotics), sensors (sensors), mechanical vibrations (mechanical vibrations), dynamic system (Dynamic systems).

INTRODUCTION

In this article we will present the design and construction of a vibration analyzer equipment to solve the main problem that is to correct or calibrate mechanical-dynamic devices. This will be possible through the vibration analyzer equipment, since through it we acquire the collected data and represent it in real time as a graph.

The system of this analyzer has 4 different channels, to have access to generate multiple readings in real time, however, two of the implemented analog sensors that it has include 3 different axis orientations (x, y, z), and another pair with a single axis, each one with an indicator to visualize the spectrum in a graphic and simple way by means of signals generated and presented by means of a computer system.

MATERIALS AND METHODS

Taking into account the problems previously exposed, in this section the analysis will be carried out in order to provide a solution to the presentation of vibration phenomena within mechanical systems and to present components, design, and construction of the equipment.

COMPONENTS AND FEATURES

Multidirectional Analog Accelerometer (ADXL335)

The multidirectional analog accelerometer, has an ADXL335 chip, which offers a range of +/- 3g. It has 3 analog outputs, each one is proportional to the acceleration in the x and i z axes. These analog outputs can be read directly by an Arduino card in order to transfer the data using Simulink-Matlab.
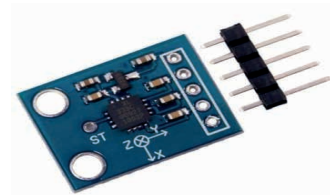

Figure 1. Module: Multidirectional analog accelerometer (ADXL335).

Characteristics

The multidirectional analog accelerometer (ADXL335), has 5 pins as can be seen in Fig.1, it works with only 3.3V. Other features to highlight are:1.8 working power, 3.3V, compatible with Arduino, PIC, AVR, analog outputs, low consumption of 350 uA.

SYSTEM MODEL

For data acquisition, a mathematical model of the system was developed to describe the behavior of the signals obtained from each of the channels that have analog sensors (ADXL335). The mathematical models were generated with the help of Simulink-Matlab software tools in block code with a friendly interface and multiple communication libraries. Another software that was used in this analyzer to check the operation and programming of the analog sensors was

Arduino IDE, which has a wide range of libraries for each of the modules generated by the developers of the platform and by the community.

The reading of the multidirectional analog accelerometer (ADXL335) are only values that it shows us by default, a formula will have to be implemented to be able to calculate the acceleration at a point through its vibration.

Next, the values of the reading of the multidirectional analog accelerometer (ADXL335) will be taken to calculate the voltage in each of the axes either in (X, Y, Z) with the help of "eqs. (1-3)". The models will give us the necessary data, they will be generated from the values collected in each axis as shown in Fig.2. Subsequently, with the data acquired, we will calculate the voltage in each of the axes with the model in Fig.3.

$$Vx = \frac{(5.0)(x)}{1023} \tag{1}$$

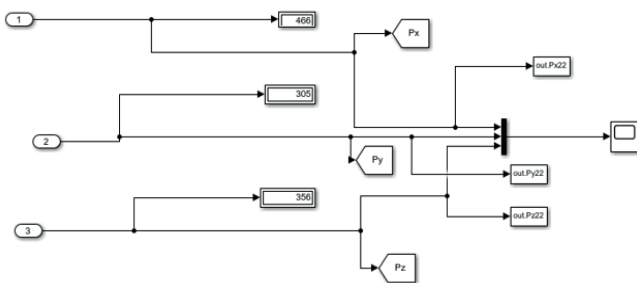$$Vy = \frac{(5.0)(y)}{1023} \tag{2}$$

$$Vz = \frac{(5.0)(z)}{1023} \tag{3}$$
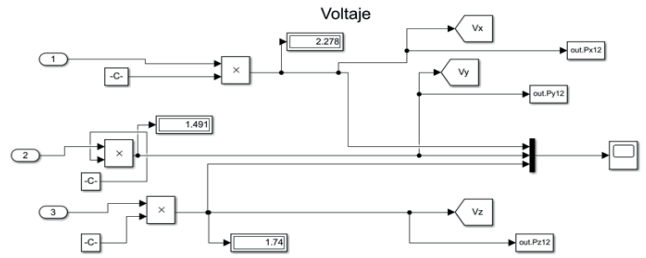

Figure 3. Model of the voltage system.

