



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

DEGREE THESIS

TITLE: Characterization of noise in MEMS accelerometers for femtosats
DEGREE: Bachelor's degree in Airnavigation Engineering
AUTHOR: Oscar Liébana Moradillo
ADVISOR: Jordi Gutiérrez Cabello
DATE: July, 24th 2018

Title: Characterization of noise in MEMS accelerometers for femtosats
Author: Oscar Liébana Moradillo
Advisor: Jordi Gutiérrez Cabello
Date: July, 24th 2018

Abstract

The aim of this thesis is to determine the characteristics of the noise of an ultralow noise MEMS accelerometer. The study of these characteristics will allow to determine if the accelerometer can be used in the scientific payload aboard a femtosat. And in the case that it is used, it will allow to calibrate the accelerometer to provide a better signal.

The method for studying the noise characteristics is the Allan Variance, which allows to identify and determine the most important noise terms that affect accelerometers. An experiment to perform the Allan Variance has been designed and performed.

The sensor studied is ultralow noise and for this reason it must be isolated from its surroundings during the experiment, which will corrupt the output signal and not allow to clearly study the noise of the sensor. To achieve this task two isolation systems have been designed and built, one for protecting the sensor from vibrations transmitted through air like currents of air created either naturally or by people moving in the room of the experiment. And another to isolate the sensor from vibrations travelling through ground such as the natural vibrations of the ground or vibrations caused moving objects in contact with the floor or people walking. These isolation systems have been installed in an underground laboratory located on the EETAC campus to ensure that there are almost no sources of vibration present during the experiment and that there are no changes in the temperature.

The isolation systems have worked properly and the sensor has not been affected by external accelerations. The results obtained are satisfactory, they agree with the information provided by the datasheet and it is considered that the sensor noise properties have been characterized properly.

Even though the results are satisfactory, it cannot be yet defined if the sensor will be used for the scientific payload aboard a femtosat.

Resum

L'objectiu d'aquest treball de fi de carrera és analitzar les característiques del soroll d'un acceleròmetre MEMS de molt baix soroll. L'estudi d'aquestes característiques permetrà determinar si l'acceleròmetre estudiat podrà ser utilitzat en un experiment científic a bord d'un femtosat. En el cas de que sigui utilitzat, també permetrà calibrar l'acceleròmetre per millorar la senyal.

El mètode utilitzat per estudiar les característiques del soroll és la Variança d'Allan. Aquest mètode permet identificar i quantificar els tipus de soroll més importants que afecten als acceleròmetres. Un experiment per poder obtenir la Variança d'Allan ha estat dissenyat i dut a terme.

El sensor que s'ha estudiat té un soroll extremadament baix a la mesura. Per aquesta raó ha d'estar aïllat de l'entorn, ja que les vibracions del terra o l'aire podrien corrompre les lectures del sensor i impossibilitar estudiar el soroll de l'acceleròmetre. Per aconseguir aquesta tasca, dos sistemes d'aïllament s'han dissenyat i construït. Un per protegir el sensor de vibracions transmèses per l'aire, corrents d'aire i moviments d'aire causats per persones movent-se dins del laboratori en el que s'està fent l'experiment. L'altre sistema s'ha creat per aïllar l'acceleròmetre de vibracions transmèses pel terra com les vibracions naturals del terra, fenòmens microsísmics o vibracions causades per persones caminant al voltant de l'experiment. Aquest sistema d'aïllament s'han instal·lat en un laboratori subterrani situat al campus de l'EETAC per assegurar-se de que hi ha poques fonts de vibracions al voltant de l'experiment i que la temperatura es manté constant.

Els sistemes d'aïllament han funcionat correctament i l'acceleròmetre no s'ha vist afectat per acceleracions externes. Els resultats obtinguts són satisfactoris i concorden amb la informació proporcionada pel datasheet del sensor. Es pot considerar que el soroll del sensor ha estat caracteritzat adequadament.

Tot i així encara no es pot saber si l'acceleròmetre estudiat serà utilitzat per l'experiment a bord el femtosat ja que el disseny de la missió es troba en les fases inicials.

I would like thank my friends and family who have helped and guided me through the course of this degree. I would also like acknowledge the support provided by my friends at NanoSat Lab.

Table of contents

Table of contents.....	V
List of Abbreviations	viii
Introduction.....	1
Chapter 1. MEMS Sensor: ADXL355	2
1.1. Choosing the sensor	2
1.2. MEMS Accelerometers.....	2
1.2.1. Principle of Operation	2
1.3. Characteristics	3
1.3.1. Resolution and Noise Properties	4
1.3.1.1. Resolution.....	4
1.3.1.2. Noise	4
Chapter 2. External Noise Sources	9
2.1. Ground Noise and Vibrations	9
2.1.1. Used Data.....	11
2.1.2. Analysis	11
2.1.3. Results.....	13
2.1.3.1. CGAR – Garraf	13
2.1.3.2. VILA – Vilanova i la Geltrú.....	15
2.1.3.3. BAIN – Plaça de Tetuan	17
2.2. Air Noise.....	19
Chapter 3. Ground Vibration Isolation	20
3.1. Required Isolation	20
3.2. Design of the System	21
3.3. Mathematical analysis	24
3.4. Construction and Assembly.....	29
Chapter 4. Air Vibration Isolation.....	30
4.1. Required Isolation	30
4.2. Design of the System	31
4.3. Construction and Assembly.....	32
Chapter 5. Allan Variance	34
5.1. Introduction	34
5.2. Accelerometer noise errors	34
5.2.1. Bias Instability.....	34
5.2.2. Velocity Random Walk.....	34
5.2.3. Rate Random Walk.....	35
5.2.4. Rate Ramp	35
5.2.5. Quantization Noise	35
5.2.6. Sinusoidal Noise	35
5.3. The AVAR Method	36
5.4. Interpretation of the ADEV plot.....	39
5.4.1. Bias Instability analysis.....	39
5.4.2. Velocity Random Walk analysis.....	39
5.4.3. Rate Random Walk analysis.....	40
5.4.4. Rate Ramp analysis.....	40
5.4.5. Quantization Noise analysis	41
5.4.6. Sinusoidal Noise analysis	41
Chapter 6. Test Specifications and Test Procedures	43
6.1. Test Design	43

6.2. Test Set-up.....	43
6.3. Test Organization and Schedule	44
6.4. Step-by-Step Procedure.....	44
Chapter 7. Results	46
7.1. Data Validation.....	46
7.2. Analysis of the noise figure	47
7.2.1. X-axis.....	48
7.2.2. Y-axis.....	49
7.2.3. Z-axis.....	50
7.3. Allan Deviation analysis	50
7.3.1. X-axis.....	51
7.3.2. Y-axis.....	52
7.3.3. Z-axis.....	53
7.3.4. Review of the results	54
Chapter 8. Conclusions.....	56
Bibliography.....	57
List of Figures.....	59
List of Tables.....	64
Annex 1. Code	65
1.1. Create Seismic Query	65
1.2. Analyse Seismic Signals	65
1.2.1. CGAR and VILA seismographs	65
1.2.2. BAIN seismograph.....	69
1.3. Analyse ground vibration isolation.....	73
1.3.1. Main program.....	73
1.3.2. Differential equations	75
1.4. ADXL355 Sensor	76
1.4.1. ADXL355 Class	76
1.4.2. Data acquisition	82
1.4.2.1. MainProgram.py.....	82
1.4.2.2. defined_functions.py	83
1.5. Allan Variance	85
1.5.1. mainAllan.m.....	86
1.5.2. Allan_AVAR_ADEV.m	87
1.5.3. AnalyseAllanDev.m.....	88
Annex 2. Seismic signals.....	90
2.1. CGAR seismograph	90
2.1.1. 1 st of February 2018.....	90
2.1.2. 11 th of February 2018.....	92
2.1.3. 21 st of February 2018.....	94
2.1.4. 1 st of March 2018	96
2.1.5. 11 th of March 2018.....	98
2.2. VILA seismograph.....	100
2.2.1. 1 st of February 2018.....	100
2.2.2. 11 th of February 2018.....	102
2.2.3. 15 th of February 2018.....	104
2.2.4. 6 th of March 2018.....	106
2.2.5. 11 th of March 2018.....	108
2.3. BAIN seismograph	110
3.2.1. 1 st of February 2018.....	110

3.2.2. 11 th of February 2018.....	112
3.2.3. 21 st of February 2018.....	114
3.2.4. 1 st of March 2018	116
3.2.5. 13 th of March 2018.....	118
Annex 3. ADXL355Z / Raspberry Pi 3B Connections	121

List of Abbreviations

ADC	Analog-Digital Converter
ADEV	Allan Deviation
AVAR	Allan Variance
FDSN	International Federation of Digital Seismographs Networks
FFT	Fast Fourier Transform
FS	Full Scale
FSR	Full Scale Range
HPF	High-Pass Filter
ICGC	Institut Cartogràfic I Geològic de Catalunya
IFFT	Inverse Fast Fourier Transform
I ² C	Inter-Integrated Circuit
LPF	Low-Pass Filter
LSB	Least Significant Bit
MEMS	Microelectromechanical system
ODR	Output Data Rate
PGA	Peak Ground Acceleration
RMS	Root Mean Square
RRW	Rate Random Walk
SND	Spectral Noise Density
SPI	Serial Peripheral Interface
SSID	Service Set Identifier
VRW	Velocity Random Walk
XML	Extensive Markup Language

Introduction

The purpose of this thesis is to study the noise characteristics of a MEMS accelerometer in order to determine if it can be used in a scientific experiment.

The experiment in question is designed to study Earth's atmosphere at altitudes between 100km and 250km and will be carried out by a swarm of spherical femtosatellites, weighting less than a 100g each. The femtosatellites will be deployed at 250km of altitude in different orbital planes around the Earth and will carry MEMS accelerometers as its main scientific payload. The objective of this accelerometers is to map Earth's atmosphere density between 100km and 250km, a region of space that is badly known because of the scarcity of satellites in this region.

The scientific experiment relies on very precise measurements of acceleration, so it requires ultralow noise, well calibrated accelerometers in order to achieve its purpose.

This thesis attempts to study the noise characteristics of a sensor in order to determine the viability of using it in a mission of these characteristics. The method used for this study is the Allan Variance, explained in detail in Chapter 5.

Computing the Allan Variance of an acceleration sensor requires the sensor to be completely still for long periods of time. In an ultralow noise sensor this can be a major challenge, because the vibrations caused by external sources can be greater than the accuracy of the sensor. If this was the case, the noise of the accelerometer would be obscured by the external noise, making the data useless for the Allan Variance method.

In order to overcome this problem, a system to attenuate external noise is proposed and built in Chapter 3 and Chapter 4. The system consists of a passive vibration isolation system composed by a stack of masses separated by springs and a floating vibration isolation table.

This thesis is organized as follows. In Chapter 1 the accelerometer that will be tested is described. Chapter 2 is devoted to a thorough analysis of the external noise sources. Chapter 3 and Chapter 4 describe the theoretical basis and the construction of a system to isolate the accelerometer from these external noise sources. Chapter 5 lays out the basic theory of the Allan Variance used to analyse inertial sensors such as accelerometers. Chapter 6 is explains the experiment set-up and methodology and the results are described in Chapter 7. Finally, conclusions are presented in Chapter 8.

Chapter 1. MEMS Sensor: ADXL355

1.1. Choosing the sensor

The key sensor of the scientific payload is an accelerometer, which has three main requirements. The first one is a limitation on the sensor size and mass, and it is set by the spacecraft's characteristics. The accelerometer must be as light and small as possible. Because the mission design is still in its very early stages, no exact limits are yet set, but based on the fact that the satellite will have a mass under 100 g, limit for the accelerometer mass is set at 5 g and 1000 mm³ (a 1 cm cube).

The second requirement is also imposed by the spacecraft properties, in this case the available power. The spacecraft works on a primary battery, and so has very limited available power. Hence, the accelerometer (as well as the rest of subsystems of the spacecraft) must consume the lowest power possible.

The third requirement is established by the scientific experiment. Because the experiment relies on very precise measurements of acceleration, the accelerometer must be ultralow noise in order for the data to be usable.

These three requirements leave MEMS accelerometers as the only option. They provide great noise characteristics and they meet the size, mass and power requirements. Piezoelectric sensors such as the ones used for seismography have excellent noise characteristics, better than MEMS accelerometers, but they consume more power and do not match the size and mass criteria.

The chosen sensor for this test was the Analog Devices ADXL355, for being the MEMS accelerometer with the lowest noise density available on the market.

1.2. MEMS Accelerometers

The ADXL355 is an ultralow noise, ultralow power, digital MEMS accelerometer manufactured by Analog Devices.

1.2.1. Principle of Operation

The ADXL355 is MEMS capacitive sensor. This means that the output voltage is proportional to the change in capacitance between two surfaces due to the change of distance between them.

A proof mass is placed between fixed parallel plates and is attached to a fixed frame using cantilever springs that allow it to move in the sensitive axis. The application of a force in this axis will cause the proof mass to move proportionally, changing the capacitance between the proof mass plates and the fixed plates

accordingly. As a result, the capacitance is proportional to the applied force. This concept is illustrated in Figure 1.1.

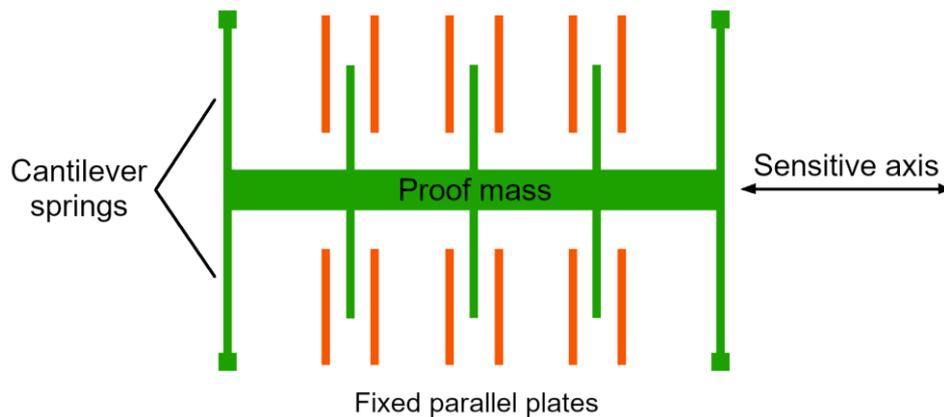


Figure 1.1. MEMS accelerometers principle of operation

The proof mass and fixed parallel plates are trapped between two parallel surfaces to limit the motion of the proof mass on the axis perpendicular to the plane while also trapping gas between the proof mass and the fixed parallel plates. This adds damping to the cantilever springs - proof mass system, which helps to reduce mechanical saturation of the sensor and attenuate the resonant frequency of the cantilever springs - proof mass system, while still maintaining a high sensitivity.

The resonant or natural frequency is defined by the combination of the stiffness of the cantilever springs and the mass of the proof mass. MEMS sensors are generally designed to have high resonant frequencies by using a lightweight proof mass and high stiffness cantilever springs. The natural frequency of the ADXL355 is at about 2.5kHz.

1.3. Characteristics

The ADXL355 provides user selectable range of ± 2.048 g, ± 4.096 g or ± 8.192 g. It weighs 0.26 grams and is 6mm \times 6mm \times 2.1mm in size, complying with the mass and size requirements explained in Section 1.1. It has three axis of measurement (X, Y and Z) and includes a temperature sensor.

The output is digital and communication with the sensor can be established through either SPI or I²C interfaces. The acceleration measurement output is 20-bit for all three axes and the temperature measurement is 12-bit.

The output data of the acceleration is filtered through LPF or HPF filters selectable by the user.

Relevant specifications can be found in Table 1.1, which correspond to the ± 2.048 g range, ambient temperature of 25 °C and $V_{\text{SUPPLY}} = 3.3$ V. Further information and specifications can be found on the ADXL355 datasheet [1]. The values of Velocity Random Walk for all axes are different in Table 1.1. and the datasheet due to errors in the datasheet.

1.3.1. Resolution and Noise Properties

As part of the scientific experiment of which ADXL355 will be part of, the sensor will measure the smallest accelerations possible. The smallest usable measure possible is determined by either the resolution or the noise of the sensor, although in most cases the noise is the limiting factor.

The resolution is the smallest value that the sensor can measure. The noise is a random variation in the output of the sensor. If this random variation surpasses the resolution of the sensor, it will determine the limit of the sensor. This is usually the case.

1.3.1.1. Resolution

The resolution of the sensor is defined by the number of bits of the ADC and the range of measurement following the formula:

$$\text{Resolution} = \frac{\text{Range}}{2^{\text{number of bits}}} \quad (1.1)$$

The smaller the range the better the resolution will be and the greater the number of bits of the ADC also the better the resolution. Because of this reason, the smallest range is used with the ADXL355, which is ± 2.048 g. The accelerometer ADC has 20 bits, providing a resolution of 3.09625 μg .

$$\text{Resolution} = \frac{\text{Range}}{2^{\text{number of bits}}} = \frac{2.048 - (-2.048)}{2^{20}} = 3.90625 \mu\text{g} \quad (1.2)$$

1.3.1.2. Noise

The spectral noise density is a measure of the distribution of noise through the frequency domain. For the ADXL355 and frequencies below 500 Hz this distribution can be considered flat with a value of 25 $\mu\text{g}/\sqrt{\text{Hz}}$ [1]. The exact graphic cannot be found on the datasheet, but it is similar to the graphic presented in Figure 1.1.

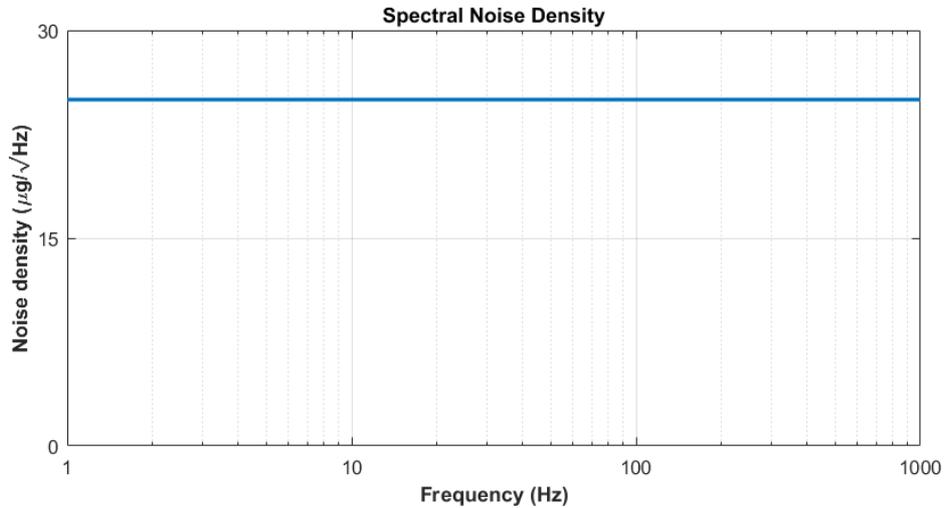


Figure 1.1. Graphic of flat SND analogous to SND on ADXL355

The SND graphic reflects that a bigger bandwidth will make the signal noisier, as more noise power is let through. The bandwidth is determined by the selected filters on ADXL355 Register 0x28. For ADXL355 the filter selection and output data rate selection is associated. For the scientific experiment, constant values of acceleration are the most interesting, so no high-pass filter is selected and the lowest low-pass filter available is selected. This LPF has the frequency cut-off at 0.977 Hz and -1.83 dB attenuation at cut-off.

The RMS noise is equivalent to the noise standard deviation and attends to the following equation:

$$\text{noise}_{\text{RMS}} = \sqrt{\int_{f_{\text{LOW}}}^{f_{\text{HIGH}}} \text{SND}(f)^2 df} \quad (1.3)$$

Where $\text{noise}_{\text{RMS}}$ is the RMS noise, f_{HIGH} is the upper frequency limit, f_{LOW} is the lower frequency limit and $\text{SND}(f)$ is the spectral noise density subject to the frequency. When the SND figure is flat, like in the case of ADXL355, Eq.(1.3) is simplified into Eq.(1.4).

$$\text{noise}_{\text{RMS}} = \text{SND} \sqrt{(f_{\text{HIGH}} - f_{\text{LOW}}) q_f} \quad (1.4)$$

Where SND is the constant value of spectral noise density and q_f is a factor depending on the “quality” of the filter. For a one pole filter this factor is 1.57 and for higher degree filter this values gets closer to 1, which would be an ideal filter. The low-pass filters of ADXL355 are programmable digital decimation filters with -1.83 dB attenuation at cut-off. Estimates of the q_f for digital decimation filter could not be found, so an assumption is made. The cut-off frequency of one pole

filter represents a -3dB attenuation, so the factor for the decimation filter of ADXL355 should be higher. An estimation is that the factor q_f is 2. The resulting RMS noise is $3.495 \cdot 10^{-5} \text{ g}$.

$$\text{noise}_{\text{RMS}} = 25 \cdot 10^{-6} \sqrt{(0.977 - 0) 2} = 3.495 \cdot 10^{-5} \text{ g} \quad (1.5)$$

As previously explained in this section, RMS noise coincides with the noise standard deviation. In order to determine the limitation on the smallest value that the sensor can measure due to the sensor internal noise, the peak noise is needed.

There is no analytic way to calculate the peak noise. The only way to determine it is by actually looking at the output signal of the sensor. Peak noise is the difference between the highest and lowest values of the noise signal divided by two. The problem with obtaining the peak noise is that the longer lasting the signal is the bigger the peak noise will be, as the signal will have more outliers.

However, an estimation can be made that peak noise is approximately 3.3 times greater than RMS noise. This approximation is correct for 99.9 % of points in the output signal [3]. The resulting peak noise is $1.153 \cdot 10^{-4} \text{ g}$.

The value of the peak noise is much greater than that of the resolution, making it the limiting factor on the measurement of the sensor.

Table 1.1. ADXL355 Specifications for ± 2.048 g range, $T_A = 25^\circ\text{C}$ and $V_{\text{SUPPLY}} = 3.3\text{V}$

Parameter	Test Conditions / Comments	Min	Typ	Max	Unit
Sensor Output Output Full Scale Range Sensitivity Scale factor Nonlinearity Cross Axis Sensitivity Sensitivity change due to temperature	X, Y, Z axes -40°C to +125°C	235520	256000 3.9 0.1 1 ± 0.01	276480	LSB/g $\mu\text{g}/\text{LSB}$ % FS % %/°C
0 g Offset 0 g output 0 g Offset vs Temperature 0 g repeatability	X, Y, Z axes -40°C to +125°C X, Y axes Z axis	-75 -0.15	± 25 ± 0.02 ± 3.5 ± 9	+75 +0.15	mg mg/°C mg mg
Noise Noise density Velocity Random Walk	X, Y, Z axes X, Y axes Z axis		25 6 8		$\mu\text{g}/\sqrt{\text{Hz}}$ mm/sec/ $\sqrt{\text{Hr}}$ mm/sec/ $\sqrt{\text{Hr}}$
ODR and Bandwidth Output Data Rate Low-Pass Filter Passband Frequency High-Pass Filter Passband Frequency When Enabled	User programmable, Register 0x28	3.906 0.977 $0.238 \times 10^{-3} \times$ ODR		4000 1000 $247 \times 10^{-3} \times$ ODR	Hz Hz Hz
Power Supply V_{SUPPLY}		2.25	2.5	3.6	V

I_{SUPPLY}	Measurement Mode	200	μA
Turn On Time	Standby Mode	21	μA
	Power-off to Standby Mode	<10	ms
	Standby Mode to Measurement Mode	<10	ms
Temperature Sensor			
Output at 25°C		1852	LSB
Scale Factor		-9.05	LSB/°C
Temperature			
Operating Temperature Range		-40	+125
			°C

Chapter 2. External Noise Sources

The Allan Variance is a method for studying the noise properties of accelerometers or gyroscopes that requires the sensor to be kept still for large periods of time. The reason why the sensor must be kept still is to measure the internal noise of the sensor. This is done when the sensor is no subject to accelerations. The constant acceleration of Earth's gravity is not avoidable, but all oscillating or unsteady accelerations must be eliminated when conducting the Allan Variance test.

For regular accelerometers (not ultralow noise), this can be easily done by just placing the sensor still on any non-moving surface. In this set up most accelerometers will not detect any acceleration (other than gravity), and the noise of the sensor can be detected and studied.

For ultralow noise accelerometers the concept is the same. It is necessary to leave the sensor still in order to detect and study the noise, but actually accomplishing this task can be challenge. The problem lies in the precision of the sensor.

The precision of an ultralow noise accelerometer can be so great, that when leaving the sensor static on a regular surface, it will measure small accelerations in the form of noise coming from external sources. If the sensor is placed in a standard table of a laboratory it will measure, for example, the vibrations produced by people walking on the laboratory or in adjacent rooms, that travel through the floor and table onto the sensor. It might also measure accelerations produced by air moving in the laboratory or vibrations of the building due to the impact of the wind. The amount of external noise captured by the accelerometer depends on the precision of the sensor. Sensors with greater precision will be affected by smaller noise sources.

If the Allan Variance is to be conducted properly this external noise sources have to be eliminated or attenuated to the highest degree possible. In order to decide how this will be achieved, the sources of noise need to be analysed.

These external noise sources can be classified in two main categories depending on the medium they propagate through: ground noise and air noise.

2.1. Ground Noise and Vibrations

Ground noise is considered as the vibrations travelling through the ground. They produce accelerations that can affect the measure of the sensor.

The origins of this vibrations are abundant, essentially anything moving in contact with the ground. People walking in the same room as the experiment or close by, traffic, moving trains, the impact of waves on the coast (if the experiment is conducted near the sea, which is the case), vibrations of the building caused by the wind, natural vibrations of the ground or seismic events are some examples.

Most of these sources of noise cannot be studied because they are highly unpredictable and each location has very unique characteristics as far as the amount of traffic close to the experiment, the amount of people, the type of soil, etc. An analysis of these factors would be extremely costly in both time and resources.

On the other hand, natural vibrations of the ground and seismic events can actually be properly studied. The most important data to know on the ground vibrations is the amplitude of the accelerations and their distribution through the frequency spectrum.

Nowadays there are large networks of seismic detectors used to detect and study seismic events. In the area of Barcelona there are several seismometers and most of them are property of ICGC (Institut Cartogràfic i Geològic de Catalunya), which makes the obtained data publicly available. These seismometers are either speedometers or accelerometers that measure ground vibration.

The experiment will be conducted in UPC - EETAC campus in Castelldefels, a coastal city surrounded by the Garraf mountains and part of the metropolitan area of Barcelona. It is assumed that the seismic qualities of the area will be a mixture of seismic properties of the nearby mountains, the seismic properties of the coastline and the seismic properties of a city. The ICGC does not have any seismograph in Castelldefels, but it does have seismographs situated in locations with the seismic properties mentioned. These seismographs are:

- CGAR: It is a seismograph located in the Garraf Massif. It will provide data on the seismic properties of the mountains close to Castelldefels. This seismograph is especially useful because it is very close to the university. It actually is so close that it can be seen from the campus. The output of this seismograph is velocity (m/s)
- VILA: It is a seismograph stationed on the sea bed in front of Vilanova de la Geltrú, Spain. It will provide information on the ground vibrations of the coastline, caused by waves and sea currents. The output of this seismograph is velocity (m/s).
- BAIN: It is an accelerometer situated in Plaça de Tetuan, Barcelona. Plaça de Tetuan is in a very densely populated area and there is a lot of traffic close by. This seismograph will provide information on the ground vibrations characteristic of the city, produced by human activity. The output of this seismograph is acceleration (m/s^2)



Figure 2.1. Map of seismographs (in red) and university (in green) locations and coordinates

2.1.1. Used Data

All seismic data presented in this document is obtained from *Institut Cartogràfic i Geològic de Catalunya (2000): Catalan Seismic Network. International Federation of Digital Seismograph Networks. Other/Seismic Network: doi: 10.7914/SN/CA.*

The data used for this analysis corresponds to five not consecutive days in the months of February and March of 2018. Data from different seismographs corresponds with different days because not all data for every date and every sensor is available. The results of this analysis are not dependent on the time of the measurement, so this is not an issue. The exact dates of the signals are presented in Table 2.1, Table 2.2 and Table 2.3. There are five signals evaluated for every seismograph and they each last 24 hours, starting at 00:00 of the corresponding date.

2.1.2. Analysis

The objective of this analysis is to determine the peak acceleration of the ground vibrations and their distribution through the frequency spectrum. To conduct this analysis, Matlab is used.

First data is obtained from ICGC's FDSN Web Services. The data is downloaded in two parts: the instrumental response part in XML format and the time signal part in miniSEED format.

The instrumental response contains information of the frequency response of the seismographs and the type of output they provide. The frequency response is presented in the form of the gain, poles and zeros of the transfer function of the sensor. The output of the seismographs is velocity for CGAR and VILA and acceleration for BAIN. Since it is the acceleration data what is being analysed,

the time derivative of the velocity signal of CGAR and VILA is computed, which yields the acceleration signal needed.

The time signal contains the output signal of the seismograph for each axis. The signal is provided in miniSEED format, a common format for seismic data. To transform the data into an easier to manipulate format, ReadMSEEDFast.m [4] is used. ReadMSEEDFast.m is Matlab routine available for download that converts miniSEED files into Matlab's structures and arrays, which are much easier to work with. The axes of measurement are HHE, HHN and HHZ, the first one is pointing East, the second is pointing North and the last is pointing upwards.

This time signal has units of counts/s for CGAR and VILA and counts/s² for BAIN. Information on the instrumental response is used to convert the data into m/s and m/s², but it is used differently for CGAR and VILA or BAIN.

In the case of BAIN, the time signal is converted into the frequency domain using the *fft* function on Matlab. This frequency signal is multiplied by the inverse of the frequency response of the seismograph to obtain the authentic frequency spectrum of the seismic signal. This authentic frequency spectrum is corrected to eliminate the alterations made by the frequency response of the seismograph. Then the signal is converted back into the time signal using *ifft*. In this way the true time signal and frequency spectrum of the seismograph is obtained.

For CGAR and VILA seismographs, the most adequate way of processing the data is very similar, only adding a time derivative when the true velocity signal is recovered. First the data is translated into the frequency spectrum using the FFT. Then this FFT is corrected by the frequency response of the sensor. This FFT is translated back into the time domain using the IFFT and then the time derivative is computed, resulting the acceleration data. The FFT of this acceleration data is also computed to view the distribution of accelerations through the frequency spectrum. This method was tried first, but although it is the most exact, a lot of information is lost in the process due to the translation from the time domain into the frequency domain, then back to the time domain and then again to the frequency domain. For this reason, an alternative method was devised.

Instead of using the complete frequency response of the seismograph, with a different amplitude at each frequency, a constant value is used. The frequency response of CGAR and VILA seismographs, although not completely, is quite flat. The result of treating the signal this way is not perfect, but it provides more information than the previous method. First the signal is divided by the constant amplitude, obtaining the velocity signal in m/s. Then this data is derived with respect to time, obtaining the acceleration. These are considered as the true values of acceleration, so the FFT of this data is calculated to obtain the frequency spectrum.

The code used for downloading the instrumental response and time signal, transforming in it to be easily used with Matlab and manipulating and interpreting the data can be found on Annex 1.

In the time signal, the shape of the vibrations can be observed and the peak ground acceleration can be obtained. The PGA is the maximum absolute

acceleration of the signal. This value is interesting because it will be used to calculate the necessary attenuation of the ground vibration in order to perform the Allan Variance.

From the frequency spectrum it will be visible at which frequencies the ground vibrations are concentrated and constant frequencies that affect the location.

2.1.3. Results

Here we present the results obtained from the seismic signal analysis of seismographs CGAR, VILA and BAIN. Only some of the graphics acquired from the seismic signals are shown here, the rest of the graphics are presented in Annex 2.

2.1.3.1. CGAR – Garraf

The time signal of the HHZ axis on the 1st of February of 2018 is presented in Figure 2.2. The vibrations detected by the seismograph are more or less constant during the day, but from around 05:00 to 13:00 there is a notable increase in the vibrations, as the amplitude doubles. This is possibly due to animal or human activity. The seismograph is situated in a mountainous area, where wildlife activity is common, so animals active during the day could contribute to the vibrations. The building in which the seismograph is situated is also used for other affairs, so it is very likely that there are people walking around the area, which can also contribute to the increase in the vibration amplitude.

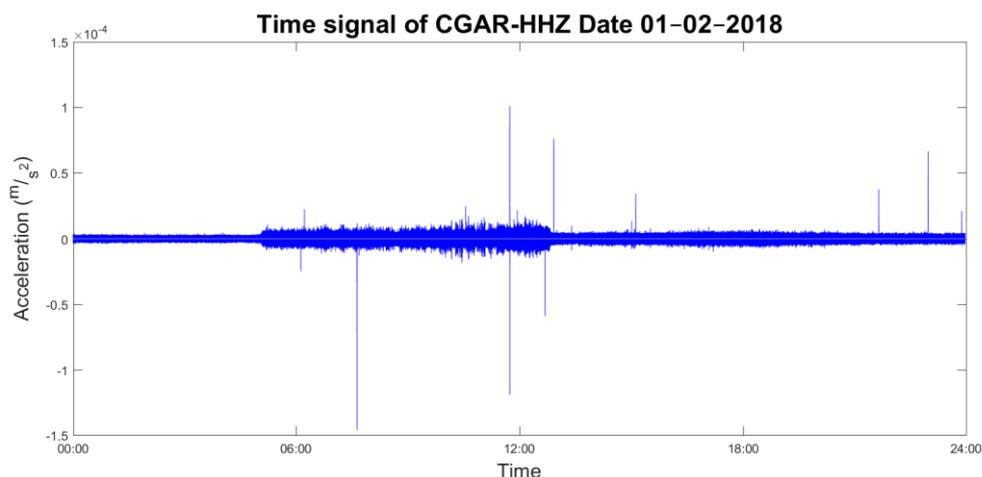


Figure 2.2. Seismic time signal for the HHZ axis of CGAR seismograph during the 1st of February 2018

Even though the persistent vibrations have an amplitude below $2 \cdot 10^{-5} \text{ m/s}^2$, there are some outliers that get up to $1.46 \cdot 10^{-4} \text{ m/s}^2$. The frequency of these outlier cannot be studied on the FFT presented below, because being outliers they

cannot be observed on a FFT of the complete signal; nevertheless, they can be studied individually. When each of the outliers of this graph is studied, no set patterns are observable on their frequencies but they are all on the range of 15 Hz to 100 Hz.

The normalized FFT of the HHZ axis on the 1st of February 2018 is presented in Figure 2.3. Below 3 Hz, there are virtually no accelerations on the ground. This is expected on the account that acceleration is proportional to the frequency squared, therefore there should be very small accelerations at low frequencies.

Over 3 Hz, the spectrum density rapidly increases up to the maximum, found at 7.06 Hz. This peak is very abrupt, being that both the ascent and descent wings are very pronounced. The descent does not reach the same values on the 3 Hz area, instead it lowers to 10^{-2} at 10 Hz, an order of magnitude greater. From there, the spectrum increases again, but this time much more calmly. Still, the values reached at the end of the spectrum are of the same magnitude of the maximum.

This signals acquired from HHZ axis during the 1st of February 2018 are an example of the vibrations captured by CGAR seismograph. The signals obtained from other axes and dates have similar tendencies for both the time signal and frequency spectrum, although there are many times at which the peak at 7.06 Hz is not present. Still, the rest of the behaviour is comparable. Also, the PGAs can vary in up to two orders of magnitude. This is because these maximums are not part of the regular behaviour of the signal, they correspond to outliers.

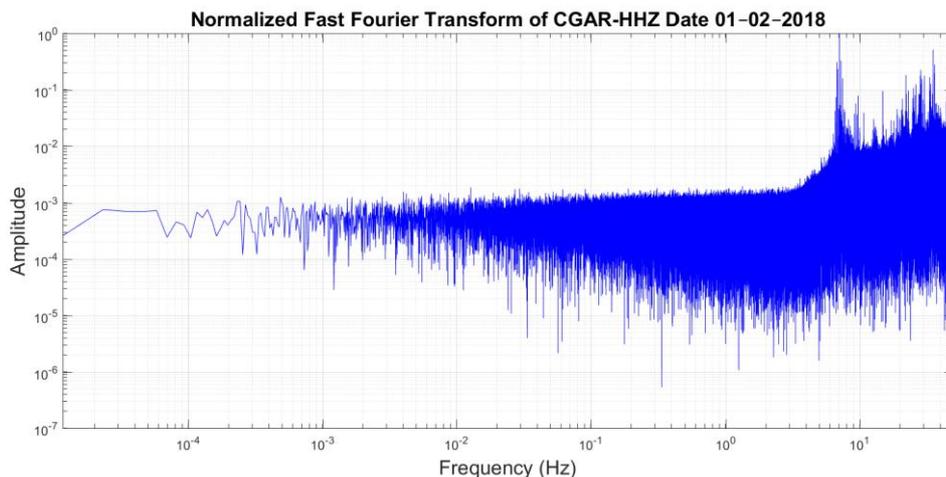


Figure 2.3. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 1st of February 2018

The maximum PGA for all axes and dates analysed is $1.49 \cdot 10^{-2} \text{m/s}^2$.

All the PGAs and frequencies of maximum concentration are found in Table 2.1. All graphics obtained from CGAR seismograph can be found at Annex 2.1.

Table 2.1. Peak ground accelerations and frequency of maximum concentration of vibrations at CGAR location

Date	HHE		HHN		HHZ	
	PGA (m/s ²)	f _{MAX} (Hz)	PGA (m/s ²)	f _{MAX} (Hz)	PGA (m/s ²)	f _{MAX} (Hz)
01/02/2018	1.05·10 ⁻⁴	28.53	1.51·10 ⁻⁴	07.06	1.46·10 ⁻⁴	07.06
11/02/2018	7.01·10 ⁻⁴	07.06	2.94·10 ⁻⁴	07.07	1.92·10 ⁻⁴	07.06
21/02/2018	6.79·10 ⁻³	33.81	4.22·10 ⁻³	07.05	3.49·10 ⁻³	07.05
01/03/2018	1.49·10 ⁻²	28.98	1.36·10 ⁻²	33.86	1.13·10 ⁻²	33.83
11/03/2018	1.42·10 ⁻⁴	33.83	2.12·10 ⁻⁴	33.83	1.43·10 ⁻⁴	33.83

2.1.3.2. VILA – Vilanova i la Geltrú

The time signal of the HHN axis on the 15th of February 2018 is presented in Figure 2.4. The accelerations captured by the seismograph do not follow a set pattern, they vary randomly through the day.

Excluding the outliers, which is mostly what is visible on the graph, the amplitude is always below 1·10⁻³ m/s². The PGA for this graph is 2.40·10⁻² m/s². The frequency of these outliers is also random, it does not follow any observable pattern. However, these frequencies are always above 10 Hz. These outliers cannot be studied on the FFT presented next because being outliers they are obscured by the rest of the signal. For that reason, they were studied individually one at a time.

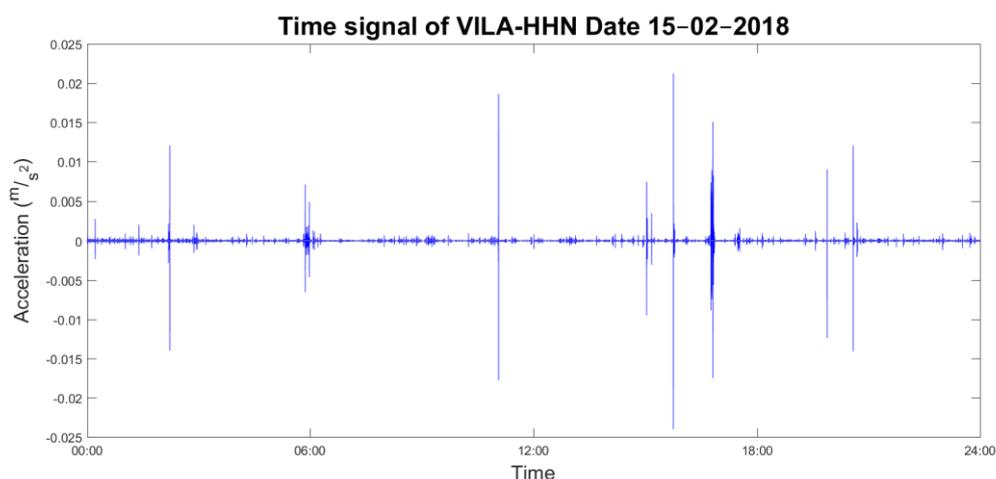


Figure 2.4. Seismic time signal for the HHN axis of VILA seismograph during the 15th of February 2018

The normalized FFT of the HHN axis on the 15th of February 2018 is presented in Figure 2.5. As in the previous seismograph, the Fourier transform shows that there are basically no accelerations at low frequencies due to the accelerations

being proportional to the frequency squared. Up to 2 Hz, the spectrum is very low and constant. From 2 Hz to 4 Hz the spectrum swiftly increases. At 4 Hz the slope of the increase reduces, while continuing being positive until 26 Hz, where a local maximum is found. Then, the spectrum amplitude decreases and increases again until at 49.86 Hz the absolute maximum is found, although both maxima are of the same order of magnitude.

The signals acquired from HHN axis during the 15th of February 2018 are an example of the vibrations captured by VILA seismograph. The signals obtained from other dates have similar qualities and tendencies. The PGA are the result of outliers of the seismic signal, not a predictable tendency, but they have very similar values for each axis. For axes HHE and HHN the order of magnitude of the PGA is the same for every day analysed and for HHZ axis only the PGAs from the 15th of February 2018 and 11th of March 2018 are of a different order of magnitude. That being one order smaller.

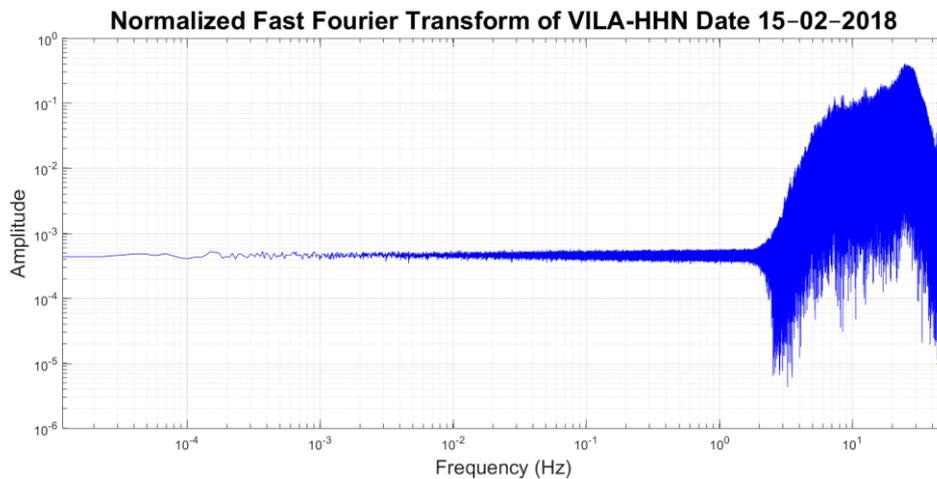


Figure 2.5. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 15th of February 2018

The maximum PGA for all axes and dates analysed is $9.93 \cdot 10^{-2} \text{ m/s}^2$.

All the PGAs and frequencies of maximum concentration are found in Table 2.2. All graphics obtained from VILA seismograph can be found at Annex 2.2.