Where 5V/1023 will be introduced, which are values determined for the use of the conversion to voltage and then to be able to occupy the voltages in each axis with their respective formula to calculate the acceleration with the help of "Eqs. (4-6)". To acquire the acceleration data of each axis we generate a model as shown in Fig.4.

$$Ax = \frac{(Vx - Vplanox)}{0.33} \tag{4}$$

$$Ay = \frac{(Vy - Vplanox)}{0.33} \tag{5}$$

$$Az = \frac{(Vz - Vplanoz + 0.33)}{0.33} \tag{6}$$

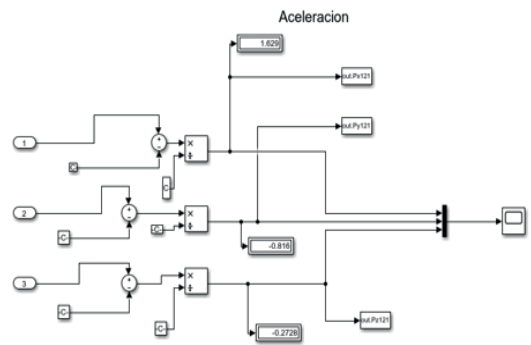Figure 2. System model on sensor data (ADXL335).

Figure 4. Model of the acceleration system.

Taking into account the given acceleration values, the position angle for each of the axes will be obtained as shown in Fig.4., and "eqs. (7-9)".

3

$$Anx = \frac{\frac{Ay}{Az}(180)}{3.14} + 180 \qquad (7)$$

$$Any = \frac{\frac{Az}{Ax}(180)}{3.14} + 180 \qquad (8)$$

$$Anz = \frac{\frac{Ax}{Ay}(180)}{3.14} + 180 \qquad (9)$$

vx= Initial voltage of the X axis

vy= Y axis initial voltage

vyz= Initial voltage of the Z axis

vplanox= Voltage in the plane of the X axis

Vplanoy= Voltage in the plane of the Y axis

vplanoz= Voltage in the plane of the Z axis

Ax=Acceleration in the plane of the x axis

ouch=Acceleration in the plane of the y-axis

Az=Acceleration in the plane of the z axis

Anx=Angle of Position in the plane of the x axis

Any=Angle of Position in the plane of the y-axis

Anz=Angle of Position in the plane of the z axis

Note: The 0.33 is the voltage with which you are working.

Arranging the models in different channels to see their response in acceleration as can be seen in Fig. 5.

Mainly the use of the blocks was to analyze the vibrations of the sensors, for which a mathematical model with block programming was designed in Simulink, representing the conditions of the previously established formula.

Finally, with the implementation of the Simulink tool from Matlab, it is possible to make a more user-friendly interface and it shows in a summarized way the system of each one of the channels of each sensor.
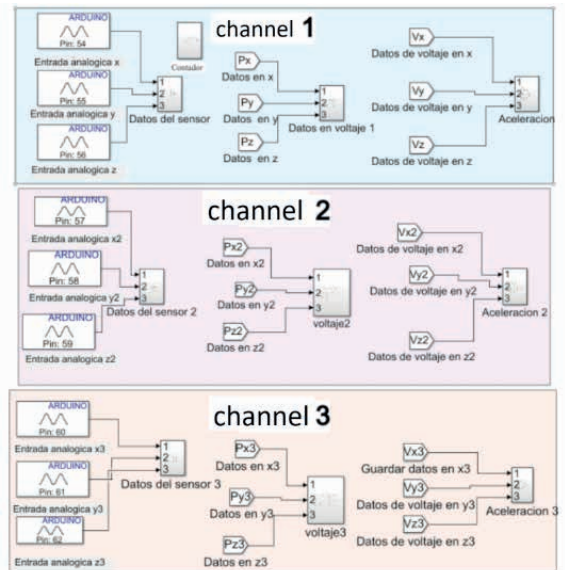

Figure 5. Presentation of the systems of each channel.

In each of the 3 different channels, we are shown 3 subsystems, which are sensor data, voltage data, acceleration as shown in Fig.6. In this section we will focus on channel 1.
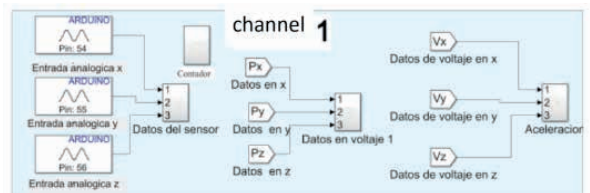
channel 1

Figure 6. Presentation of channel 1 systems.

In the first section, we can find the sensor data for each of its axes Fig.7.
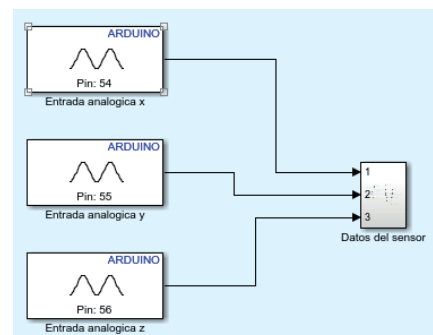
data block

Figure 7. Channel 1: Sensor 1 data section.

In Fig.8, we can see the block code for obtaining the data in the x axis, in the y axis, and finally in the z axis. It is worth mentioning that this subsystem belongs to channel 1.
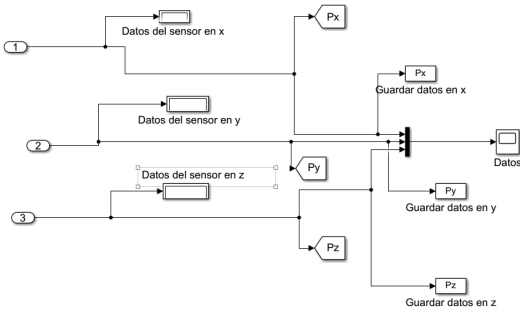

Figure 8. Code blocks to obtain data from channel 1.

voltage block

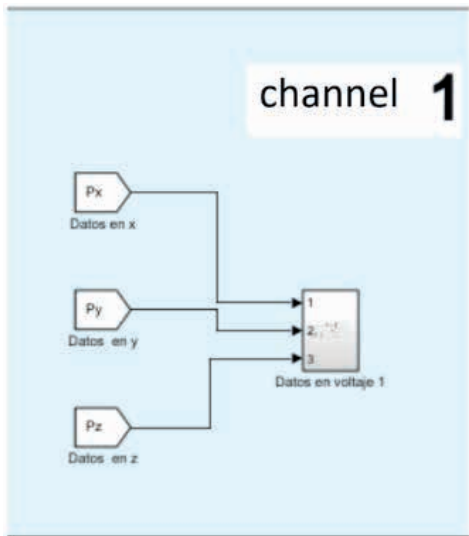In the second section, we can find the sensor data regarding the system voltage for each of its axes as in Fig.9.

Figure 9. Channel 1: Sensor 1 voltage section.

If you want to know the data thrown for the voltage system, we go to the second section of channel 1 and we will obtain the information of each one of the axes of Fig. 10.
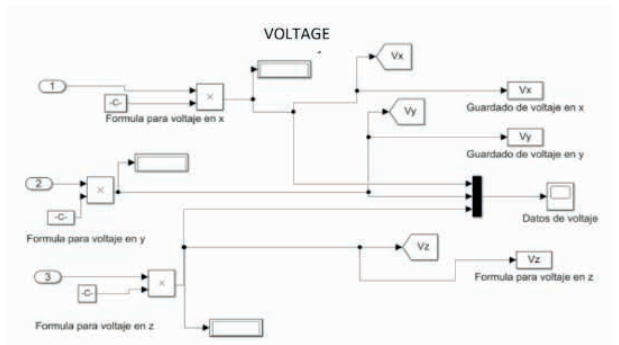
Figure 10. Code blocks for the sensor 1 voltage section.