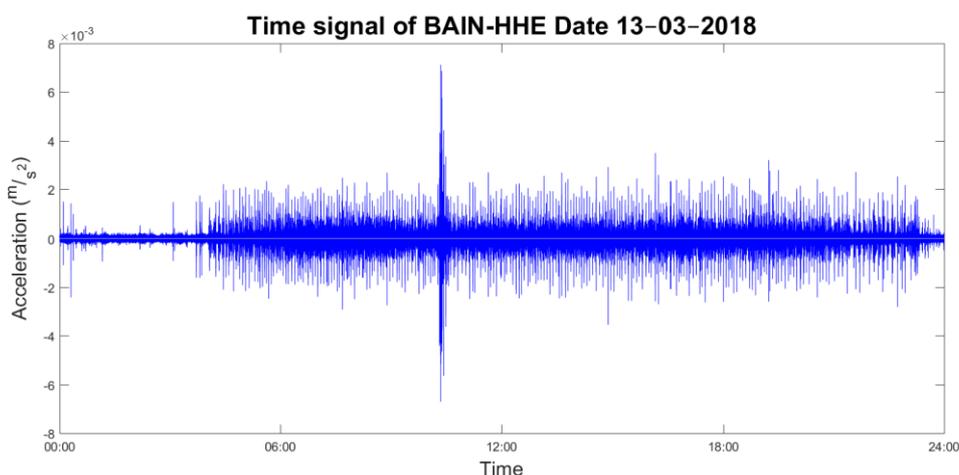
Table 2.2. Peak ground accelerations and frequency of maximum concentration of vibrations at VILA location

Date	HHE		HHN		HHZ	
	PGA (m/s ²)	f _{MAX} (Hz)	PGA (m/s ²)	f _{MAX} (Hz)	PGA (m/s ²)	f _{MAX} (Hz)
01/02/2018	$5.43 \cdot 10^{-2}$	28.35	$6.47 \cdot 10^{-2}$	26.80	$2.56 \cdot 10^{-2}$	49.85
11/02/2018	$4.82 \cdot 10^{-2}$	49.88	$9.93 \cdot 10^{-2}$	49.89	$2.16 \cdot 10^{-2}$	49.89
15/02/2018	$2.05 \cdot 10^{-2}$	49.84	$2.40 \cdot 10^{-2}$	49.86	$9.23 \cdot 10^{-3}$	49.85
06/03/2018	$2.36 \cdot 10^{-2}$	49.87	$3.74 \cdot 10^{-2}$	49.88	$1.94 \cdot 10^{-2}$	49.86
11/03/2018	$2.64 \cdot 10^{-2}$	49.86	$3.00 \cdot 10^{-2}$	49.87	$6.86 \cdot 10^{-3}$	49.86

2.1.3.3. BAIN – Praça de Tetuan

The time signal of the HHE axis on the 13th of March 2018 is presented in Figure 2.6. For this case, vibrations are greater from 04:00 to 23:00. This is due to human activity being concentrated in this time frame. In these hours there is much more traffic of both cars and people in the area, but the greatest contributor to this increase on these hours is the subway. The metro is only operative during those hours and it has a fixed schedule. It is the reason for such an abrupt change in the amplitude of vibrations.

Still, there are outliers. In this graphic there is only one, but in other axes or dates there might be many. The maximum outlier from this graphic is $7.12 \cdot 10^{-3}$ m/s². The frequencies of these outliers is random and ranges from 2Hz up to 100Hz. These outliers cannot be studied on the FFT presented next because being outliers, they are obscured by the rest of the signal. For that reason, they were studied individually one at a time.

**Figure 2.6.** Seismic time signal for the HHE axis of BAIN seismograph during the 13th of March 2018

The normalized FFT of the HHE axis on the 13th of March 2018 is presented in Figure 2.7. Below 4Hz, there are virtually no accelerations on the ground. This is expected as explained in section 2.1.3.1. From 4 Hz to 15 Hz the spectrum increases rapidly and after 15 Hz until 63.04 Hz it keeps increasing, but much more slowly. At 63.04 Hz it reaches the absolute maximum, from which the spectrum declines until the end of the spectrum.

The seismic signal shown in Figure 2.6 and Figure 2.7 is an example of a signal obtained from BAIN seismograph, but the rest of the signals analysed from BAIN sensor have similar tendencies. They all share the same frequency of maximum spectral density and the PGAs are all in the order of magnitude at 10^{-3} m/s².

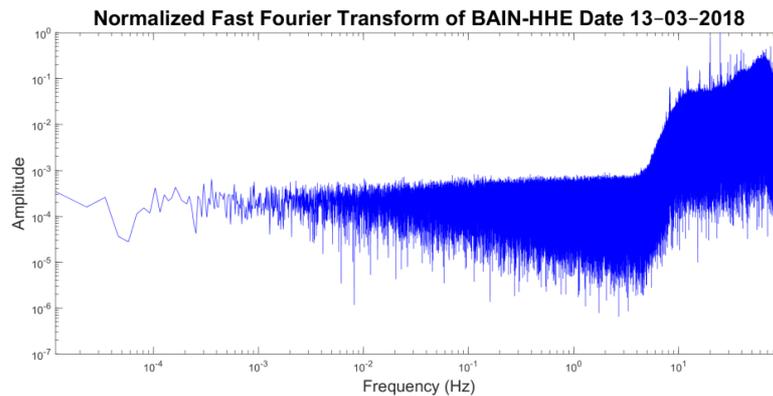


Figure 2.7. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 13th of March 2018

The maximum PGA for all axes and dates analysed is $9.79 \cdot 10^{-3}$ m/s².

All the PGAs and frequencies of maximum concentration are found in Table 2.3. All graphics obtained from BAIN seismograph can be found at Annex 2.3.

Table 2.3. Peak ground accelerations and frequency of maximum concentration of vibrations at BAIN location

Date	HHE		HHN		HHZ	
	PGA (m/s ²)	f _{MAX} (Hz)	PGA (m/s ²)	f _{MAX} (Hz)	PGA (m/s ²)	f _{MAX} (Hz)
01/02/2018	$5,35 \cdot 10^{-3}$	63.04	$5.30 \cdot 10^{-3}$	63.78	$4.71 \cdot 10^{-3}$	63.78
11/02/2018	$4.99 \cdot 10^{-3}$	39.64	$5.12 \cdot 10^{-3}$	39.64	$4.66 \cdot 10^{-3}$	39.64
21/02/2018	$3.42 \cdot 10^{-3}$	20.00	$4.02 \cdot 10^{-3}$	39.64	$4.30 \cdot 10^{-3}$	39.64
01/03/2018	$3.68 \cdot 10^{-3}$	63.78	$4.15 \cdot 10^{-3}$	63.78	$5.49 \cdot 10^{-3}$	63.78
13/03/2018	$7.12 \cdot 10^{-3}$	24.88	$9.79 \cdot 10^{-3}$	24.87	$8.01 \cdot 10^{-3}$	24.88

2.2. Air Noise

Air noise is considered as the vibrations that travel through the air. They produce accelerations that can affect the measure of the sensor.

The origins of this noise are varied, as they will be anything that makes the air surrounding the sensor move. This can be natural air currents, people or objects moving in the same room as the sensor, doors being opened or closed can particularly move a lot of air, even sound waves can induce vibrations on the sensor. Sound waves are changes in the air pressure and they can cause objects to oscillate.

These sources of noise are very difficult to study because they are notably unpredictable. This is especially true for human created air noise. The vibrations transmitted to the sensor by a person walking by depend on the speed that the person is moving at, the size of the person and the distance between the person and the sensor. The same is true for a door being opened or closed. These factors are not predictable, so there is no use in studying them deeply. For this analysis it is enough to acknowledge their existence in order to mitigate them.

Chapter 3. Ground Vibration Isolation

The goal of the ground isolation system is to minimize the effects of the external noise coming from the ground. The isolation required depends on the intensity of the vibrations and the expected noise of the sensor.

The objective of the attenuation is that the sensor never experiences an acceleration greater than its peak noise. If it is possible, it is desired to add a factor of 10 for protection. This means that the accelerometer should not experience higher than its peak noise divided by 10.

The peak noise of the accelerometer is $1.153 \cdot 10^{-4}$ g as calculated in section 1.3.1.2., therefore the isolation systems must reduce vibrations to below this value and if possible to less than $1.153 \cdot 10^{-5}$ g, adding a factor of 10 for protection.

3.1. Required Isolation

The PGAs presented in section 2.1.3. are the peak accelerations that are detected by the seismograph near the campus location. A compilation of the results is shown in Table 3.1.

Table 3.1. PGAs for each seismograph

Seismograph	PGA (m/s ²)
CGAR	$1.49 \cdot 10^{-2}$
VILA	$9.93 \cdot 10^{-2}$
BAIN	$9.79 \cdot 10^{-3}$

The maximum of these values is $9.93 \cdot 10^{-2}$ m/s² and it is taken as the reference value that must be attenuated to below $1.153 \cdot 10^{-4}$ g or $1.131 \cdot 10^{-3}$ m/s². Therefore, the required attenuation is:

$$\text{Attenuation} = \frac{9.93 \cdot 10^{-2} \text{ m/s}^2}{1.11532 \cdot 10^{-4} \text{ g} \cdot \frac{9.81 \text{ m/s}^2}{1 \text{ g}}} = 99.06 \approx 100 \quad (3.1)$$

The PGA must be attenuated by a factor of more than 100. These peaks of acceleration do not happen at a specific frequency, making the design of the ground vibration isolation more difficult. As a consequence, the ground vibration isolation system must be as strict as possible. The resonant frequencies of the system must be the lowest possible and the attenuation at higher frequencies

must be above a factor of 100 and if it is possible 1000, to add a factor of 10 for protection.

3.2. Design of the System

The requirements of the system in terms of isolation are very strict, so when thinking of ways to solve the task at hand, it is inevitable to think of how other vibration sensitive instruments have dealt with this problem. One particular instrument that comes to mind is the Laser Interferometer Gravitational-Wave Observatory (LIGO). LIGO is far more complex and precise instrument than the sensor used in this thesis, but the same principles can be used for this application.

The initial design of the system is inspired by [8]. A stack of masses separated by elastomer springs to provide passive vibration isolation.

The natural frequency of a spring–mass system is,

$$f = \sqrt{\frac{m}{k}} \quad (3.2)$$

where m is the mass and k the spring stiffness constant.

The difficulties presented by using the method described by [8] is that in order to achieve low natural frequencies, extremely massive objects must be used due to the large stiffness of elastomer springs. This rules out the use of elastomer springs. An alternative is to use metal springs instead of elastomer springs. Metal springs have a much lower spring constants, providing therefore a lower natural frequency for the same mass. Metal springs are also cheaper, easier to get a hold of and can be made to certain specifications, something which is much harder in elastomers. The use of metal springs is still problematic in some aspects. Metal springs provide very low damping properties. This means that any force on the system will create vibrations that will endure for large periods of time. Also, the natural frequency achieved using not very heavy masses, which are easier to find and transport and are below 10kg, is not small enough. For this reason, alternative or ways to improve this isolation stack must be found.

An initial alternative was to place the sensor floating in a container filled with fluid. In this way the water would “absorb” the vibrations and almost none would be transmitted to the sensor. The sensor would be set up in some sort of floating body that would also protect it from the water. The main concern of using this method is that the floating body would drift around the water and eventually crash with the walls of the water container, destroying the measurements of the sensor. A passive vibration isolation system using flotation that avoided this problem was devised by Liu, C. in 2012 [14] and is used as a primary source for the design of the vibration isolation presented in this work.

Liu, C. floating vibration isolation system is capable of reaching natural frequencies low enough for this thesis application, but the attenuation properties are not great, especially at high frequencies. For this reason, a combination of LIGO method and Liu, C. is used.

A diagram of Liu's floating vibration isolation system is shown in Figure 3.1 and a diagram of the combination of LIGO and Liu, C. vibration isolation systems; the system used in this thesis; is shown in Figure 3.2.

The main equations describing the behaviour of Liu, C. system are:

$$(m_f + m_{\text{added}})\ddot{h}' + C_D\dot{h}' + K_E h' = F_{\text{ex}} \quad (3.3)$$

$$m_{\text{added}} = \frac{\rho}{2} \left(\frac{WL^3}{6h} + \frac{WHL^2}{w} \right) \quad (3.4)$$

$$K_E = k + \rho gWL \left(1 + \frac{L}{2w} \right) \quad (3.5)$$

m_{added} is an important aspect of this vibration isolation system. It is the artificial mass added to the system by the displacement of water when the floating body is moving. m_f is the mass of the floating body including the sensor, C_D is the damping coefficient created by the friction with water. K_E is the effective stiffness of the system, composed by the spring's stiffness constant and the buoyancy force of having the floating body submerged in water. ρ is the water density, F_{ex} is an external force and h' is the displacement of the floating body from the equilibrium position.

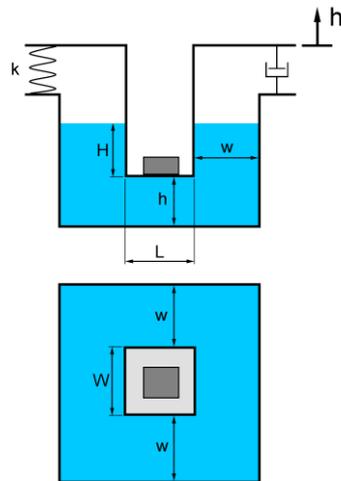


Figure 3.1. Floating vibration isolation system designed by Liu, C. viewed from the front and the top

The characteristic equation of motion of classic linear vibration isolation system is the following.

$$m\ddot{x} + c\dot{x} + kx = F_{ex} \quad (3.6)$$

Comparing with Eq.(3.7) with Eq.(3.3), it can be seen that Liu's floating vibration isolation system behaves like a classic linear vibration isolation system. Therefore, it can be analysed as such.

A change in the notation in Eq.(3.3) must be made to analyse it with the complete ground vibration isolation system.

$$h' = x_4 - x_3 \quad (3.7)$$

$$\dot{h}' = \dot{x}_4 - \dot{x}_3 \quad (3.8)$$

$$\ddot{h}' = \ddot{x}_4 - \ddot{x}_3 \quad (3.9)$$

The variable h' is changed for the difference between x_4 and x_3 . x_4 and x_3 are displacements from the equilibrium position and consequently, the result is the same.

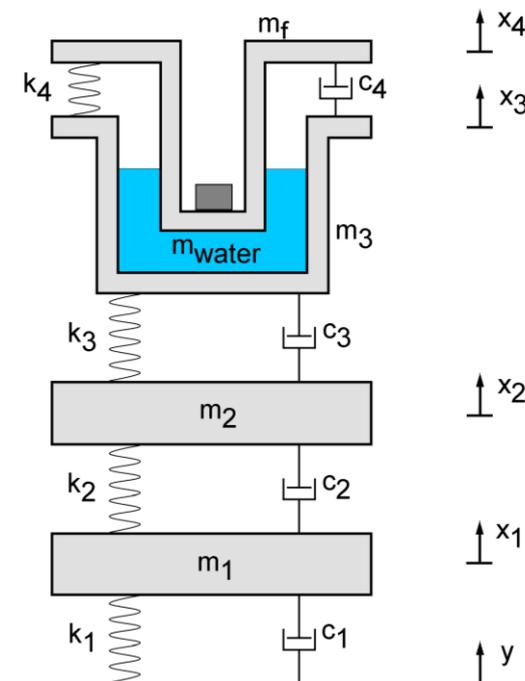


Figure 3.2. Ground Vibration Isolation System

The result of combining the vibration isolation systems of LIGO and Liu, C. is a four-degree of freedom system.

3.3. Mathematical analysis

To derive the equations of motion of this system, Lagrangian mechanics are used [16]. They can be described as

$$L = T - V \quad (3.10)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j \quad (3.11)$$

L is known as the Lagrangian, T is the total kinetic energy of the system, V is the total potential energy of the system, q_j are the generalized coordinates of the system, which for this case are x_1 , x_2 , x_3 and x_4 and Q_j are generalized forces, which are all the non-conservative forces of the system. The equilibrium position is considered as the origin of the generalized coordinates. Applying these equations to the ground vibration isolation system:

$$T = \frac{1}{2} m_1 \dot{x}_1^2 + \frac{1}{2} m_2 \dot{x}_2^2 + \frac{1}{2} (m_c + m_{\text{water}}) \dot{x}_3^2 + \frac{1}{2} (m_f + m_{\text{added}}) \dot{x}_4^2 \quad (3.12)$$

$$V = \frac{1}{2} k_1 (x_1 - y)^2 + \frac{1}{2} k_2 (x_2 - x_1)^2 + \frac{1}{2} k_3 (x_3 - x_2)^2 + \frac{1}{2} K_E (x_4 - x_3)^2 \quad (3.13)$$

$$L = T - V = \frac{1}{2} m_1 \dot{x}_1^2 + \frac{1}{2} m_2 \dot{x}_2^2 + \frac{1}{2} (m_c + m_{\text{water}}) \dot{x}_3^2 + \frac{1}{2} (m_f + m_{\text{added}}) \dot{x}_4^2 - \frac{1}{2} k_1 (x_1 - y)^2 - \frac{1}{2} k_2 (x_2 - x_1)^2 - \frac{1}{2} k_3 (x_3 - x_2)^2 - \frac{1}{2} K_E (x_4 - x_3)^2 \quad (3.14)$$

Developing Eq.(3.11) for each generalized coordinate:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_1} \right) = \frac{d}{dt} (m_1 \dot{x}_1) = m_1 \ddot{x}_1 \quad (3.15)$$

$$\frac{\partial L}{\partial x_1} = -k_1 (x_1 - y) + k_2 (x_2 - x_1) \quad (3.16)$$

$$Q_1 = -c_1(\dot{x}_1 - \dot{y}) - c_2(\dot{x}_1 - \dot{x}_2) \quad (3.17)$$

$$m_1\ddot{x}_1 + (c_1 + c_2)\dot{x}_1 - c_2\dot{x}_2 - c_1\dot{y} + (k_1 + k_2)x_1 - k_2x_2 - k_1y = 0 \quad (3.18)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_2}\right) = \frac{d}{dt}(m_2\dot{x}_2) = m_2\ddot{x}_2 \quad (3.19)$$

$$\frac{\partial L}{\partial x_2} = -k_2(x_2 - x_1) + k_3(x_3 - x_2) \quad (3.20)$$

$$Q_2 = -c_2(\dot{x}_2 - \dot{x}_1) - c_2(\dot{x}_2 - \dot{x}_3) \quad (3.21)$$

$$m_2\ddot{x}_2 - c_2\dot{x}_1 + (c_2 + c_3)\dot{x}_2 - c_3\dot{x}_3 - k_2x_1 + (k_2 + k_3)x_2 - k_3x_3 = 0 \quad (3.22)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_3}\right) = \frac{d}{dt}((m_c + m_{\text{water}})\dot{x}_3) = (m_c + m_{\text{water}})\ddot{x}_3 \quad (3.23)$$

$$\frac{\partial L}{\partial x_3} = -k_3(x_3 - x_2) + K_E(x_4 - x_3) \quad (3.24)$$

$$Q_3 = -c_3(\dot{x}_3 - \dot{x}_2) - (c_4 + C_D)(\dot{x}_3 - \dot{x}_4) \quad (3.25)$$

$$(m_c + m_{\text{water}})\ddot{x}_3 - c_3\dot{x}_2 + (c_3 + c_4 + C_D)\dot{x}_3 - (c_4 + C_D)\dot{x}_4 - k_3x_2 + (k_3 + K_E)x_3 - K_Ex_4 = 0 \quad (3.26)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_4}\right) = \frac{d}{dt}((m_f + m_{\text{added}})\dot{x}_4) = (m_f + m_{\text{added}})\ddot{x}_4 \quad (3.27)$$

$$\frac{\partial L}{\partial x_4} = -K_E(x_4 - x_3) \quad (3.28)$$

$$Q_4 = -(c_4 + C_D)(\dot{x}_4 - \dot{x}_3) \quad (3.29)$$

$$(m_f + m_{\text{added}})\ddot{x}_4 - (c_4 + C_D)\dot{x}_3 + (c_4 + C_D)\dot{x}_4 - K_Ex_3 + K_Ex_4 = 0 \quad (3.30)$$

The resulting equations of motion are:

$$m_1\ddot{x}_1 + (c_1 + c_2)\dot{x}_1 - c_2\dot{x}_2 - c_1\dot{y} + (k_1 + k_2)x_1 - k_2x_2 - k_1y = 0 \quad (3.31)$$

$$m_2\ddot{x}_2 - c_2\dot{x}_1 + (c_2 + c_3)\dot{x}_2 - c_3\dot{x}_3 - k_2x_1 + (k_2 + k_3)x_2 - k_3x_3 = 0 \quad (3.32)$$

$$(m_c + m_{\text{water}})\ddot{x}_3 - c_3\dot{x}_2 + (c_3 + c_4 + C_D)\dot{x}_3 - (c_4 + C_D)\dot{x}_4 - k_3x_2 + (k_3 + K_E)x_3 - K_Ex_4 = 0 \quad (3.33)$$

$$(m_f + m_{\text{added}})\ddot{x}_4 - (c_4 + C_D)\dot{x}_3 + (c_4 + C_D)\dot{x}_4 - K_Ex_3 + K_Ex_4 = 0 \quad (3.34)$$

The parameters of the design of the system are the masses and the springs stiffness constants. The concept for the design is simple, achieve the minimum

natural frequencies possible. Being a four-degree of freedom system, there are also four natural frequencies and in order to avoid increasing the vibrations of the system excessively at the natural frequencies, they must not coincide. Limitations also exist due to the availability of materials and the materials properties. For example, when choosing the metal springs, although it is beneficial to use springs with low stiffness, they should still be able to sustain the weight of the masses without permanently deforming.

The chosen parameters for the isolation system are shown in Table 3.2. and the chosen parameters for floating isolation part are shown in Table 3.3.