Acceleration

In this third section we will find the sensor data regarding the acceleration of the system for each of its axes.
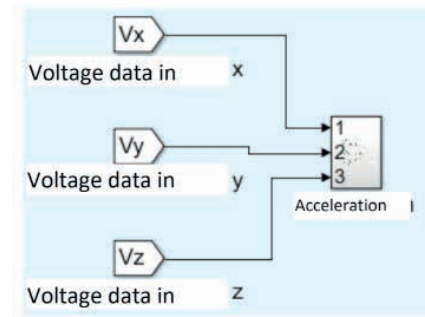

Figure 11. Channel 1: Sensor 1 acceleration section.

If you want to know the data thrown for the acceleration system, we go to the third section of channel 1 and we will obtain the information of each one of the axes of Fig.11.
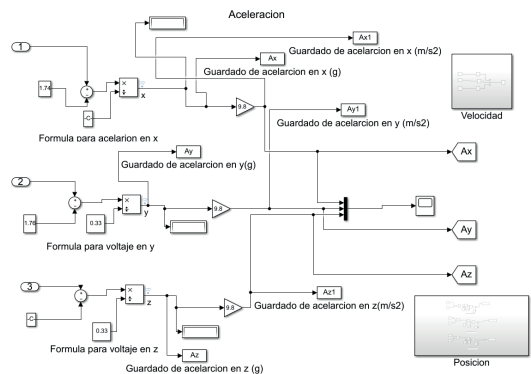

Figure 12. Code blocks for the acceleration section of sensor 1.

As can be seen in Fig.12, we can observe the block code to obtain the acceleration data in the x axis, in the y axis, and finally in the z axis. It is worth mentioning that this subsystem belongs to channel 1.

Channel 2

For channel 2, we have 3 subsystems. The first is for sensor data, the second is for voltage data, and the third is for acceleration as shown in Fig.13.
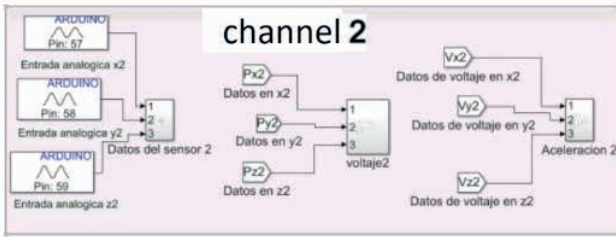

Figure 13. Presentation of Channel 2 systems.

data block

In the first section, we can find the sensor data for each of its axes as shown in Fig. 14.
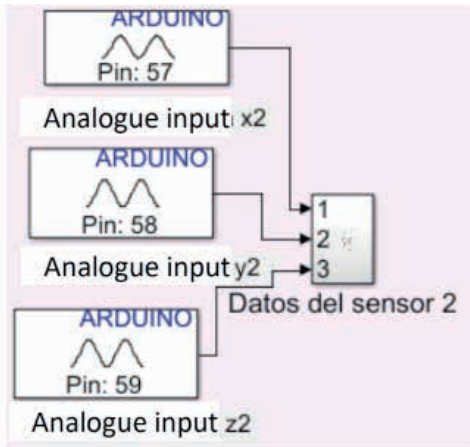

Figure 14. Channel 2: Sensor 2 data section.

As it can be seen in Fig.15, we can observe the block code to obtain the data in the x axis, in the y axis, and finally in the z axis. It is worth mentioning that this subsystem belongs to channel 2.
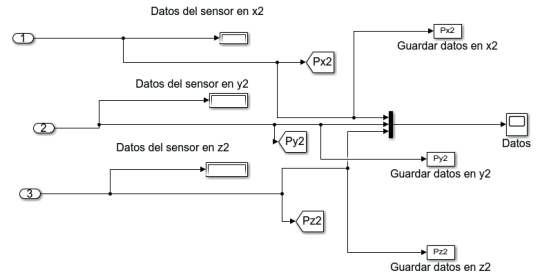

Figure 15. Code blocks for the sensor 2 data section.

voltage blocks

In the second section, we can find the sensor data regarding the system voltage for each of its axes as in Fig.16.
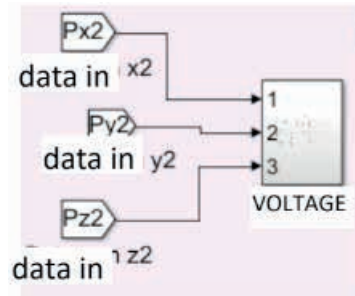

Figure 16. Channel 2: Sensor 2 voltage section.

If you want to know the data thrown for the voltage system, we go to the second section of channel 2 and we will obtain the information of each one of the axes as in Fig.17.
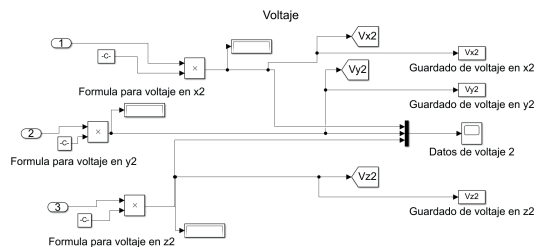

Figure 17. Code blocks for the sensor 2 voltage section.

acceleration block

In this third section we will find the sensor data regarding the acceleration of the system for each of its axes as shown in Fig.18.
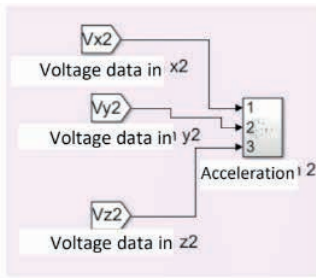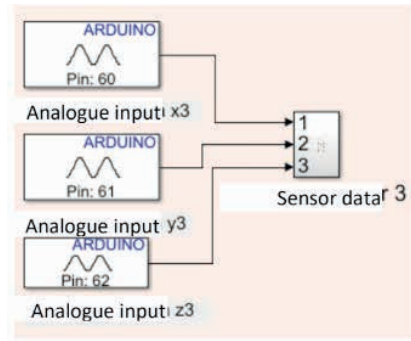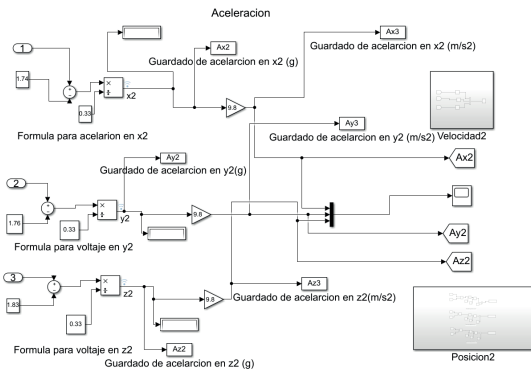
Figure 18. Channel 2: Sensor 2 acceleration section.

Figure 19 Code Blocks for Sensor 2 Acceleration Section

As can be seen in Fig.19, we can observe the block code to obtain the acceleration data in the x axis, in the y axis, and finally in the z axis. It is worth mentioning that this subsystem belongs to channel 2.

Channel 3

And finally for channel 3, we have subsystems just like channel 1 and channel 2. The first subsystem is for sensor data, the second is for voltage, and the third is for acceleration as shown in the figure. Fig.20.

Figure 20. Presentation of channel 3 systems.

Data block

In the first section, we can find the sensor data for each of its axes as can be seen in Fig.21.

Figure 21. Channel 3: Sensor 3 data section.

As can be seen in Fig.22, we can observe the block code to obtain the data in the x axis, in the y axis, and finally in the z axis. It is worth mentioning that this subsystem belongs to channel 3.
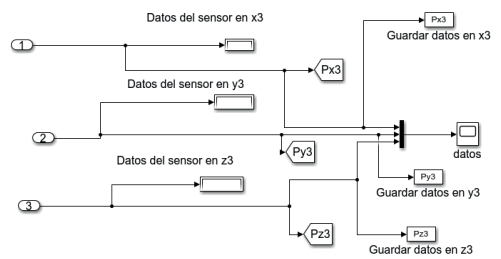

Figure 22. Code blocks for the sensor 3 voltage section.