Table 3.2. Design parameters of ground vibration isolation system

$m_1 = 5.57 \text{ kg}$	$k_1 = 9920 \text{ N/m}$
$m_2 = 5.55 \text{ kg}$	$k_2 = 7036 \text{ N/m}$
$m_c = 2.22 \text{ kg}$	$k_3 = 4764 \text{ N/m}$
$m_{\text{water}} = 3.20 \text{ kg}$	$k_4 = 772 \text{ N/m}$
$m_f = 1.12 \text{ kg}$	

Table 3.3. Design parameters for floating isolation

$H = 3 \text{ cm}$	$h = 4 \text{ cm}$
$W = 13 \text{ cm}$	$w = 5 \text{ cm}$
$L = 13 \text{ cm}$	

As a result of these chosen parameters $m_{\text{added}} = 1.25 \text{ kg}$ and $K_E = 1153 \text{ N/m}$.

The other variables affecting the behaviour of the system the damping coefficients. In the previous equations, the damping force is proportional to the displacement velocity between the masses.

$$F_{\text{damping}} = c\dot{x} \quad (3.35)$$

This type of damping is known as viscous damping. It is easy to understand and use, but this is not the type of damping experienced by springs.

The damping on metal springs is known as hysteretic or structural damping. Hysteretic damping has the characteristic of being proportional to the displacement, not the velocity, it is independent of frequency and linked to the spring's stiffness [12]. The equation of motion of a single-degree of freedom hysteretically damped mass-spring system is:

$$m\ddot{x} + k^*x = 0 \quad (3.36)$$

$$k^* = k(1 + j\eta) \quad (3.37)$$

The springs stiffness and damping are bound together in the complex stiffness k^* . k is the spring stiffness and η the hysteric damping loss factor, which for steel springs like the ones used in the ground vibration isolation system is 0.1 [13].

The problem with Eq. (3.36) is that it cannot be used in time analysis of vibrations, it can only be used on frequency domain. For this reason, an equivalent viscous coefficient is used. This coefficient is described by [12]:

$$c_{eq} = \frac{k\eta}{w_n} \quad (3.38)$$

Where w_n is the natural frequency of the spring. Applying Eq. (3.38) to Eq. (3.36):

$$m\ddot{x} + c_{eq}\dot{x} + kx = 0 \quad (3.39)$$

This equation can now be used for time analysis and also frequency analysis. Which are performed using in Matlab using differential equations solver ode45.

The time analysis was used to determine the time required for the system to “settle” when it is excited by for example, placing the sensor inside. It was concluded that when setting the sensor in the structure, or manipulating the structure in any way for that matter, it should be left to rest untouched for 24 before the measurements of the sensor can be considered useful.

The results of the frequency analysis are shown in Figure 3.3. The resulting ground vibration isolation system has four natural frequencies: 2.2Hz, 4.13 Hz, 6.87 Hz and 10.09 Hz but only three are visible on the graphic due to their combined effects. The most important of the natural frequencies is 2.2 Hz, which causes the vibrations at said frequency to increase by a factor of 13. This peak corresponds to the floating vibration isolation stage. The signal starts being attenuated at 4.5 Hz and it quickly decays. At 13 Hz, the minimum necessary attenuation is accomplished and at 17.5 Hz the desired attenuation is reached. The attenuation reaches a minimum of 10^{-5} at 38 Hz and then it increases back again until it stabilishes a constant value of $3 \cdot 10^{-3}$. This effect is created by the floating vibration isolation step [14].

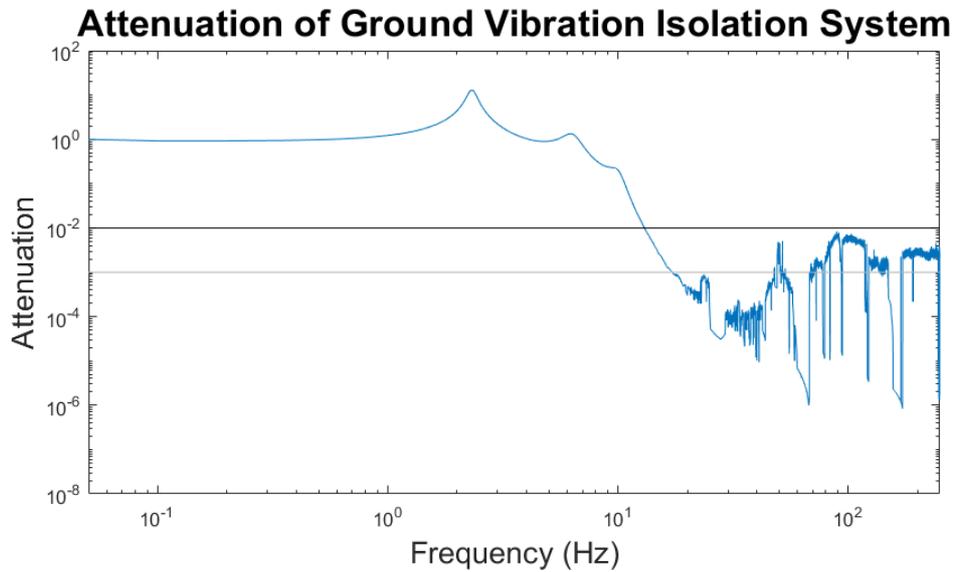


Figure 3.3. Attenuation of Ground Vibration Isolation System. The attenuation of the system is in blue, the minimum necessary attenuation is in black and the desired attenuation is in grey

Although the minimum attenuation objective is met, a factor of 10 for protection is not reached for high frequencies. To overcome this problem additional vibration isolation equipment is employed. The ground vibration isolation system is mounted on an optical table, which will provide more isolation. The specifications of the optical table are unknown, but they are made of very heavy granite stones, so it is assumed that the natural frequency is very low and the high frequencies are attenuated greatly.

3.4. Construction and Assembly

Here is explained the followed process when assembling the ground vibration isolation system and the materials used.

The first part of the assembly was to build the floating container and body of the floating isolation. A representation of the system is shown in Figure 3.4. The floating container function is to hold water inside and in this water the floating body will be submerged, for that reason, both need to be watertight. In order to achieve this, acrylic glass panels are used. Acrylic glass is very machinable which allows it to be cut to precise specifications. The cut was done using the CNC milling machine at Tinkerers Fab Lab, situated in the EETAC campus. The pieces were glued together using an adhesive specific for this material.

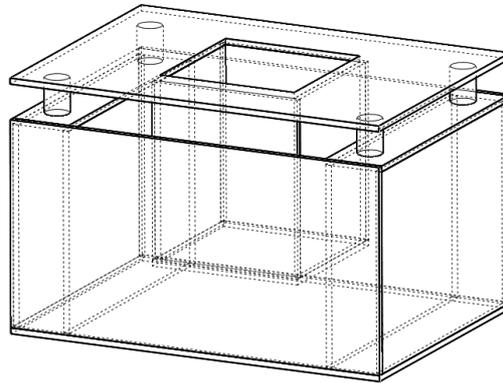


Figure 3.4. Floating vibration isolation system

For masses m_1 and m_2 ceramic tiles were used. Two for each mass were necessary to meet the desired mass.

The metal springs were bought according to the specifications. For each “step” of the ground isolation system, four springs are necessary, so the stiffness constant of each spring is the stiffness of the design divided by four.

Chapter 4. Air Vibration Isolation

As shown in section 2.2 there is no quantitative information on the amplitude, frequency or effects of air-induced noise on the sensor. The solution, therefore, must encompass all possible sources, amplitudes and frequencies.

The ideal solution, which would completely eliminate the air noise would be to conduct the Allan Variance in vacuum. The absence of air would eradicate any noise that travels the air because of the lack of a medium to propagate through. Unfortunately, this option was not available and even if this was the case, it would still pose some challenges. Specially because given the size and shape of vacuum chambers, the ground vibration isolation system would have to be specifically designed to fit and operate in vacuum.

With the option of using a vacuum chamber ruled out, the next best option would be, instead of isolating the sensor from external sources of noise, to actually eliminate these sources. This option was not available for ground noise because in that case some of the sources cannot be controlled, like seismic noise or noise caused by people external to the experiment. However, in the case of air vibration these sources are much more controllable.

Most of the causes of air noise are caused by human activities in the proximities of the sensor, therefore the access to the room or laboratory of the experiment will be restricted. While the sensor is making measurements, no people will be allowed to enter the room of the experiment. This will eliminate movement on the air caused by people walking by. Also, the room will be completely closed at all times to avoid air currents. In this way, all air noise will be cancelled.

These restrictions on human activities would be nice to have, but rooms and laboratories are shared spaces and even though no everybody has access to them, their use cannot be blocked to people that also require to use them. Furthermore, the experiment could last several days and there is no way to actually know if someone entered or did not enter the room. The solution to this problem is to add a layer of protection from the surrounding air to the sensor.

The previous considerations will still be taken into account and the study will be conducted in a room where it is not expected to be anyone present while the experiment is running.

The design of the protection layer for the sensor is laid out in the following sections.

4.1. Required Isolation

This layer of protection must meet the following criteria regarding the needed isolation:

- It must isolate the sensor from the surrounding air.

- It must be sealed in order to not let air through, blocking air movement.

Other than the isolation criteria, it must also meet certain requirements that allow it to be used easily and effectively. These conditions are:

- It must be big enough to protect the sensor and the ground vibration isolation system.
- It must be able to be opened and closed easily.
- It must allow to connect cables from inside to the outside like electrical connections.

4.2. Design of the System

The proposed design consists of a square prism made of wood with a missing side, much like a box. The missing side is the bottom face of the prism and the top side of the prism is detachable to allow quick and easy connections with the sensor. Also at the top, there is a hole for allowing cables to pass through. The hole is padded with viscoelastic foam to seal it and to dissipate vibrations affecting the cables from the outside.

At the bottom, the sides that make contact with the floor are covered with foam to allow the sides to make complete contact with it and ensure that the air inside is enclosed.

The box is composed of five wooden panels. Two of the side panels are 64cm in height and 58cm in width, the other two side panels are 64cm in height and 56cm in width and the top panel is 58cm in height and width. The hole in the top panels is 3cm in diameter. All the panels have 1cm thickness.

The dimensions of the box when assembled are 65cm in height and 58cm in width. An illustration is shown in Figure 4.1.

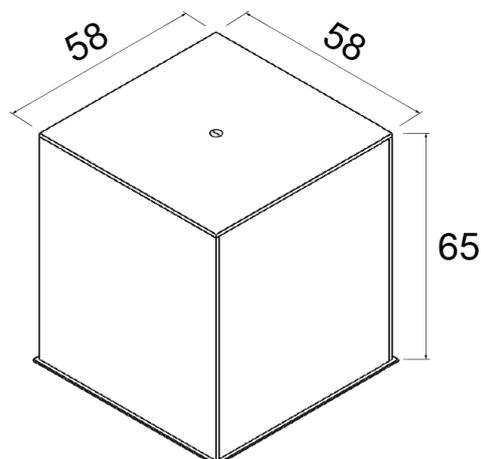


Figure 4.1. Design of the air isolation system. Units are in centimetres.

The padding in the hole of the top panel is made of viscoelastic cushioning. This cushioning has a small incision big enough to allow cables through, but still small enough to compress the cables and leave no gap.

4.3. Construction and Assembly

Here is explained the process followed to build the air isolation system. First, the five wooden panels were cut out from a single piece of wood. Next the five panels were assembled together. Then the hole was drilled onto the top panel and finally the padding for the bottom and the top hole was added.

In the construction of the air isolation system, pine laminated plywood was used for producing the box panels. Each panel was designed in SolidWorks and was cut using the CNC milling machine at Tinkerers Fab Lab. The wood panels were designed to have a tight fit in order to not leave gaps on the joints so the CNC milling machine provided the need precision for cutting the panels.

In the assembly of the panels, 90° metal framing angles and 3mm screws were used to fix the structure. The top panel is detachable and there is no padding between the side panels and the top panel. For that reason, the assembly of the “box” was started attaching the side panels to top panel, before joining the side panels between themselves. This allowed the side panels to have completely flush contact with the top panel, effectively ensuring that when the top panel is not secured to the side panels there are no gaps left. An illustration of this part of the assembly is shown in Figure 4.2. When the assembly of the wood panels was completed, the top panel was detached from the side panels.



Figure 4.2. Air vibration isolation system assembly. Attaching side panels to top panel.

Even though the wood panels were cut out using the a CNC milling machine, the cut was not perfect. The panels were not completely rectangular because the angles between the sides were not exactly 90° and the dimensions were slightly different from one panel to another. For that reason, starting the assembly attaching the side panels to the top panel became especially useful. The top edges of the side panels became completely flat and levelled while on the bottom edges there were some imperfections. This was the expected result and for that reason the foam was attached to the bottom edge of the side panels. Also as part of the imperfections, there were some air gaps between the edges of the side panels. These gaps were filled with hot glue, eliminating the possibility of air flowing into the air isolation system and adding structural integrity to the frame.

The hole on the top panel was done using a drill with a hole saw attachment, 25mm in diameter.

The padding used for the top hole was two pieces of viscoelastic foam. The first was a square 14cm in width and 2cm in thickness and it was glued on the inside of the air isolation system. The second piece was a circular, 4cm in diameter and 2cm in thickness and it was placed inside the drilled hole. This last piece was 4cm in diameter, bigger than the hole which was 2.7cm in diameter. In this way the padding would always compress the cables passing through, ensure that no air gaps are left. Incisions were made with sharp blade on both pieces of viscoelastic foam. These incisions were just big enough to let a microUSB connector to pass through and the incisions on each piece of foam were perpendicular to each other. Illustrations of the padding is shown in Figure 4.3.

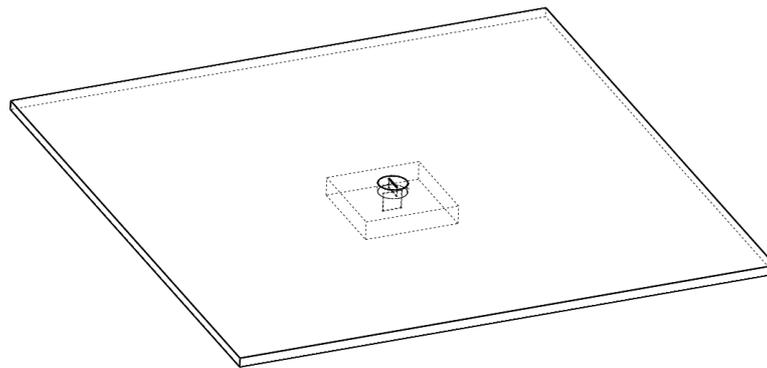


Figure 4.3. Illustration of air vibration isolation system top panel with the viscoelastic padding.

The bottom edges of the structure are padded with adhesive foam. This allows the edges to have complete contact with the floor without gaps. The weight of the structure compresses the foam closing gaps produced by imperfections in the bottom edges of the panels. These small imperfections are small differences on the length of the panels. The foam is about half centimetre in thickness.

Chapter 5. Allan Variance

5.1. Introduction

The Allan Variance is a method commonly used for analysing the noise characteristics of inertial sensors such as gyroscopes and accelerometers.

The AVAR method was initially devised by David Allan in 1966 [17] for studying the frequency and phase instability of precision oscillators like atomic clocks and crystal oscillators. It has since then been adopted as one of the preferred methods for analysing inertial sensors and several modifications have been introduced. Although it was not designed for this purpose, the close analogies between precision oscillators and inertial sensors allowed the AVAR to be introduced as a noise analysis method for gyroscopes and accelerometers by the IEEE community [18] [19].

The AVAR method is used to represent the intrinsic noise in a system as function of averaging time and the results are related to some of the most important noise characteristics in accelerometers, which are presented in the following section. The Allan Variance analyses a sequence of data in the time domain and as a result allows to identify and quantify the underlying causes of the noise.

5.2. Accelerometer noise errors

The most prevailing noise errors in accelerometers and their effects on the output signal are described in the following subsections. The most relevant of noise errors are bias instability, velocity random walk, rate random walk, rate ramp, quantization noise and sinusoidal noise.

5.2.1. Bias Instability

The bias instability indicates how the value of bias changes with time. The source of this noise are the electronics components susceptible to random flickering [19]. For this reason, it is also referred as rate flicker noise [20]. The units of the bias instability are m/s^2 or m/hr^2 .

5.2.2. Velocity Random Walk

The result of integrating the output of an accelerometer over time is the velocity. The VRW is a measure of the error that appears on the velocity when it is calculated in this way. The longer the integration time is, the greater the error will be. For this reason, it can severely limit the measurement of velocity over large periods of time [22].

VRW is usually measured in $\text{m/s}/\sqrt{\text{s}}$ or $\text{m/s}/\sqrt{\text{hr}}$ and can be interpreted in the following way: if a sensor has $\text{VRW} = 1 \text{ m/s}/\sqrt{\text{hr}}$, the standard deviation of the error in velocity after 5 hours will be $1 \text{ m/s}/\sqrt{\text{hr}} \cdot \sqrt{5} = 2.24 \text{ m/s}$.

The main sources of this error are high frequency noise terms with correlations times much shorter than the sampling time. [19].

5.2.3. Rate Random Walk

Rate random walk is a white jerk noise that appears as a random drift rate. It is a long term, very low frequency phenomenon and their causes are unknown. The units of this error are $\text{m/s}^2/\sqrt{\text{s}}$ or $\text{m/s}^2/\sqrt{\text{hr}}$.

The RRW is interpreted similar to VRW. If an accelerometer has $\text{RRW} = 1 \text{ m/s}^2/\sqrt{\text{hr}}$, the expected standard deviation of the error in acceleration after 5 hours will be $1 \text{ m/s}^2/\sqrt{\text{hr}} \cdot \sqrt{5} = 2.24 \text{ m/s}^2$.

5.2.4. Rate Ramp

In contrast with the previous errors, over large periods of time, this error becomes more of a deterministic error rather than a random noise. It has units of $\text{m/s}^2/\text{s}$ or $\text{m/s}^2/\text{Hr}$ and the error created is defined as:

$$\text{Error} = R t \quad (5.1)$$

Where R is the rate ramp coefficient and t is the time interval.

5.2.5. Quantization Noise

It is the noise caused by transforming an analog signal into a digital signal. It is related to the resolution of the analog-to-digital converter. The greater the ADC resolution is, the smaller the quantization noise will be. It is expressed in units of m/s^2 or m/hr^2 .

5.2.6. Sinusoidal Noise

This noise usually takes the shape of a decaying sinusoidal function in the AVAR plot and it is characterized by one or more frequencies and amplitude. The cause of this noise can sometimes be attributed to periodic environmental changes.

5.3. The AVAR Method

In order to obtain useful results, data must be sampled for large periods of time, usually several days, and the accelerometer must be kept still. The objective of keeping the sensor still is that the only variations in the sensor output, can be attributed to the noise of the sensor, not external forces. Only the variations of the output affect the resulting AVAR, therefore constant accelerations experienced by the sensor such as gravity do not change the outcome of the AVAR. Neither do other factors that remain constant like the orientation of the sensor.

The Allan Variance can be computed in two different ways: either using the acceleration, which is the output of the sensor, or the velocity obtained from integrating the output acceleration. Still, both are a function of cluster or averaging time, so the first step is to compute the cluster times and separate the data into clusters.

A cluster is bundle of consecutive data with n data points. Assuming the period between each data point is t_0 , the duration of each cluster is $T = n \cdot t_0$. The limit to the size of the clusters is $N/2 - 1$, being N the total number of data points, which would divide the data set into only two clusters. A visual representation of the clusters is shown in Figure 5.1.

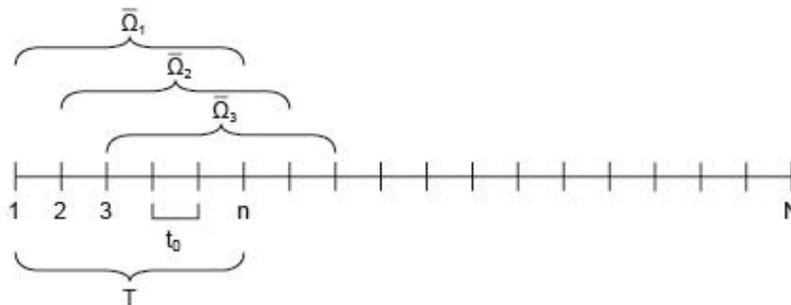


Figure 5.1. Overlapping clusters of Allan Variance

Using the output acceleration, the cluster average is

$$\bar{\Omega}_k(T) = \frac{1}{T} \int_{t_k}^{t_k+T} \Omega(t) dt \quad (5.2)$$

Where $\Omega(t)$ is the instantaneous acceleration and $\bar{\Omega}_k$ is the average of the cluster with T duration starting at the k data point. The average of the next cluster is therefore

$$\bar{\Omega}_{k+1}(T) = \frac{1}{T} \int_{t_{k+1}}^{t_{k+1}+T} \Omega(t) dt \quad (5.3)$$

The difference between the averages of adjacent clusters is

$$\xi_k = \bar{\Omega}_{k+1} - \bar{\Omega}_k \quad (5.4)$$

The resulting Allan Variance is

$$\sigma^2(T) = \frac{1}{2(N-2n)} \sum_{k=1}^{N-2n} \xi_k^2 \quad (5.5)$$

The AVAR can also be defined using the velocity, which is the way it is calculated in this thesis. The velocity is obtained integrating the output acceleration,

$$\theta(t) = \int^t \Omega(t) dt \quad (5.6)$$

where $\Omega(t)$ is the instantaneous acceleration and $\theta(t)$ is the resulting velocity. The acceleration is discrete in time and so is the velocity. The sample times can be defined as $t_k = kt_0$, for this reason velocity is defined as $\theta_k = \theta(kt_0)$.

The velocity is divided into clusters and the analysis used is analogous to the previous for acceleration. However, the clusters are defined in the same way, but they are separated differently. A visual representation is shown in Figure 5.2.