Voltage block

In the second section, we can find the sensor data regarding the system voltage for each of its axes as in Fig.23.
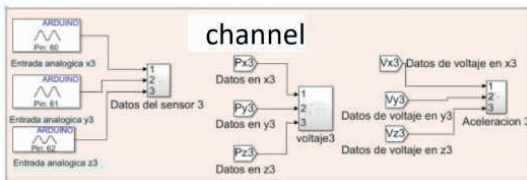

Figure 23. Channel 3: Sensor 3 voltage section.
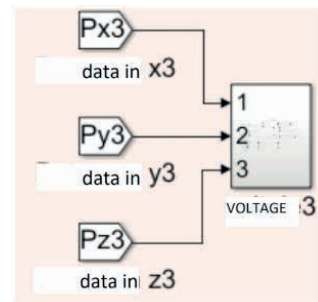
If you want to know the data thrown for the voltage system, we go to the second section of channel 3 and we will obtain the information of each one of the axes as shown in Fig.24.
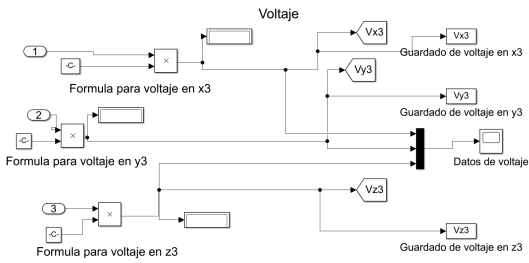

Figure 24. Code blocks for the sensor 3 voltage section.

acceleration block

In this third section we will find the sensor data regarding the acceleration of the system for each of its axes as in Fig.25.
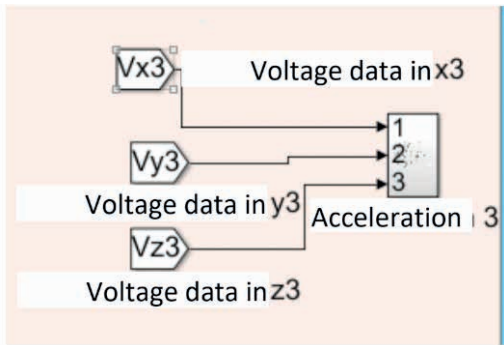

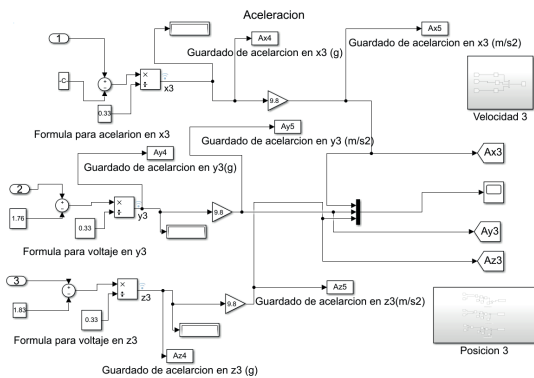Figure 25. Channel 3: Sensor 3 acceleration section.

Figure 26. Code blocks for the acceleration section of sensor 3

As can be seen in Fig.26, we can observe

the block code to obtain the acceleration data in the x axis, in the y axis, and finally in the z axis. It is worth mentioning that this subsystem belongs to Channel 3.

CONNECTIONS AND CIRCUIT

Multidirectional Analog Accelerometer (ADXL335)

The multidirectional analog accelerometer (ADXL335) has 5 pins. The connections of this accelerometer with the Arduino MEGA 2056 are shown below Fig.27.
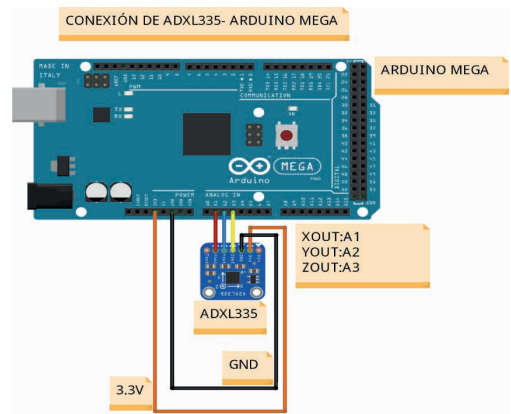

Figure 27. Connection of the multidirectional analog accelerometer (ADXL335) with the Arduino Mega 2560 microcontroller.

Analog accelerometer connection pins (ADXL335):

- Power (3.3V).

- X-OUT: Output of the acceleration signal on the X axis.

- Y-OUT: Acceleration signal output on the Y axis.

- Z-OUT: Z axis acceleration signal output.

- Ground (GND).

The final connection of the sensors with the microcontroller is shown in Fig.28. The components cover the analog connections from A0 to A8 of our Arduino.
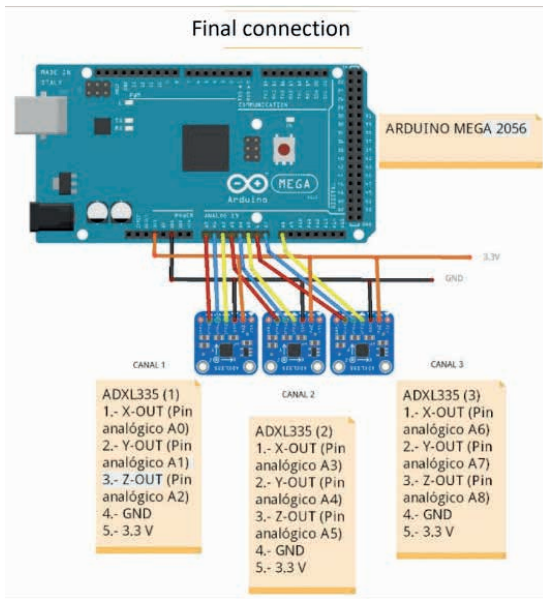
Figure 28. Final connection.

COMMUNICATION

Matlab is one of the pioneer software's in the application of engineering for the development of solutions to problems of all kinds. MathWorks is a matrix programming environment for algorithm development, data analysis, simulation, and numerical computation.

Simulink is a dedicated graphical environment for the model-based design and simulation of multi-domain and integrated dynamic systems. In addition, MathWorks makes more than 80 additional products available for specialized tasks such as data analysis and image processing.

Matlab is an abbreviation of the term "Matrix Laboratory", since it was initially called a matrix programming language and is considered a fourth generation programming language.

RESULTS

Mainly the use of the blocks was to analyze and observe in a more graphic way the vibrations of the sensors, for which its mathematical model was made with block programming in Simulink, representing

the conditions of the previously established formula. Through the output of our block program in Matlab-Simulink, it shows us a window in which we can observe the signals corresponding to the axesx,and,z ofeach of the channels Fig. 29 and Fig. 30.
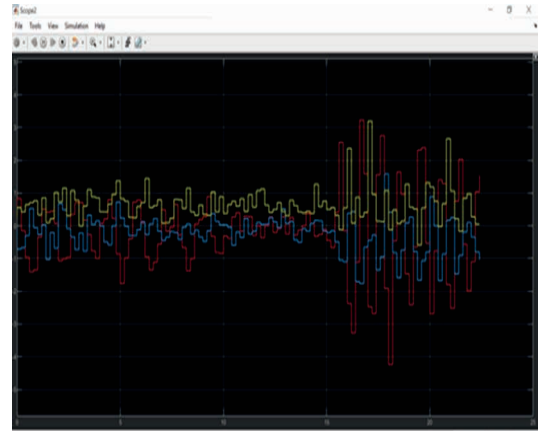

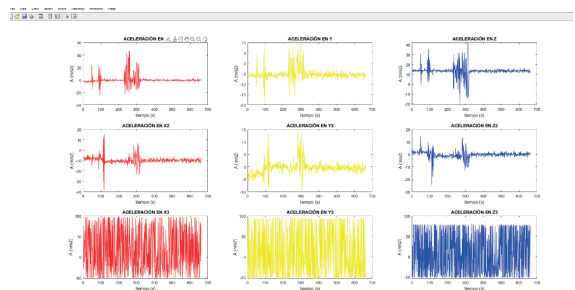Figure 29. Obtaining the signals (x, y, z) in real time from the sensors when applying impact to a surface.