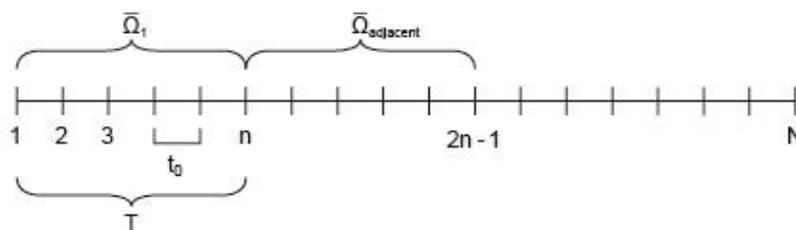


Figure 5.2. Non-overlapping clusters of Allan Variance

The cluster average is now defined as

$$\bar{\Omega}_k(T) = \frac{\theta_{k+n} - \theta_k}{T} \quad (5.7)$$

the cluster average of the adjacent cluster average is

$$\bar{\Omega}_{\text{adjacent}}(T) = \frac{\theta_{k+2n} - \theta_{k+n}}{T} \quad (5.8)$$

The AVAR is defined as

$$\sigma^2(T) = \frac{1}{2} \langle (\bar{\Omega}_{\text{adjacent}} - \bar{\Omega}_k)^2 \rangle \quad (5.9)$$

Where $\langle \quad \rangle$ is the ensemble average. The resulting AVAR is

$$\sigma^2(T) = \frac{1}{2T^2(N-2n)} \sum_{k=1}^{N-2n} (\theta_{k+2n} - 2\theta_{k+n} + \theta_k)^2 \quad (5.10)$$

The Allan Variance is now defined as function of cluster time. Using different cluster times allows to observe random processes with different correlation times. To identify these processes, the AVAR for different cluster sizes is computed. n is the number of data points in a cluster. The most common way to observe the random processes in inertial sensors is to plot a log-log graph of the Allan Deviation against the cluster time T . The Allan Deviation is simply the square root of the Allan Variance. The shape of this plot for inertial sensors is shown in Figure 5.3.

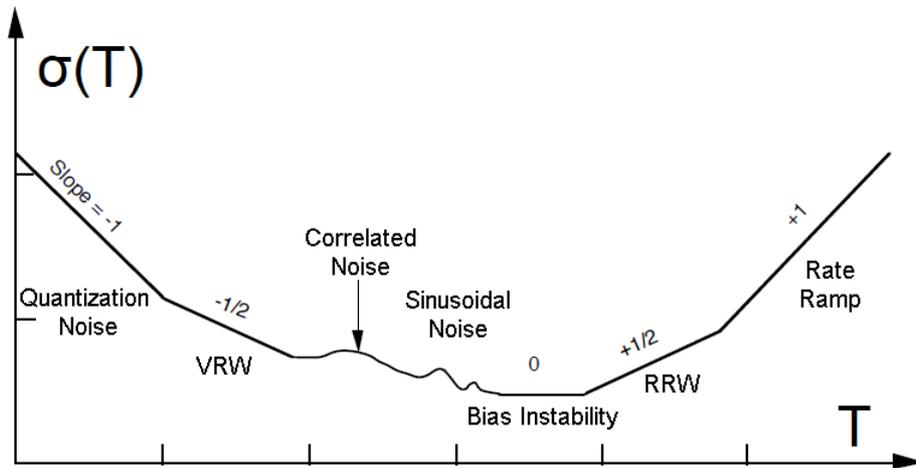


Figure 5.3. Typical ADEV plot for inertial sensors

5.4. Interpretation of the ADEV plot

The Allan Deviation log-log plot is extraordinarily useful because allows to easily quantify the noise errors affecting inertial sensors described in section 5.2.

5.4.1. Bias Instability analysis

The bias instability is found at the point where the slope of the Allan Deviation plot is zero as shown in Figure 5.4. This is usually at the lowest point of the plot. The value at said point of the plot is

$$\sqrt{\frac{2 \ln 2}{\pi}} B \approx 0.664 B \quad (5.11)$$

Where B is the bias instability coefficient.

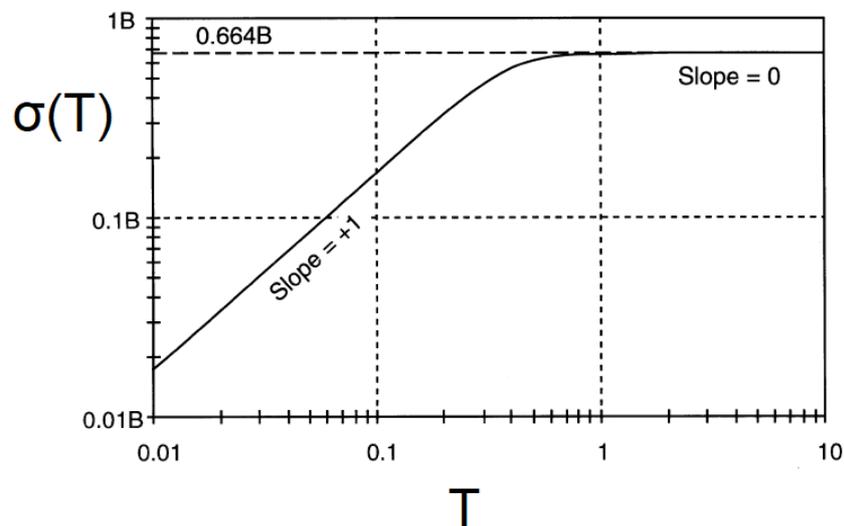


Figure 5.4. Bias instability on the ADEV plot

5.4.2. Velocity Random Walk analysis

The VVR can be observed on the ADEV graph where the slope is $-1/2$ as shown in Figure 5.5. The VVR coefficient N can be directly obtained, reading the value of the slope $-1/2$ at $T = 1$.

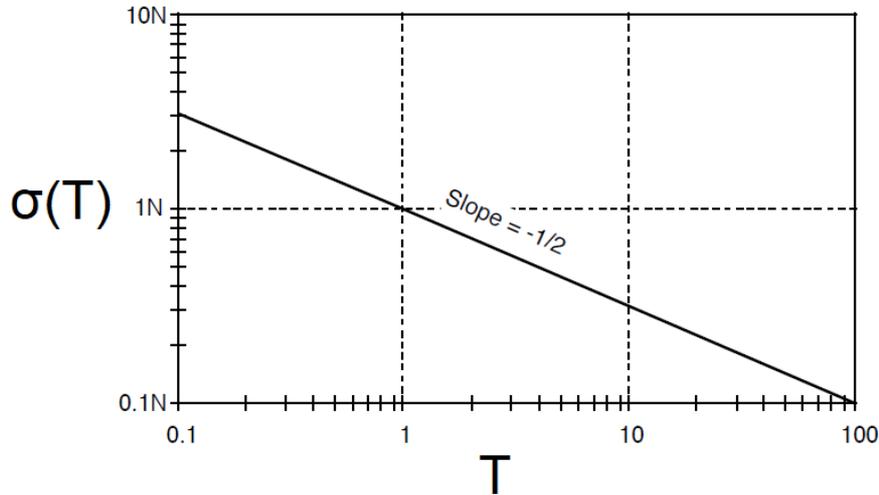


Figure 5.5. Velocity Random Walk on the ADEV plot

5.4.3. Rate Random Walk analysis

The RRW is presented in the ADEV on the +1/2 slope. The magnitude of the RRW coefficient can be read at $T = 3$ of the slope. Figure 5.6. shows an example of the RRW on an ADEV plot.

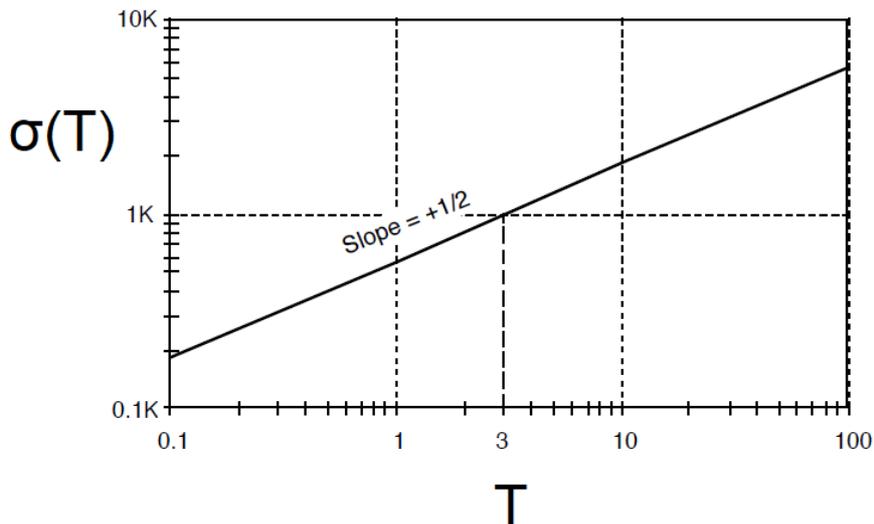


Figure 5.6. Rate Random Walk on the ADEV plot

5.4.4. Rate Ramp analysis

The Rate Ramp appear on the +1 slope of the ADEV plot. The value of the Rate Ramp coefficient is read at $T = \sqrt{2}$. Figure 5.7. shows an example of the Rate Ramp on an ADEV plot.

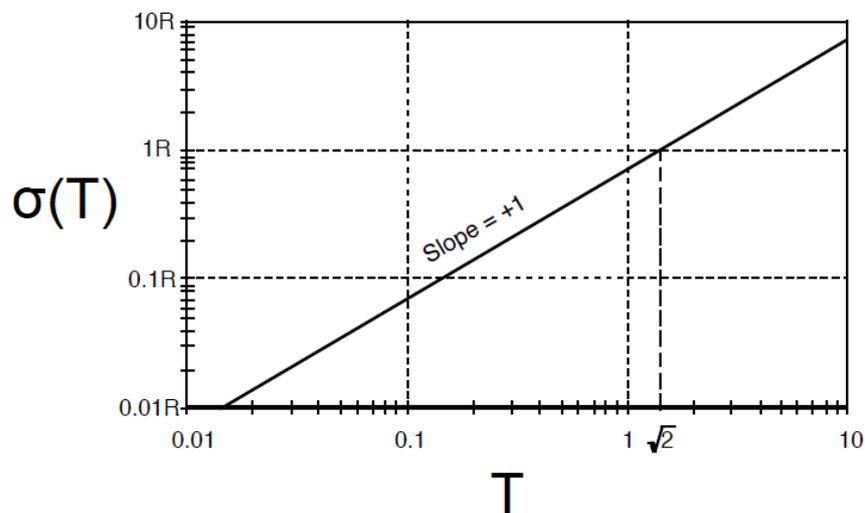


Figure 5.7. Rate Ramp on the ADEV plot

5.4.5. Quantization Noise analysis

The Quantization Noise can be observed on the ADEV graph where the slope is -1 as shown in Figure 5.8. The quantization noise coefficient Q can be directly obtained, reading the value of the -1 slope at $T = \sqrt{3}$.

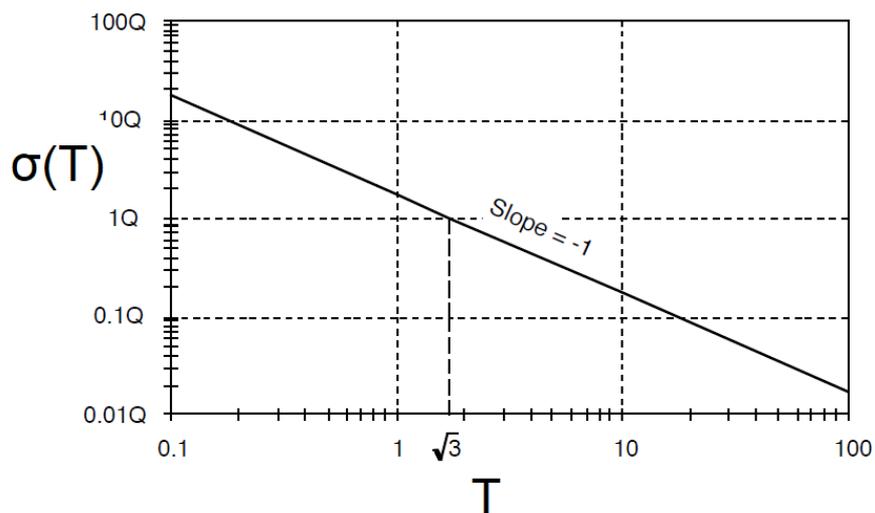


Figure 5.8. Quantization Noise on the ADEV plot

5.4.6. Sinusoidal Noise analysis

The analysis of Sinusoidal Noise is not as easy as the previous noise errors. The estimation of the Sinusoidal Noise requires the observation of several peaks.

Also, the amplitudes of consecutive peaks decrease rapidly and can be masked by other peaks or frequencies [19].

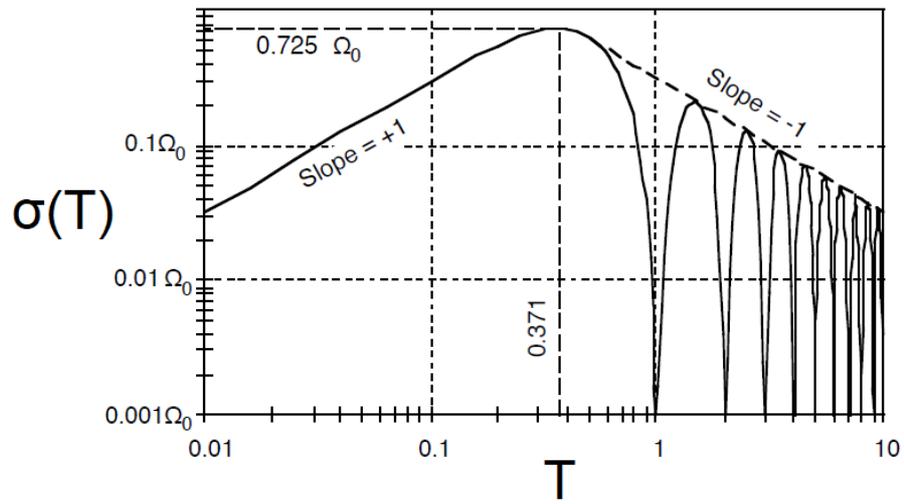


Figure 5.9. Sinusoidal Noise on the ADEV plot

Chapter 6. Test Specifications and Test Procedures

In this chapter are explained all the equipment, procedures, requirements and precautions that should be done in order to perform the Allan Variance test.

6.1. Test Design

The accuracy of the AVAR depends on the number of clusters studied. More clusters lead to better accuracy, but from a finite set of data only a finite set of clusters can be formed. The accuracy, expressed in percentage of the AVAR is defined by

$$\text{Error} = \frac{1}{2\sqrt{\frac{N}{n}} - 1} \cdot 100 \quad (6.1)$$

N being the length of the data set and n the points in each cluster. Using this equation, a test can be designed to study a noise of a certain characteristic time with a given accuracy. This is not the objective of this study, but this equation can be used to know what the error will be for different correlation times.

Performing the Allan Variance requires the sensor to be isolated from movement and vibrations. To ensure that this is the case, the test will be performed during a weekend, when there is less activity at the university. If the test is started on a Saturday morning and stopped the next Monday morning, the test will have an approximate duration of 50 hours. Assuming that the correlation times of the errors being studied are below 2 hours, the maximum error on the Allan Variance will be of 10%. For lower correlation times, the error will be lower, but that must be computed when the correlation error is found.

6.2. Test Set-up

All the required components and test facilities that were required for the realization of the Allan Variance are here detailed:

- EVAL-ADXL355Z
- Female/Female jumper wire
- Raspberry Pi 3B
- Raspberry Pi 3B Power Supply
- Electric extension cord
- Optical table
- Ground Vibration Isolation system
- Air Vibration Isolation system
- Laptop

The EVAL-ADXL355Z is the evaluation board of sensor to be studied. The Female/Female jumper wire is used to connect the accelerometer to the Raspberry Pi 3B, which is used as the power source and controller for the sensor. The Raspberry Pi 3B is configured to emit a Wi-Fi signal, in this way, the Raspberry can be connected to wirelessly without the need of an external Wi-Fi provider. The Raspberry power supply and the electric extension cord are used to power the Raspberry board. The optical table, ground vibration isolation system and air vibration isolation system are used to isolate the sensor from undesired accelerations. The laptop is used to control the Raspberry.

6.3. Test Organization and Schedule

Only one person is required to perform the Allan Variance with the previously explained Set-up. The tasks to be carried out are setting the sensor vibration isolation systems and connect it to the electricity and start and stop data acquisition. In order to ensure that there are minimum disturbances during the experiment, step AVAR – 14, shown in section 6.4. should happen on Friday late afternoon or Saturday morning. In this way, the data will be gathered during the weekend, when there is less activity on the surroundings of the laboratory. This laboratory should also hold a constant temperature.

6.4. Step-by-Step Procedure

All the steps necessary to perform the Allan Variance with the set up explained in section 6.2. are presented in Table 6.1.

Table 6.1. Step-by-Step procedure of Allan Variance experiment

Step ID	Description
AVAR – 1	Set up ground vibration isolation system on top of the optical table
AVAR – 2	Connect ADXL355 to Raspberry Pi 3B according to Annex 3.
AVAR – 3	Place ADXL355 and Raspberry in the ground vibration isolation system
AVAR – 4	Set air vibration isolation system side panels surrounding the ground vibration isolation system
AVAR – 5	Pass microUSB connector of the Raspberry power supply through the top panel of the air vibration isolation system
AVAR – 6	Connect Raspberry power supply to Raspberry
AVAR – 7	Close the air vibration isolation system putting the top panel over the side panels. At the same time, ensure that the Raspberry power supply cable inside the air vibration isolation system is just long to reach the Raspberry board without adding any force
AVAR – 8	Staple the Raspberry power supply cable to the top panel
AVAR – 9	Place electric extension cord on the optical table, next to the air vibration isolation system
AVAR – 10	Connect electric extension cord to the electricity
AVAR – 11	Connect Raspberry power supply to electric extension cord
AVAR – 12	Wait 24 hours for the ground vibration isolation system to reach equilibrium. The set up must left undisturbed, from this step until the end of the experiment, the optical table and all the element on top of it should be left untouched
AVAR – 13	Connect to Raspberry using the emitted Wi-Fi signal with SSID RPI-TFG
AVAR – 14	Start data acquisition
AVAR – 15	Immediately leave the room, being careful not to create vibrations in the process such as closing the door abruptly
AVAR – 16	Wait 50 hours
AVAR – 17	Come back to the room and connect to Raspberry in the same manner as step AVAR – 13
AVAR – 18	Stop data acquisition
AVAR – 19	Download txt file with the data gathered
AVAR – 20	Analyse data

Chapter 7. Results

This section is devoted the analysis of the results obtained performing the test described in Chapter 6.

The data obtained in the experiment is the acceleration output for the three axes and the temperature output. The acceleration output is in g but it is but it is converted into m/s^2 for the analysis ($1\text{g} = 9.81 \text{ m/s}^2$). The temperature output is in $^{\circ}\text{C}$.

On ideal conditions, the sensor Z-axis would be perfectly aligned with Earth's gravity and the readings would be 0 m/s^2 for the X-axis and Y-axis and 9.81 m/s^2 for the Z-axis. This is not the case as there are biases for each axis and in this test there are no methods to determine the alignment of the sensor with Earth's gravity. This is not a problem because the constant accelerations do not affect the resulting AVAR.

The first step is to verify that the data acquired is valid and that the noise the noise of the sensor matches the expected one.

7.1. Data Validation

In order to ensure that the acceleration data gathered during the Allan Variance is not affected by external accelerations, the ground signal of CGAR and BAIN seismographs is checked. Unfortunately, the signal from VILA seismographs was not available for the time of the experiment. Still, with these two seismographs should be enough to know if there has been a seismological event affecting both seismographs. If that is the case, both signals should peak at similar times and the amplitude should be greater than the maximums found in section 2.1.3.

Fortunately, this was not the case. The Allan Variance was conducted during the weekend of 14th and 15th of July, 2018 and the time signals found for both seismographs follow the same patterns explained in section 2.1.3. The PGA for these period of time is actually lower than the PGA of the previously examined dates at $5.62 \cdot 10^{-3} \text{ m/s}^2$.

The output of the sensor can also be affected by changes in the temperature. The temperature affects the linearity of the sensor sensitivity and the value of the value of the bias. The reading of the temperature of the sensor is shown in Figure 7.1.

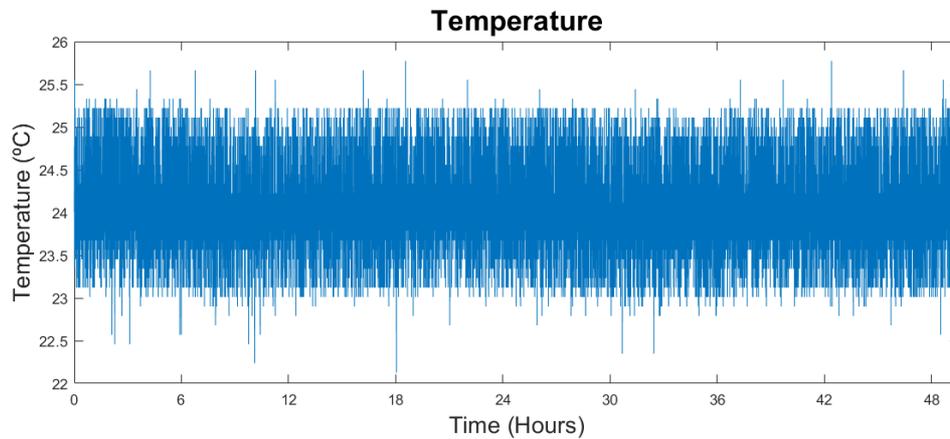


Figure 7.1. Evolution of temperature through time

Although the temperature signal is very noisy, it can be seen that the temperature remains constant throughout the experiment, at around 24 °C. The laboratory in which the experiment was conducted is situated underground, so it is not prone to sudden changes in temperature due to the weather or the differences in temperature between night and day. The accelerations measured are therefore not affected by changes in temperature.

7.2. Analysis of the noise figure

In this section, the analysis of the output of the accelerometer on the time domain is conducted.

The noise on the output of the sensor should match the datasheet specifications. The noise values calculated in section 1.3.1. as well as other information regarding the bias of the sensor are shown in Table 7.1.