Figure 30. Graphs thrown from the system.

In the graph produced by Matlab within the time range considered the infinite time variable, the continuous behavior is shown when a blow is applied to the surface to generate a vibration and in turn generate an analog signal. It has 3 analog outputs, one is the voltage given to the sensor and the other sensors are proportional to the acceleration in that axis x, i z. These outputs can be read directly by an Arduino card or by any analog-to-digital converter.

Using the Matlab-Simulink data inspector tool, the different types of output signals can be

analyzed in real time for a better visualization of the signals corresponding to the x, y, and z axes of each of the channels Fig.31.
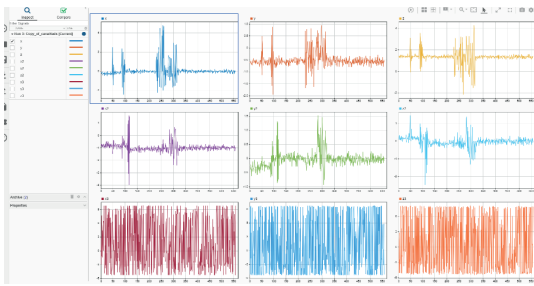

Figure 31. Graphs of the 3 channels of each of its axes.

At the end of the simulation, we can analyze the data that was automatically saved in an Excel document using divided tables for each of the channels 1, 2 and 3 as shown in Fig.32.
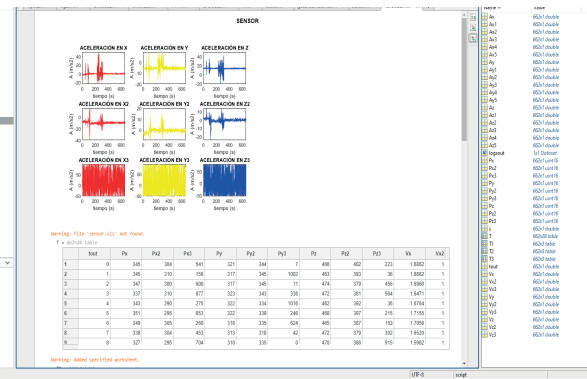

Figure 32 Simulation data

CONCLUSIONS

In accordance with what was previously seen in the investigations and the results obtained, the administrative support activities which were presented in a timely manner, both synchronously and asynchronously, and the development of a project or device which would allow us to record the vibrations of different environments or scenarios replacing the high cost sensors normally used in companies, by modules and sensors for a microcontroller with a much cheaper cost and which in turn covers the problem giving a specific solution. Thus achieving a portable

and suitable product to analyze vibrations which has 3 different communication channels for 3 different possible situations, generating signals and graphs of the data acquired from the sensors for each channel that the device has. With knowledge in some design, simulation and programming applications, it was possible to develop the final device, subsequently going through final tests. And because of this, social service is one of the fundamental activities both for academic training and for training in the professional field, since our knowledge is put to the test to solve different problems of daily life.

REFERENCES

[1] Comunicación Serial Arduino Simulink/Matlab - [enero, 2022]. (2021, July 7). Control Automático Educación. https://controlautomaticoeducacion.com/arduino/comunicacion-serial-arduino-simulink-matlab/

[2] ADXL335 Datasheet and Product Info | Analog Devices. (n.d.). Www.analog.com. Retrieved January 14, 2022, from https://www.analog.com/en/products/adxl335.html

[3] Arduino Mega 2560 Rev3 | Arduino Official Store. (n.d.). Store.arduino.cc. https://www.arduino.cc/en/Main/arduinoBoardMega2560

[4] Rao, S. S., David Sepúlveda García, Ricardo Rodríguez Figueroa, & Del, G. (2012). *Vibraciones mecánicas*. Pearson.