Table 7.1. Noise and bias specifications of ADXL355

Specification	Value (m/s ²)
RMS noise	$3.48 \cdot 10^{-4}$
Peak-to-Peak noise	$2.63 \cdot 10^{-3}$
Bias	$\pm 2.45 \cdot 10^{-1}$
Bias Repeatability	X,Y: $\pm 3.45 \cdot 10^{-2}$ Z: $\pm 8.83 \cdot 10^{-2}$

7.2.1. X-axis

The graphic of the time signal for the X-axis is shown in Figure 7.2.

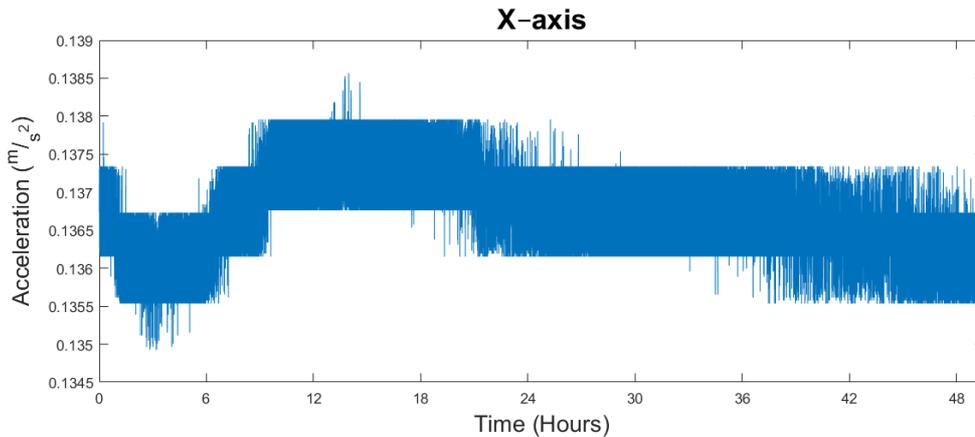


Figure 7.2. Time signal of the X-axis

There are two odd characteristics in the graphic of the X-axis. First, the output is not steady, it changes with time. Valuing the signal at the “centre” of the noise, the signal first decreases, then increase and then it decreases again. The periods of these changes is very large, from 3 to 36 hours, which indicates that this behaviour is not induced by external accelerations, as there are no external forces that can cause this type of accelerations of such a long period. Secondly, the noise of the signal is “bounded” between a bottom and ceiling.

The first of these patterns seems to be caused by the bias repeatability of the sensor, specified at ± 3.5 mg or $\pm 3.45 \cdot 10^{-2}$ m/s². In the changes observed in the output, the bias of the sensor is oscillating $1.5 \cdot 10^{-3}$ m/s² peak-to-peak and therefore it is within the limits of the datasheet.

The second behaviour might be due to the effect of the axis not being stimulated at all. The RMS of the signal, when computed in periods where the signal is stable, is $3.441 \cdot 10^{-4}$ m/s², which coincides with the expected RMS noise and the peak-to-peak noise is $1.383 \cdot 10^{-3}$ m/s², lower than anticipated. This indicates that the noise of the sensor, although with an unusual behaviour, is within the limits of the specifications provided by the manufacturer.

Then, it can be concluded that there are no external noises introduced in the sensor, signifying that the vibrations isolation systems worked properly and that the readings are valid.

Computing the RMS noise of the X-axis signal for the totality of the measurements is $5.154 \cdot 10^{-4}$ m/s² and the peak-to-peak noise is $3.640 \cdot 10^{-3}$ m/s². The mean bias of the sensor is 0.137 m/s², lower than limit of ± 0.245 m/s² of the datasheet.

7.2.2. Y-axis

The graphic of the time signal for the Y-axis is shown in Figure 7.3.

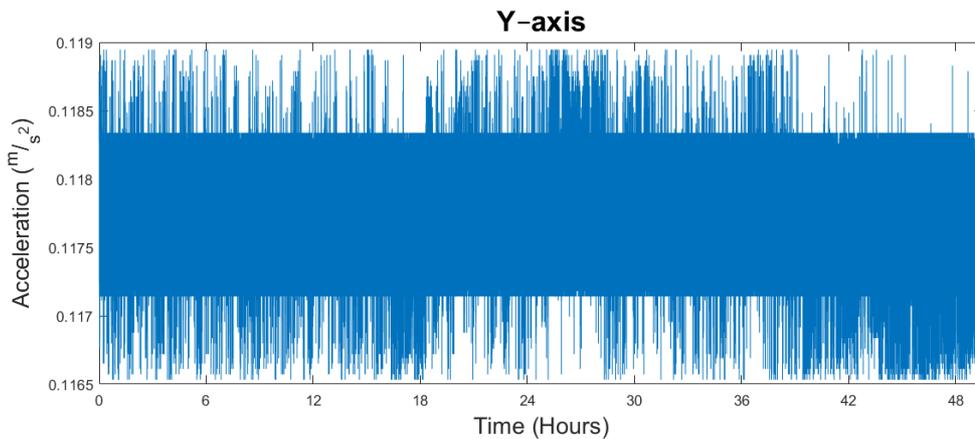


Figure 7.3. Time signal of the Y-axis

The output of the Y-axis does not follow the same trends as the X-axis. The signal, while noisy, does not change with time, further indicating that this effect is not caused by external forces.

The noise of the signal still shares some attributes with the X-axis. The signal is still bounded between a ceiling and bottom. Although, these limits are not so strict as in the X-axis.

The values of noise coincide with those expected. The RMS noise is $3.651 \cdot 10^{-4}$ m/s² and the peak-to-peak noise is $2.413 \cdot 10^{-3}$ m/s². The differences of these values with those calculated in section 1.3.1., according to the datasheet specifications [1], are below 7 % for both numbers.

The bias of the sensor is 0.1177 m/s², under the limit of ± 0.245 m/s² of the datasheet.

7.2.3. Z-axis

The graphic of the time signal for the Z-axis is shown in Figure 7.4.

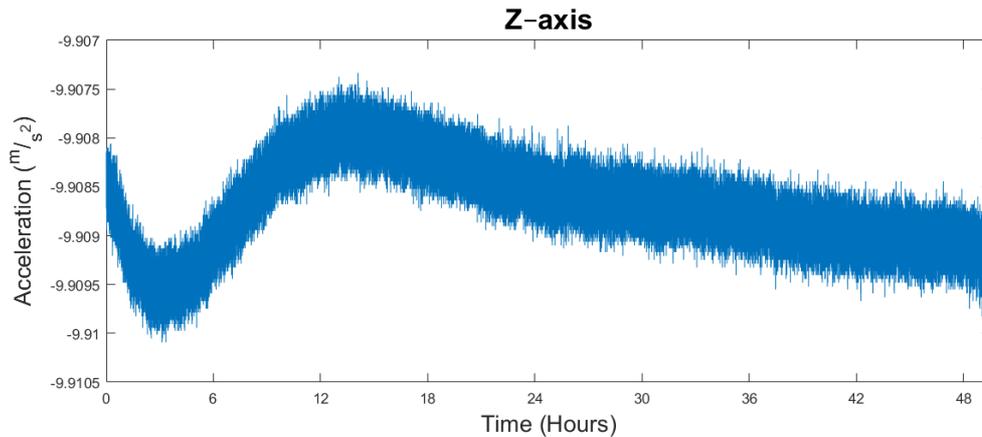


Figure 7.4. Time signal of the Z-axis

The output of the Z-axis shares the same changes of the signal over time with the X-axis. The Z-axis also changes with time and follows the same pattern as the X-axis output, the changes on the bias are the same. Ranging from -9.908 m/s^2 to -9.9095 m/s^2 , the bias oscillates $1.8 \cdot 10^{-3} \text{ m/s}^2$ peak-to-peak, very similar to X-axis at $1.5 \cdot 10^{-3} \text{ m/s}^2$. The periods are also the same, drifting from 3 hours to 36 hours.

This behaviour is attributed to the same cause as the X-axis, the bias repeatability. The bias repeatability is specified at $\pm 9 \text{ mg}$ or $\pm 8.83 \cdot 10^{-2} \text{ m/s}^2$ for Z-axis. Therefore, these oscillations in the bias are within the limits of the datasheet.

The noise of the signal performs better than expected. That is when not taking into account the effects of the bias repeatability. The RMS noise in that case is $1.495 \cdot 10^{-4} \text{ m/s}^2$, less than half of what was expected and the peak-to-peak noise is $1.190 \cdot 10^{-3} \text{ m/s}^2$, around half of what was expected.

The RMS noise on the total Z-axis signal is $4.402 \cdot 10^{-4} \text{ m/s}^2$ and the peak-to-peak noise is $2.757 \cdot 10^{-3} \text{ m/s}^2$.

The bias of the signal cannot be properly analysed from Figure 7.4. because the signal is being stimulated by gravity.

7.3. Allan Deviation analysis

The analysis of the Allan Deviation is presented in this section. For each axis, the noise terms are determined.

Table 7.2. contains the specifications of the datasheet regarding the noise terms [1]. These terms are found on three graphs of the datasheet. The information of the quantization noise, the RRW, RR or sinusoidal noise is provided in the datasheet.

Table 7.2. Noise terms of ADXL355 accelerometer according to the datasheet

Noise terms	X-axis	Y-axis	Z-axis
VRW (m/s/\sqrt{s})	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$
BI (m/s²)	$2 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	$4 \cdot 10^{-5}$

7.3.1. X-axis

The Allan Deviation log-log plot of the X-axis is shown in Figure 7.5., as well as the slopes at which noise errors are found.

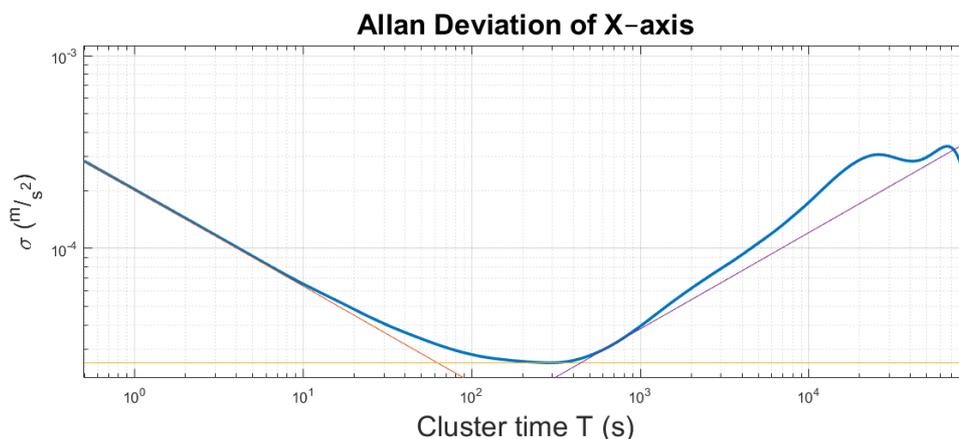


Figure 7.5. Allan Deviation for the X-axis (in blue), Velocity Random Walk slope (in orange), Bias Instability slope (in yellow) and Rate Random Walk slope (in purple)

The ADEV plot obtained follows the typical patterns of the ADEV for inertial sensors.

For the X-axis, only the terms of the VRW, BI and RRW are found. The slope of the ADEV plot starts very close to $-1/2$, and it increases going forward. Consequently, no quantization noise is found, which is found at slope -1 . The ADXL355's analog-to-digital converter has 20bit precision, which is very high. For this reason, it was expected that the quantization noise would be very small. And it is so small in fact, that none is detected in the experiment. Rate Ramp noise is

also not detected, being that the slope never reaches +1. Also, no sinusoidal noise is appreciated.

The VRW is detected at $2.009 \cdot 10^{-4} \text{ m/s}/\sqrt{\text{s}}$ by reading the value of the $-1/2$ slope (in orange in Figure 7.5.) at cluster time $T = 1 \text{ s}$. The error in this measure depends on the number of clusters of the signal and is defined by Eq.(6.1). The VRW is detected at cluster time $T = 0.5 \text{ s}$ and the resulting error is 0.09 %. The value of the VRW found is 2 times greater than the value provided by the datasheet.

The BI found is $3.830 \cdot 10^{-5} \text{ m/s}^2$ by reading the value of the ADEV at slope 0 (in yellow in Figure 7.5.), found on the first local minimum of the graph and dividing by 0.664, as explained in section 5.4.1. The error of the BI can be determined in the same way as the VRW. The BI is found at cluster time $T = 291 \text{ s}$ and the error is 2.05%. The value found on the experiment is 1.92 times greater than the value provided in the datasheet. This noise term is ten times smaller than the VRW and for this reason creates a much smaller impact on the accelerometer output.

The RRW is detected at $2.094 \cdot 10^{-6} \text{ m/s}^2/\sqrt{\text{s}}$ by reading the value of slope $+1/2$ (in purple in Figure 7.5.) at cluster time $T = 3 \text{ s}$. The error of this measure is calculated using the same method as in the VRW and BI. The error is 3.15%. The value of this noise term cannot be compared to an expected value because it is not provided by the datasheet. Still, this noise is one order of magnitude smaller than the BI and two orders of magnitude smaller than the VRW and ergo is of much smaller importance. It does not create a relevant error compared with the other two noise terms found.

7.3.2. Y-axis

The Allan Deviation log-log plot of the Y-axis is shown in Figure 7.6., as well as the slopes at which noise errors are found.

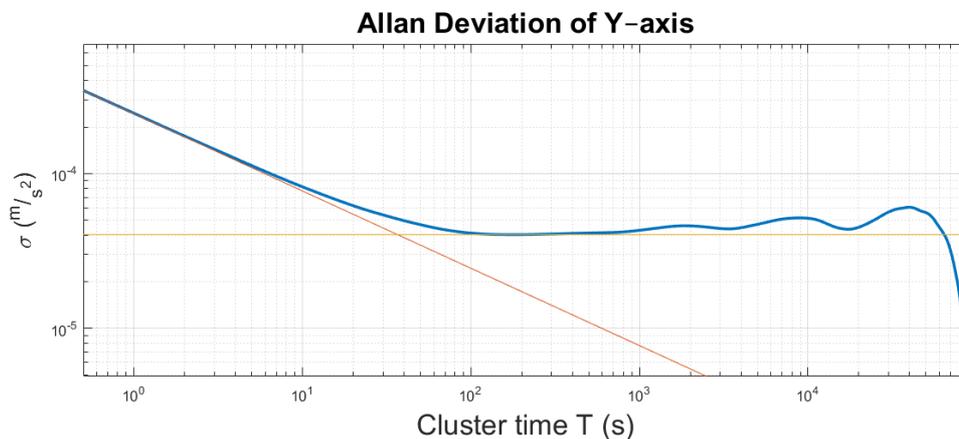


Figure 7.6. Allan Deviation for the Y-axis (in blue), Velocity Random Walk slope (in orange) and Bias Instability slope (in yellow)

The noise terms found on the Y-axis are the VRW and the BI. Like on the X-axis, the slope of the ADEV plot never is -1 , $+1/2$ or $+1$ and hence, no quantification noise, RRW or RR are found. Also no sinusoidal noise is found. Although from cluster time $T = 1000$ s the ADEV plot develops an oscillating behaviour it should not be confused with sinusoidal noise. The sinusoidal noise increases and then decreases rapidly, as seen in Figure 5.9., and in this case it seems to increase slowly and not decrease. Therefore, these oscillations cannot be attributed to sinusoidal noise.

The VRW is obtained in the same manner as the X-axis and it is $2.439 \cdot 10^{-4}$ m/s/ \sqrt{s} . It is found at cluster time $T = 0.5$ s, which constitutes an error of the 0.09% according to Eq.(6.1). The VRW found in the experiment is 2.44 times greater than expected according to the datasheet.

The BI is measured at $6.071 \cdot 10^{-5}$ m/s² making it 3.03 times greater than the value of the datasheet. The BI is found at cluster time $T = 169$ s and the corresponding error in the measure is 1.56%. The BI is 4 times smaller than VRW and hence it contributes less to the error of the signal. Still, this contribution is more important in the Y-axis than in the X-axis.

7.3.3. Z-axis

The Allan Deviation log-log plot of the Z-axis is shown in Figure 7.7., as well as the slopes at which noise errors are found. The shape of this ADEV plot is very typical for accelerometers or other inertial sensors.

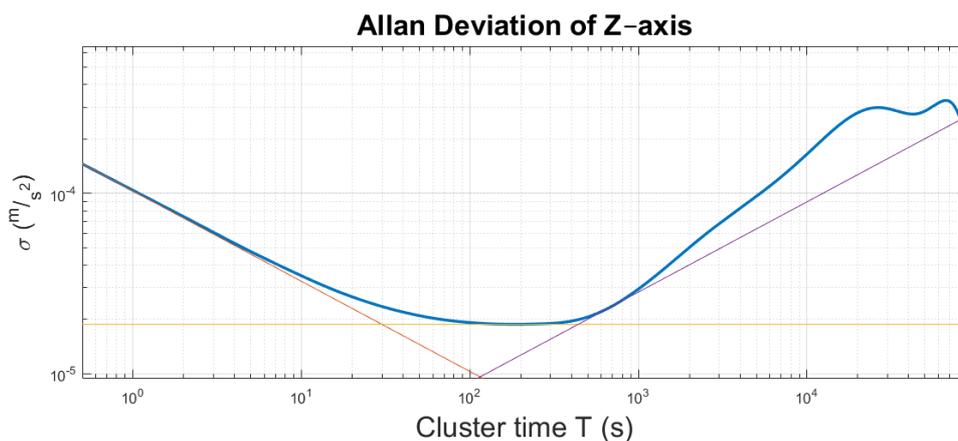


Figure 7.7. Allan Deviation for the Z-axis (in blue), Velocity Random Walk slope (in orange), Bias Instability slope (in yellow) and Rate Random Walk slope (in purple)

In the ADEV of the Z-axis the VRW, BI and RRW are found. The slope of the ADEV plot starts at $-1/2$ and increases up to cluster time $2 \cdot 10^4$ s, where the signal starts to oscillate. Like in the case of the Y-axis, this pattern is not to be

interpreted as sinusoidal noise. The slope never passes through -1 and never reaches $+1$, therefore quantification noise and RR noise are not found.

The accelerometer is equipped with a 20bit ADC, which provides great resolution. For this reason, the quantification is so small that it is not detected. Also, the sensor seems to not be affected by a RR noise at all.

The VRW is found to be $1.028 \cdot 10^{-4} \text{ m/s}/\sqrt{\text{s}}$, 20 % smaller than the value of the datasheet. It is detected at cluster time $T = 0.5 \text{ s}$ with an error of 0.09 %.

The BI is $2.830 \cdot 10^{-5} \text{ m/s}^2$, almost 1.5 times smaller than the values of the datasheet. It is detected at cluster time $T = 191 \text{ s}$ making the error in the measurement 1.66 %.

The RRW, found at cluster $T = 685 \text{ s}$, is $1.555 \cdot 10^{-6} \text{ m/s}^2/\sqrt{\text{s}}$ and has an error of 3.15 %. The measurement cannot be compared with an expected value because no values of the RRW appear on the data sheet. The value of the RRW is the smallest of all the noise terms found and hence contributes much less to the total noise of the signal.

7.3.4. Review of the results

Here is presented a summary of the results of the experiment.

In Table 7.3 the noise terms for each axis are found. The errors in this measures ranges from 0.09 % to 3.15%. They are very precise and because these errors are related to the duration of the experiment it can be considered that the duration was appropriate for measuring the noises found.

Table 7.3. Compilation of noise terms obtained by the experiment

Noise terms	X-axis	Y-axis	Z-axis
Quantization noise (m/s^2)	–	–	–
VRW ($\text{m/s}/\sqrt{\text{s}}$)	$2.009 \cdot 10^{-4}$	$2.439 \cdot 10^{-4}$	$1.028 \cdot 10^{-4}$
BI (m/s^2)	$3.830 \cdot 10^{-5}$	$6.071 \cdot 10^{-5}$	$2.830 \cdot 10^{-5}$
RRW ($\text{m/s}^2/\sqrt{\text{s}}$)	$2.094 \cdot 10^{-6}$	–	$1.555 \cdot 10^{-6}$
RR ($\text{m/s}^2/\text{s}$)	–	–	–

The results of can be considered that meet the specifications of the datasheet as far as the VRW and BI. X-axis and Y-axis both approximately double the value of the datasheet, but on the datasheet they are obtained visually from another ADEV plot so there is quite a lot of error in the measure. Also, the values are very similar between both axes which was expected because the specifications are the all the same for X-axis and Y-axis whereas some differ for Z-axis. This

indicates that the X-axis and Y-axis are coupled together in a MEMS accelerometer of sensitive to planar acceleration and Z-axis is independent.

In contrast with the rest of axes, the Z-axis outperforms the values of the datasheet, although not by a great measure.

Chapter 8. Conclusions

The aim of this thesis has been characterizing and quantifying the noise properties of an ultralow noise MEMS accelerometer. An accelerometer that, if suitable, would go aboard a femtosat as part of the science payload in a space mission. The accelerometer studied was chosen according to the requirements of the mission and the scientific experiment.

The study of the different types of noises affecting the output of the accelerometer will allow to determine if the sensor is actually appropriate for the mission and in the case that this is the sensor used for the mission, these noise characteristics would be used to calibrate the sensor and create a more precise and reliable output, therefore improving the scientific experiment.

In order to be able to study the noise characteristics of the accelerometer, the sensor has been isolated from external factors that could alter the output of the accelerometer. An extensive study of the possible sources of noise has been done and the necessary isolation has been devised by comparing the magnitude of these sources of external noise with the amplitude of the noise of the sensor.

To achieve this isolation a system to passively attenuate vibrations has been designed and assembled.

The method for studying the noise characteristics of the sensor has been the Allan Variance. This method has been studied and so have the types of noises affecting accelerometers. The necessary test to perform the Allan Variance has been designed and implemented.

The results obtained of the Allan Variance are considered satisfactory. The values noise characteristics of the signal indicate that the isolation system has worked properly and that the reading of the sensor correspond with its natural noise.

It has been found that the sensor is not affected by quantization noise, rate ramp or sinusoidal noise and for the Y-axis of the sensor, also not the rate random walk. The velocity random walk and bias instability have been determined for all three axis of the accelerometer and the rate random walk has been determined for X-axis and Z-axis. The values found conform with the specification of the datasheet.

The objective of the thesis has then been accomplished. Although it is still too early in the mission stages to determine if the sensor is the most appropriate for the task, it is a valid candidate. Further research of other sensors and advancements on the mission stages must be done to select the definitive sensor. Still, the procedures and structures designed, built and implemented could be used to study other accelerometers.

Bibliography

- [1] Analog Devices, "ADXL354/ADXL355 Low Noise, Low Drift, Low Power, 3-Axis MEMS Accelerometers Data Sheet (Rev. A)" (2018)
- [2] Analog Devices, "EVAL-ADXL354/EVAL-ADXL355 User Guide UG-1030 (Rev. A)" (2018)
- [3] McCarthy, M., "Peak-to-Peak Resolution Versus Effective Resolution AN-615", Analog Devices, Application Note (2003)
- [4] Mityska, M., "ReadMSEEDFast(filename)", Matlab program (2017) (URL: <https://es.mathworks.com/matlabcentral/fileexchange/46532-readmseefast-filename>)
- [5] Bormann, P., Wielandt, E., "Seismic Signals and Noise", Chapter 4 in *New Manual of Sismological Observatory Practice 2*, Bormann, 1-62, Potsdam: Deutsches GeoForschungsZentrum GFZ (2013)
- [6] Rivin, E.I., *Passive Vibration Isolation*, American Society of Mechanical Engineers PRESS, New York, NY (2003)
- [7] Zuo, L., "Element and System Design for Active and Passive Vibration Isolation", Ph.D. Dissertation, Massachusetts Institute of Technology (2005)
- [8] Giaime, J., Saha, P., Shoemaker, D., Sievers, L., "A passive vibration isolation stack for LIGO: Design, modelling, and testing", *Review of Scientific Instruments*, 67 (1), 208-214 (1996)
- [9] Eshkabilov, S., "Modeling and simulation of non-linear and hysteresis behaviour of magneto-rheological dampers in the example of quarter-car model", *International Journal of Theoretical and Applied Mathematics*, 2 (2), 170-189 (2016)
- [10] Pavlou, E.A., "Dynamic Analysis of Systems with Hysteretic Damping", M. Sc. Thesis, RICE University (1999)
- [11] Pellicer, M., Solà-Morales, J., "Analysis of a viscoelastic spring-mass model", *Journal of Mathematical Analysis and Applications*, 294 (2), 687-698 (2004)
- [12] Beards, C. F., "The vibrations of systems having one degree of freedom", Chapter 2 in *Engineering Vibration Analysis with Application to Control Systems*, 10-87, Edward Arnold, London, UK (1995)
- [13] Frankovich, D., "The Basics of Vibration Isolation Using Elastomeric Materials", Aearo Technologies LLC, Technical Report, (Visited 2018)
- [14] Liu, C. Xu, D., Ji, J. "Theoretical design and experimental verification of a tuneable floating vibration isolation system", *Journal of Sound and Vibration*, 331 (21), 4691-4703 (2012)
- [15] Liu, C., Xu, D., Zhou, J. X., Bishop, S., "On theoretical and experimental study of a two-degree-of-freedom anti-resonance floating vibration isolation system", *Journal of Vibration and Control*, 21 (10), 1886-1901 (2015)
- [16] Vandiver, J. K., Gossard, D., "2.003SC, Engineering Dynamics", Massachusetts Institute of Technology: MIT OpenCourseWare (2011)
- [17] Allan, D., "Statistics of atomic frequency standards", *Proceedings of IEEE*, 54 (2), 221-230 (1966)
- [18] "IEEE Standard Specification Format Guide and Test Procedure for Linear, Single-Axis, Non - Gyroscopic Accelerometers", IEEE Std. 1293-1998 (R2008) (2008)

- [19] "IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros", IEEE Std. 952-1997 (R2003) (2003)
- [20] ESA Requirements and Standards Division, "Space engineering. Gyro terminology and performance specification", ECSS-E-ST-60-21C (2017)
- [21] Stockwell, W., "Bias Stability Measurement: Allan Variance", Crossbow Technology, Inc., Technical Report, (Visited 2018)
- [22] Stockwell, W., "Angle Random Walk", Crossbow Technology, Inc., Technical Report, Visited 2018
- [23] El-Sheimy, N., Hou, H., Niu, X., "Analysis and Modeling of Inertial Sensors Using Allan Variance", *IEEE Transactions on Instrumentation and Measurement*, 57 (1), 140-149 (2008)

List of Figures

Figure 1.1. Graphic of flat SND analogous to SND on ADXL355	5
Figure 2.1. Map of seismographs (in red) and university (in green) locations and coordinates	11
Figure 2.2. Seismic time signal for the HHZ axis of CGAR seismograph during the 1 st of February 2018	13
Figure 2.3. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 1 st of February 2018	14
Figure 2.4. Seismic time signal for the HHN axis of VILA seismograph during the 15 th of February 2018	15
Figure 2.5. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 15 th of February 2018.....	16
Figure 2.6. Seismic time signal for the HHE axis of BAIN seismograph during the 13 th of March 2018	17
Figure 2.7. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 13 th of March 2018	18
Figure 3.1. Floating vibration isolation system designed by Liu, C. viewed from the front and the top	22
Figure 3.2. Ground Vibration Isolation System	23
Figure 3.3. Attenuation of Ground Vibration Isolation System. The attenuation of the system is in blue, the minimum necessary attenuation is in black and the desired attenuation is in grey.....	28
Figure 3.4. Floating vibration isolation system.....	29
Figure 4.1. Design of the air isolation system. Units are in centimetres.	31
Figure 4.2. Air vibration isolation system assembly. Attaching side panels to top panel.	32
Figure 4.3. Illustration of air vibration isolation system top panel with the viscoelastic padding.	33
Figure 5.1. Overlapping clusters of Allan Variance.....	36
Figure 5.2. Non-overlapping clusters of Allan Variance.....	37
Figure 5.3. Typical ADEV plot for inertial sensors	38
Figure 5.4. Bias instability on the ADEV plot	39
Figure 5.5. Velocity Random Walk on the ADEV plot.....	40
Figure 5.6. Rate Random Walk on the ADEV plot.....	40
Figure 5.7. Rate Ramp on the ADEV plot.....	41
Figure 5.8. Quantization Noise on the ADEV plot.....	41
Figure 5.9. Sinusoidal Noise on the ADEV plot	42
Figure 7.1. Evolution of temperature through time	47
Figure 7.2. Time signal of the X-axis	48
Figure 7.3. Time signal of the Y-axis	49
Figure 7.4. Time signal of the Z-axis	50
Figure 7.5. Allan Deviation for the X-axis (in blue), Velocity Random Walk slope (in orange), Bias Instability slope (in yellow) and Rate Random Walk slope (in purple)	51
Figure 7.6. Allan Deviation for the Y-axis (in blue), Velocity Random Walk slope (in orange) and Bias Instability slope (in yellow).....	52
Figure 7.7. Allan Deviation for the Z-axis (in blue), Velocity Random Walk slope (in orange), Bias Instability slope (in yellow) and Rate Random Walk slope (in purple)	53

Figure A2.1. Seismic time signal for the HHE axis of CGAR seismograph during the 1 st of February 2018	90
Figure A2.2. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 1 st of February 2018	90
Figure A2.3. Seismic time signal for the HHN axis of CGAR seismograph during the 1 st of February 2018	91
Figure A2.4. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 1 st of February 2018	91
Figure A2.5. Seismic time signal for the HHZ axis of CGAR seismograph during the 1 st of February 2018	91
Figure A2.6. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 1 st of February 2018	92
Figure A2.7. Seismic time signal for the HHE axis of CGAR seismograph during the 11 th of February 2018	92
Figure A2.8. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 11 th of February 2018	92
Figure A2.9. Seismic time signal for the HHN axis of CGAR seismograph during the 11 th of February 2018	93
Figure A2.10. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 11 th of February 2018	93
Figure A2.11. Seismic time signal for the HHZ axis of CGAR seismograph during the 11 th of February 2018	93
Figure A2.12. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 11 th of February 2018	94
Figure A2.13. Seismic time signal for the HHE axis of CGAR seismograph during the 21 st of February 2018	94
Figure A2.14. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 21 st of February 2018	94
Figure A2.15. Seismic time signal for the HHN axis of CGAR seismograph during the 21 st of February 2018	95
Figure A2.16. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 21 st of February 2018	95
Figure A2.17. Seismic time signal for the HHZ axis of CGAR seismograph during the 21 st of February 2018	95
Figure A2.18. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 21 st of February 2018	96
Figure A2.19. Seismic time signal for the HHE axis of CGAR seismograph during the 1 st of March 2018	96
Figure A2.20. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 1 st of March 2018	96
Figure A2.21. Seismic time signal for the HHN axis of CGAR seismograph during the 1 st of March 2018	97
Figure A2.22. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 1 st of March 2018	97
Figure A2.23. Seismic time signal for the HHZ axis of CGAR seismograph during the 1 st of March 2018	97
Figure A2.24. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 1 st of March 2018	98
Figure A2.25. Seismic time signal for the HHE axis of CGAR seismograph during the 11 th of March 2018	98

Figure A2.26. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 11 th of March 2018.....	98
Figure A2.27. Seismic time signal for the HHN axis of CGAR seismograph during the 11 th of March 2018	99
Figure A2.28. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 11 th of March 2018.....	99
Figure A2.29. Seismic time signal for the HHZ axis of CGAR seismograph during the 11 th of March 2018	99
Figure A2.30. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 11 th of March 2018	100
Figure A2.31. Seismic time signal for the HHE axis of VILA seismograph during the 1 st of February 2018	100
Figure A2.32. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 1 st of February 2018	100
Figure A2.33. Seismic time signal for the HHN axis of VILA seismograph during the 1 st of February 2018	101
Figure A2.34. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 1 st of February 2018	101
Figure A2.35. Seismic time signal for the HHZ axis of VILA seismograph during the 1 st of February 2018	101
Figure A2.36. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 1 st of February 2018	102
Figure A2.37. Seismic time signal for the HHE axis of VILA seismograph during the 11 th of February 2018	102
Figure A2.38. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 11 th of February 2018	102
Figure A2.39. Seismic time signal for the HHN axis of VILA seismograph during the 11 th of February 2018	103
Figure A2.40. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 11 th of February 2018.....	103
Figure A2.41. Seismic time signal for the HHZ axis of VILA seismograph during the 11 th of February 2018	103
Figure A2.42. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 11 th of February 2018	104
Figure A2.43. Seismic time signal for the HHE axis of VILA seismograph during the 15 th of February 2018	104
Figure A2.44. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 15 th of February 2018	104
Figure A2.45. Seismic time signal for the HHN axis of VILA seismograph during the 15 th of February 2018	105
Figure A2.46. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 15 th of February 2018.....	105
Figure A2.47. Seismic time signal for the HHZ axis of VILA seismograph during the 15 th of February 2018	105
Figure A2.48. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 15 th of February 2018	106
Figure A2.49. Seismic time signal for the HHE axis of VILA seismograph during the 6 th of March 2018	106
Figure A2.50. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 6 th of March 2018.....	106

Figure A2.51. Seismic time signal for the HHN axis of VILA seismograph during the 6 th of March 2018	107
Figure A2.52. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 6 th of March 2018	107
Figure A2.53. Seismic time signal for the HHZ axis of VILA seismograph during the 6 th of March 2018	107
Figure A2.54. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 6 th of March 2018.....	108
Figure A2.55. Seismic time signal for the HHE axis of VILA seismograph during the 11 th of March 2018	108
Figure A2.56. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 11 th of March 2018.....	108
Figure A2.57. Seismic time signal for the HHN axis of VILA seismograph during the 11 th of March 2018	109
Figure A2.58. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 11 th of March 2018	109
Figure A2.59. Seismic time signal for the HHZ axis of VILA seismograph during the 11 th of March 2018	109
Figure A2.60. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 11 th of March 2018.....	110
Figure A2.61. Seismic time signal for the HHE axis of BAIN seismograph during the 1 st of February 2018	110
Figure A2.62. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 1 st of February 2018.....	110
Figure A2.63. Seismic time signal for the HHN axis of BAIN seismograph during the 1 st of February 2018	111
Figure A2.64. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 1 st of February 2018	111
Figure A2.65. Seismic time signal for the HHZ axis of BAIN seismograph during the 1 st of February 2018	111
Figure A2.66. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 1 st of February 2018.....	112
Figure A2.67. Seismic time signal for the HHE axis of BAIN seismograph during the 11 th of February 2018	112
Figure A2.68. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 11 th of February 2018.....	112
Figure A2.69. Seismic time signal for the HHN axis of BAIN seismograph during the 11 th of February 2018	113
Figure A2.70. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 11 th of February 2018	113
Figure A2.71. Seismic time signal for the HHZ axis of BAIN seismograph during the 11 th of February 2018	113
Figure A2.72. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 11 th of February 2018.....	114
Figure A2.73. Seismic time signal for the HHE axis of BAIN seismograph during the 21 st of February 2018	114
Figure A2.74. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 21 st of February 2018.....	114
Figure A2.75. Seismic time signal for the HHN axis of BAIN seismograph during the 21 st of February 2018	115

Figure A2.76. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 21 st of February 2018.....	115
Figure A2.77. Seismic time signal for the HHZ axis of BAIN seismograph during the 21 st of February 2018	115
Figure A2.78. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 21 st of February 2018.....	116
Figure A2.79. Seismic time signal for the HHE axis of BAIN seismograph during the 1 st of March 2018.....	116
Figure A2.80. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 1 st of March 2018	116
Figure A2.81. Seismic time signal for the HHN axis of BAIN seismograph during the 1 st of March 2018.....	117
Figure A2.82. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 1 st of March 2018.....	117
Figure A2.83. Seismic time signal for the HHZ axis of BAIN seismograph during the 1 st of March 2018.....	117
Figure A2.84. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 1 st of March 2018	118
Figure A2.85. Seismic time signal for the HHE axis of BAIN seismograph during the 13 th of March 2018	118
Figure A2.86. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 13 th of March 2018.....	118
Figure A2.87. Seismic time signal for the HHN axis of BAIN seismograph during the 13 th of March 2018	119
Figure A2.88. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 13 th of March 2018.....	119
Figure A2.89. Seismic time signal for the HHZ axis of BAIN seismograph during the 13 th of March 2018	119
Figure A2.90. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 13 th of March 2018	120

List of Tables

Table 1.1. ADXL355 Specifications for ± 2.048 g range, $T_A = 25^\circ\text{C}$ and $V_{\text{SUPPLY}} = 3.3\text{V}$	7
Table 2.1. Peak ground accelerations and frequency of maximum concentration of vibrations at CGAR location	15
Table 2.2. Peak ground accelerations and frequency of maximum concentration of vibrations at VILA location	17
Table 2.3. Peak ground accelerations and frequency of maximum concentration of vibrations at BAIN location	18
Table 3.1. PGAs for each seismograph.....	20
Table 3.2. Design parameters of ground vibration isolation system	26
Table 3.3. Design parameters for floating isolation	26
Table 6.1. Step-by-Step procedure of Allan Variance experiment.....	45
Table 7.1. Noise and bias specifications of ADXL355	47
Table 7.2. Noise terms of ADXL355 accelerometer according to the datasheet	51
Table 7.3. Compilation of noise terms obtained by the experiment	54

Annex 1. Code

In this annex all the code used for conducting this thesis.

1.1. Create Seismic Query

This is the code is used to create URL queries. When the created URL is typed into an internet navigator, the chosen seismic data will be downloaded in miniSEED format. These queries only work for seismographs part of the ICGC network.

Not all the data is available for every seismograph, so for certain dates or seismographs the queries will not respond when used on the navigator.

This code is written in Matlab 2017.

```
%% Create queries for ICGC

% User input:
%   network: Seismic Network. Only use 'CA'.
%   station: Seismograph code. For this thesis: 'CGAR', 'VILA' or
%   'BAIN'.
%   start: Starting date for the seismic signal. Format: YYYY-MM-DD
%   final: Ending date for the seismic signal. Format: YYYY-MM-DD
% Output:
%   query: Resulting query.

network = 'CA'; %Catalan Seismic Network
station = 'BAIN';
start = '2018-02-01';
final = '2018-02-12';
query = strcat('http://ws.icgc.cat/fdsnws/dataselect/1/query?net=', ...
    network, '&sta=', station, '&start=', start, '&end=', final);
fprintf('\n %s \n\n', query);
```

1.2. Analyse Seismic Signals

1.2.1. CGAR and VILA seismographs

This is the code used to analyse the seismic signals from CGAR and VILA seismographs. The data from these seismographs is of velocity, for that reason this data must be derived respect to time to obtain the acceleration, which is what has to be analysed.

The data is divided by a constant value obtained from the frequency response to translate the data into m/s. Then this data is derived against time obtaining the true acceleration. Finally, the FFT of the acceleration signal is calculated.

The PGA and frequency of maximum spectrum density are computed and printed on the command window.

This code is written in Matlab 2017.

```

%% Analyse seismic signals from either CGAR or VILA seismographs

% Input from user:
%   file: name of the .mat file where the information of the signal is
%   stored. This file shall contain the instrumental response
information
%   like the poles, zeros and constants of the transfer function and
the
%   transfer function itself in zpk format (poles_HHE, zeros_HHE,
%   const_HHE, poles_HHN, zeros_HHN, const_HHN, poles_HHZ, zeros_HHZ
and
%   const_HHZ), frequency and period of the output (fs and dt) and the
%   seismic signal in column vectors. (signalHHE, signalHHN,
signalHHZ).

% Output:
%   - Plots of:
%       - Acceleration output of the seismograph in m/s^2, corrected
for
%       the instrumental response.
%       - Normalized Fast Fourier Transform of the signal.
%   - Peak Ground Acceleration
%   - Frequency of maximum spectrum density

%% Load file
clear all
close all
file = 'CGAR20180201';
load(file);

%% Preprocess
% Make the signal vectors length an even number
if mod(length(signalHHE),2) ~= 0
    signalHHE = signalHHE(1:(end-1));
end
if mod(length(signalHHN),2) ~= 0
    signalHHN = signalHHN(1:(end-1));
end
if mod(length(signalHHZ),2) ~= 0
    signalHHZ = signalHHZ(1:(end-1));
end
% Create time vector
timevec_HHE = 0:dt:dt*(length(signalHHE)-1);
timevec_HHN = 0:dt:dt*(length(signalHHN)-1);
timevec_HHZ = 0:dt:dt*(length(signalHHZ)-1);

%% HHE axis
% Translate data into velocity
vel_HHE = signalHHE./gain;
% Compute time derivative
acc_HHE=diff(vel_HHE)./diff(timevec_HHE');
% Compute FFT of the acceleration
n = length(vel_HHE);
N = n;
acc_fft_HHE = fft(acc_HHE,N);

```

```

acc_fft_HHE = acc_fft_HHE(1:end/2);
acc_fft_HHE = flip(acc_fft_HHE);
freq_HHE = fs*(1/N:1/N:1);
freq_HHE = freq_HHE(1:end/2);
% Determine and print PGA and frequency of maximum spectrum density
PGA_HHE = max(abs(acc_HHE));
[max_fres_HHE, ind_HHE] = max(abs(acc_fft_HHE));
fprintf('\n %s \n', file);
fprintf('PGA_HHE: %g \n',PGA_HHE);
fprintf('Freq res HHE: %g \n', freq_HHE(ind_HHE));
% Plot acceleration signal
figure('units','normalized','outerposition',[0 0.3 0.8 0.7]);
timevec_HHE = timevec_HHE/3600;
plot(timevec_HHE(1:(end-1)),acc_HHE,'b');
hold on;
date = strcat('Date',{' '},file(11:12),'-',file(9:10),'-',file(5:8));
name = strcat('Time signal of CGAR-HHE',{' '},date);
% name = strcat('Time signal of VILA-HHE',{' '},date);
title(name{1});
xlabel('Time');
ylabel('Acceleration (m/{s^2})');
xticks([0 6 12 18 24])
xticklabels({'00:00','06:00','12:00','18:00','24:00'})
line([0 timevec_HHE(end)],[0 0],'Color',...
      [0.7 0.7 0.7],'LineWidth',0.1);
xlim([0 24]);
% Plot acceleration FFT
figure('units','normalized','outerposition',[0 0.1 0.8 0.7]);
loglog(freq_HHE,abs(acc_fft_HHE)./max(abs(acc_fft_HHE)),'b');
name = strcat('Normalized Fast Fourier Transform of CGAR-HHE',...
              {' '},date);
% name = strcat('Normalized Fast Fourier Transform of VILA-HHE',...
%               {' '},date);
title(name{1});
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([freq_HHE(1) freq_HHE(end)]);
grid on
% Free memory
clearvars n N vel_HHE acc_HHE acc_fft_HHE freq_HHE
clearvars PGA_HHE timevec_HHE max_fres_HHE ind_HHE

%% HHN axis
% Translate data into velocity
vel_HHN = signalHHN./gain;
% Compute time derivative
acc_HHN=diff(vel_HHN)./diff(timevec_HHN');
% Compute FFT of the acceleration
n = length(vel_HHN);
N = n;
acc_fft_HHN = fft(acc_HHN,N);
acc_fft_HHN = acc_fft_HHN(1:end/2);
acc_fft_HHN = flip(acc_fft_HHN);
freq_HHN = fs*(1/N:1/N:1);
freq_HHN = freq_HHN(1:end/2);
% Determine and print PGA and frequency of maximum spectrum density
PGA_HHN = max(abs(acc_HHN));
[max_freq_HHN, ind_HHN] = max(abs(acc_fft_HHN));
fprintf('PGA_HHN: %g \n',PGA_HHN);
fprintf('Freq res HHN: %g \n', freq_HHN(ind_HHN));
% Plot acceleration signal

```

```

figure('units','normalized','outerposition',[0 0.3 0.8 0.7]);
timevec_HHN = timevec_HHN/3600;
plot(timevec_HHN(1:(end-1)),acc_HHN,'b');
hold on;
date = strcat('Date',{' '},file(11:12),'-',file(9:10),'-',file(5:8));
name = strcat('Time signal of CGAR-HHN',{' '},date);
% name = strcat('Time signal of VILA-HHN',{' '},date);
title(name{1});
xlabel('Time');
ylabel('Acceleration (^m/{s^2})');
xticks([0 6 12 18 24])
xticklabels({'00:00','06:00','12:00','18:00','24:00'})
line([0 timevec_HHN(end)],[0 0],'Color',...
      [0.7 0.7 0.7],'LineWidth',0.1);
xlim([0 24]);
% Plot acceleration FFT
figure('units','normalized','outerposition',[0 0.1 0.8 0.7]);
loglog(freq_HHN,abs(acc_fft_HHN)./max(abs(acc_fft_HHN)),'b');
name = strcat('Normalized Fast Fourier Transform of CGAR-HHN',...
              {' '},date);
% name = strcat('Normalized Fast Fourier Transform of VILA-HHN',...
%              {' '},date);
title(name{1});
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([freq_HHN(1) freq_HHN(end)]);
grid on
% Free memory
clearvars n N vel_HHN acc_HHN acc_fft_HHN freq_HHN
clearvars PGA_HHN timevec_HHN max_fres_HHN ind_HHN

%% HHZ axis
% Translate data into velocity
vel_HHZ = signalHHZ./gain;
% Compute time derivative
acc_HHZ=diff(vel_HHZ)./diff(timevec_HHZ');
% Compute FFT of the acceleration
n = length(vel_HHZ);
N = n;
acc_fft_HHZ = fft(acc_HHZ,N);
acc_fft_HHZ = acc_fft_HHZ(1:end/2);
acc_fft_HHZ = flip(acc_fft_HHZ);
freq_HHZ = fs*(1/N:1/N:1);
freq_HHZ = freq_HHZ(1:end/2);
% Determine and print PGA and frequency of maximum spectrum density
PGA_HHZ = max(abs(acc_HHZ));
[max_fres_HHZ, ind_HHZ] = max(abs(acc_fft_HHZ));
fprintf('PGA_HHN: %g \n',PGA_HHZ);
fprintf('Freq res HHN: %g \n', freq_HHZ(ind_HHZ));
% Plot acceleration signal
figure('units','normalized','outerposition',[0 0.3 0.8 0.7]);
timevec_HHZ = timevec_HHZ/3600;
plot(timevec_HHZ(1:(end-1)),acc_HHZ,'b');
hold on;
date = strcat('Date',{' '},file(11:12),'-',file(9:10),'-',file(5:8));
name = strcat('Time signal of CGAR-HHZ',{' '},date);
% name = strcat('Time signal of VILA-HHZ',{' '},date);
title(name{1});
xlabel('Time');
ylabel('Acceleration (^m/{s^2})');
xticks([0 6 12 18 24])

```

```

xticklabels({'00:00','06:00','12:00','18:00','24:00'})
line([0 timevec_HHZ(end)], [0 0], 'Color', ...
     [0.7 0.7 0.7], 'LineWidth', 0.1);
xlim([0 24]);
% Plot acceleration FFT
figure('units', 'normalized', 'outerposition', [0 0.1 0.8 0.7]);
loglog(freq_HHZ, abs(acc_fft_HHZ) ./ max(abs(acc_fft_HHZ)), 'b');
name = strcat('Normalized Fast Fourier Transform of CGAR-HHZ', ...
             {' '}, date);
% name = strcat('Normalized Fast Fourier Transform of VILA-HHZ', ...
%             {' '}, date);
title(name{1});
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([freq_HHZ(1) freq_HHZ(end)]);
grid on
% Free memory
clearvars n N vel_HHZ acc_HHZ acc_fft_HHZ freq_HHZ
clearvars PGA_HHZ timevec_HHZ max_fres_HHZ ind_HHZ

```

1.2.2. BAIN seismograph

This is the code used to analyse the seismic signals from BAIN seismograph. The output signal of this seismograph is of acceleration, therefore no time derivative is needed in this case.

First the data is translated into the frequency spectrum using the FFT and this FFT is corrected by the frequency response of the sensor. The resulting data is the FFT of the corrected signal. This data is then translated back into the time domain to obtain the acceleration signal. The PGA and frequency of maximum spectrum density are computed and printed on the command window.

This code is written in Matlab 2017.

```

%% Analyse seismic signals from BAIN seismograph

% Input from user:
% file: name of the .mat file where the information of the signal is
% stored. This file shall contain the instrumental response
% information like the poles, zeros and constants of the transfer
% function and the transfer function itself in zpk format
% (poles_HHE, zeros_HHE, const_HHE, poles_HHN, zeros_HHN, const_HHN,
% poles_HHZ, zeros_HHZ and const_HHZ), frequency and period of the
% output (fs and dt) and the seismic signal in column vectors
% (signalHHE, signalHHN, signalHHZ).

% Output:
% - Plots of:
%   - Acceleration output of the seismograph in m/s^2, corrected
%     for the instrumental response.
%   - Normalized Fast Fourier Transform of the signal.
% - Peak Ground Acceleration
% - Frequency of maximum spectrum density

%% Load file
clear all
close all

```

```

file = 'BAIN20180313';
load(file);

%% Preprocess
% Create time vector
timevec_HHE = 0:dt:dt*(length(signalHHE)-1);
timevec_HHN = 0:dt:dt*(length(signalHHN)-1);
timevec_HHZ = 0:dt:dt*(length(signalHHZ)-1);
% Create frequency response system
sys_HHE = zpk(zeros_HHE, poles_HHE, const_HHE);
sys_HHN = zpk(zeros_HHN, poles_HHN, const_HHN);
sys_HHZ = zpk(zeros_HHZ, poles_HHZ, const_HHZ);
% Eliminate offsets
signalHHE = signalHHE - mean(signalHHE);
signalHHN = signalHHN - mean(signalHHN);
signalHHZ = signalHHZ - mean(signalHHZ);

%% HHE axis
% Compute FFT of original data set
n = length(timevec_HHE);
N = n;
fft_HHE = fft(signalHHE, N);
xfft_HHE = fs*(0:1/N:(1-1/N));
fft_HHE = fft_HHE(2001:end/2);
fft_HHE = flip(fft_HHE);
xfft_HHE = xfft_HHE(1:(end/2-2000));
% Correct for the frequency response
[ gain_HHE, phase_HHE, wout_HHE ] = bode(sys_HHE, xfft_HHE*2*pi);
gain_HHE = gain_HHE(:);
fout_HHE = wout_HHE/(2*pi);
fft_HHE_filt = fft_HHE./gain_HHE;
% Translate back into the time domain
signal_HHE_filt = real(ifft(fft_HHE_filt));
signal_HHE_filt = flip(signal_HHE_filt);
fn = fs/2;
dtn = 1/fn;
timevec_HHE_filt = (0:dtn:dt*(length(timevec_HHE)-1))/3600; % horas;
% Determine and print PGA and frequency of maximum spectrum density
PGA_HHE = max(abs(signal_HHE_filt));
[ max_fres_HHE, ind_HHE ] = max(abs(fft_HHE_filt));
fprintf('%s \n', file);
fprintf('PGA_HHE: %g \n', PGA_HHE);
fprintf('Freq res HHE: %g \n', fout_HHE(ind_HHE));
% Plot acceleration signal
figure('units','normalized','outerposition',[0 0.3 0.8 0.7]);
plot(timevec_HHE_filt(1:end-2000), signal_HHE_filt, 'b');
hold on;
date = strcat('Date',{' '},file(11:12),'-',file(9:10),'-',file(5:8));
name = strcat('Time signal of BAIN-HHE',{' '},date);
title(name{1});
xlabel('Time');
ylabel('Acceleration (^m/_{s^2})');
xticks([0 6 12 18 24]);
xticklabels({'00:00','06:00','12:00','18:00','24:00'})
line([0 timevec_HHE_filt(end)], [0 0], 'Color', ...
      [0.7 0.7 0.7], 'LineWidth', 0.1);
xlim([0 timevec_HHE_filt(end)]);
% Plot acceleration FFT
figure('units','normalized','outerposition',[0 0.1 0.8 0.7]);
loglog(fout_HHE(1:end), abs(fft_HHE_filt(1:end)) ...
       ./max(abs(fft_HHE_filt(1:end))), 'b');

```

```

name = strcat('Normalized Fast Fourier Transform of BAIN-HHE',...
    {' '},date);
title(name{1});
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([fout_HHE(1) fout_HHE(end)]);
hold on
% Free memory
clearvars n N fft_HHE xfft_HHE fft_HHE xfft_HHE gain_HHE ...
    wout_HHE sys_HHE xfft_HHE gain_HHE fout_HHE fft_HHE_filt ...
    signal_HHE_filt fn dtn timevec_HHE_filt

%% HHN axis
% Compute FFT of original data set
n = length(timevec_HHN);
N = n;
fft_HHN = fft(signalHHN, N);
xffft_HHN = fs*(1/N:1/N:1);
fft_HHN = fft_HHN(2001:end/2);
fft_HHN = flip(fft_HHN);
xffft_HHN = xffft_HHN(1:end/2-2000);
% Correct for the frequency response
[ gain_HHN, phase_HHN, wout_HHN ] = bode(sys_HHN, xffft_HHN*2*pi);
gain_HHN = gain_HHN(:);
fout_HHN = wout_HHN/(2*pi);
fft_HHN_filt = fft_HHN./gain_HHN;
% Translate back into the time domain
signal_HHN_filt = real(ifft(fft_HHN_filt));
signal_HHN_filt = flip(signal_HHN_filt);
fn = fs/2;
dtn = 1/fn;
timevec_HHN_filt = (0:dtn:dt*(length(timevec_HHN)-1))/3600; % horas
% Determine and print PGA and frequency of maximum spectrum density
PGA_HHN = max(abs(signal_HHN_filt));
[ max_fres_HHN, ind_HHN ] = max(abs(fft_HHN_filt));
fprintf('PGA_HHN: %g \n', PGA_HHN);
fprintf('Freq res HHN: %g \n', fout_HHN(ind_HHN));
% Plot acceleration signal
figure('units','normalized','outerposition',[0 0.3 0.8 0.7]);
plot(timevec_HHN_filt(1:end-2000),signal_HHN_filt,'b');
hold on
date = strcat('Date',{' '},file(11:12),'-',file(9:10),'-',file(5:8));
name = strcat('Time signal of BAIN-HHN',{' '},date);
title(name{1});
xlabel('Time');
ylabel('Acceleration (^m/_{s^2})');
xticks([0 6 12 18 24])
xticklabels({'00:00','06:00','12:00','18:00','24:00'})
line([0 timevec_HHN_filt(end)],[0 0],'Color',...
    [0.7 0.7 0.7],'LineWidth',0.1);
xlim([0 timevec_HHN_filt(end)]);
% Plot acceleration FFT
figure('units','normalized','outerposition',[0 0.3 0.8 0.7]);
loglog(fout_HHN(1:end),abs(fft_HHN_filt(1:end))...
    ./max(abs(fft_HHN_filt(1:end))), 'b');
hold on
date = strcat('Date',{' '},file(11:12),'-',file(9:10),'-',file(5:8));
name = strcat('Normalized Fast Fourier Transform of BAIN-HHN',...
    {' '},date);
title(name{1});
xlabel('Frequency (Hz)');

```

```

ylabel('Amplitude');
xlim([fout_HHN(1) fout_HHN(end)]);
% Free memory
clearvars n N fft_HHN xfft_HHN fft_HHN xfft_HHN gain_HHN ...
        wout_HHN sys_HHN xfft_HHN gain_HHN fout_HHN fft_HHN_filt ...
        signal_HHN_filt fn dtn timevec_HHN_filt

%% HHZ
% Compute FFT of original data set
n = length(timevec_HHZ);
N = n;
fft_HHZ = fft(signalHHZ, N);
xffft_HHZ = fs*(1/N:1/N:1);
fft_HHZ = fft_HHZ(2001:end/2);
fft_HHZ = flip(fft_HHZ);
xffft_HHZ = xffft_HHZ(1:end/2-2000);
% Correct for the frequency response
[gain_HHZ,phase_HHZ,wout_HHZ] = bode(sys_HHZ,xffft_HHZ*2*pi);
gain_HHZ = gain_HHZ(:);
fout_HHZ = wout_HHZ/(2*pi);
fft_HHZ_filt = fft_HHZ./gain_HHZ;
% Translate back into the time domain
signal_HHZ_filt = real(ifft(fft_HHZ_filt));
signal_HHZ_filt = flip(signal_HHZ_filt);
fn = fs/2;
dtn = 1/fn;
timevec_HHZ_filt = (0:dtn:dt*(length(timevec_HHZ)-1))/3600; % horas
% Determine and print PGA and frequency of maximum spectrum density
PGA_HHZ = max(abs(signal_HHZ_filt));
[max_fres_HHZ, ind_HHZ] = max(abs(fft_HHZ_filt));
fprintf('PGA_HHZ: %g \n',PGA_HHZ);
fprintf('Freq res HHZ: %g \n\n', fout_HHZ(ind_HHZ));
% Plot acceleration signal
figure('units','normalized','outerposition',[0 0.3 0.8 0.7]);
plot(timevec_HHZ_filt(1:end-2000),signal_HHZ_filt,'b');
hold on
name = strcat('Time signal of BAIN-HHZ',{' '},date);
title(name{1});
xlabel('Time');
ylabel('Acceleration (^m/{s^2})');
xticks([0 6 12 18 24])
xticklabels({'00:00','06:00','12:00','18:00','24:00'})
line([0 timevec_HHZ_filt(end)],[0 0],'Color',...
      [0.7 0.7 0.7],'LineWidth',0.1);
xlim([0 timevec_HHZ_filt(end)]);
% Plot acceleration FFT
figure('units','normalized','outerposition',[0 0.3 0.8 0.7]);
loglog(fout_HHZ(1:end),abs(fft_HHZ_filt(1:end))...
       ./max(abs(fft_HHZ_filt(1:end))), 'b');
name = strcat('Normalized Fast Fourier Transform of BAIN-HHZ',...
             {' '},date);
title(name{1});
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([fout_HHZ(1) fout_HHZ(end)]);
hold on
% Free memory
clearvars n N fft_HHZ xfft_HHZ fft_HHZ xfft_HHZ gain_HHZ ...
        wout_HHZ sys_HHZ xfft_HHZ gain_HHZ fout_HHZ fft_HHZ_filt ...
        signal_HHZ_filt fn dtn timevec_HHZ_filt

```

1.3. Analyse ground vibration isolation

This is the code used to analyse the performance of the ground vibration isolation system. The code is composed of a main program, which contains the variables of the system and calculates the attenuation through different frequencies and the function with the differential equations.

This code is written in Matlab 2017.

1.3.1. Main program

```
clear all
close all

% Gravity acceleration
g = 9.81; % m/s^2

% Spring stack isolation constants
m1 = 2.78 + 2.79; % kg
m2 = 2.79 + 2.76; % kg
Mcontainer = 2.2175; % kg
Mwater = 3.196; % kg
m3 = Mcontainer + Mwater; % kg
k1 = 4*(2480); % N/m
k2 = 4*(1759); % N/m
k3 = 4*(1191); % N/m
loss1 = 0.1; % addimensional
loss2 = 0.1; % addimensional
loss3 = 0.1; % addimensional

% Floating isolation constants
Mmetacrilato = 0.942; % kg
Macquisition = 0.181; % kg
Mf = Mmetacrilato + Macquisition; % kg
wi = 0.05; % m
H0 = 0.03; % m
h0 = 0.04; % m
L = 0.13; % m
W = 0.13; % m
rho = 1000; % kg/m^3
k4 = 4*193; % N/m
loss4 = 0.1; % addimensional
Madded = (rho/2) * (W*L^3/(6*h0) + (W*H0*L^2/wi)); % kg
Ke = k4 + rho * g * W * L * (1 + L/(2*wi)); % N/m
Cd = 100; % kg/s

% Mass and Spring constants matrices
M = diag([m1 m2 m3 (Mf+Madded)]);
K = [k1+k2, -k2, 0, 0;
     -k2, k2+k3, -k3, 0;
     0, -k3, k3+Ke, -Ke;
     0, 0, -Ke, Ke];

% Natural frequencies
E = inv(M)*K;
[V,D] = eig(E);
w_n = sqrt(D); % rad/s
```

```

f_n = w_n./(2*pi); % Hz

% Damping constantns of the springs
c1 = k1*loss1/(2*pi*f_n(1,1)); % kg/s
c2 = k2*loss2/(2*pi*f_n(2,2)); % kg/s
c3 = k3*loss3/(2*pi*f_n(3,3)); % kg/s
c4 = k4*loss4/(2*pi*f_n(4,4)); % kg/s

% Values for solving differential equations
x0 = [0, 0, 0, 0, 0, 0, 0, 0]; % Initial conditions
dt = 0.001; % s
ft = 1/dt; % Hz
endtime = 100; % s
tspan = 0:dt:endtime; % time vector

% Ground excitation
A = 1e-1; % m/s^2
fase = 0; % rad

% Preallocate vectors
f_e = zeros(100,1);
Attenuation = zeros(100,1);

for n = 1:5000

    f_e(n) = 0.05*n;
    w_e = 2*pi*f_e(n);

    y = zeros(length(tspan),1);
    ydot = zeros(length(tspan),1);
    ydotdot = zeros(length(tspan),1);
    for i = 1:length(tspan)
        ydotdot(i) = sum(A.*sin(w_e.*tspan(i) - fase));
        ydot(i) = sum(-(A./w_e).*cos(w_e.*tspan(i) - fase));
        y(i) = sum(-(A./w_e.^2).*sin(w_e.*tspan(i) - fase));
    end

    % Solve differential equations using a
    % 4th order Runge Kutta solver
    [ts, xs] = ode45(@(t,x) FourDOF(t,x, m1,m2,m3,Mf,Madded, ...
        k1,k2,k3,Ke, c1,c2,c3,c4,Cd, ...
        A,w_e,fase), tspan, x0);

    % Steady state is considered at the las 10% of the signal
    steady = fix(90/100*length(tspan));
    % Join position data
    xy_result = [xs(steady:end,1), xs(steady:end,3), ...
        xs(steady:end,5), xs(steady:end,7), y(steady:end)];

    % Compute velocity of all variables
    xydot = diff(xy_result)./diff(tspan(steady:end)');
    % Compute acceleration of all variables
    acc = diff(xydot)./diff(tspan(steady:(end-1))');

    % Compute attenuation
    Attenuation(n) = max(abs(acc(:,5)))/max(abs(acc(:,4)));

end

% Process Attenuation
Attenuation = 1./Attenuation;

```

```

Attenuation = Attenuation./Attenuation(1);

% Plot Attenuation
figure();
loglog(f_e,Attenuation);
hold on
title('Attenuation of Ground Vibration Isolation
System','FontSize',20);
xlabel('Frequency (Hz)','FontSize',16);
ylabel('Attenuation','FontSize',16);
xlim([f_e(1) f_e(end)]);
loglog([f_e(1) f_e(end)], [1/1000 1/1000],'Color',[0.7 0.7 0.7]);
loglog([f_e(1) f_e(end)], [1/100 1/100], 'k');

```

1.3.2. Differential equations

```

function dxdt = FourDOF(t,x, m1,m2,m3,Mf,Madded, ...
                        k1,k2,k3,Ke,c1,c2,c3,c4,Cd,A,w,fase)

%
%
% Differential equations for the ground vibration isolation system
%
% Inputs: All doubles
% - t: time variable. This variable is used by Matlab,
%       the user shall not provide it
% - x: state variable. This variable is used by Matlab,
%       the user shall not provide it
% - m1: 1st mass
% - m2: 2nd mass
% - m3: 3rd mass (mass of the floating container and the water)
% - Mf: mass of the floating structure and data acquisition
%       system (ADXL355, Raspberry Pi 3B, cables, structure, etc).
% - Madded: virtual mass created by the displacement of water
% - k1: spring stiffness constant of the 1st spring
% - k2: spring stiffness constant of the 2nd spring
% - k3: spring stiffness constant of the 3rd spring
% - Ke: spring stiffness constant of the 4th spring plus the "spring
%       constant" created by the floating structure being
%       submerged in water
% - c1: sprind damping constant of the 1st spring
% - c2: sprind damping constant of the 2nd spring
% - c3: sprind damping constant of the 3rd spring
% - c4: sprind damping constant of the 4th spring
% - Cd: Damping coefficient floating structure with water
% - A: Ground acceleration amplitude
% - w: Ground acceleration frequency
% - fase: Ground acceleration fase
%
% Output:
% - dxdt: Column vector with the values of the derivatives of x
%         [x1dot; x1dotdot; x2dot; x1dotdot; x2dot;
%          x1dotdot; x3dot; x3dotdot; x4dot; x4dotdot]
%
% x = [x1; x1dot; x2; x2dot; x3; x3dot; x4; x4dot];

% Ground excitation
y = -(A/w^2)*sin(w*t - fase);
ydot = -(A/w)*cos(w*t - fase);

```

```

% Differential equations
dxdt = zeros(8,1);

dxdt(1) = x(2);
dxdt(2) = (1/m1)*(- (c1+c2)*x(2) + c2*x(4) - (k1+k2)*x(1) ...
           + k2*x(3) + k1*y + c1*ydot);
dxdt(3) = x(4);
dxdt(4) = (1/m2)*(c2*x(2) - (c2+c3)*x(4) + c3*x(6) + k2*x(1) ...
           - (k2+k3)*x(3) + k3*x(5));
dxdt(5) = x(6);
dxdt(6) = (1/m3)*(c3*x(4) - (c3 + c4 + Cd)*x(6) + (c4 + Cd)*x(8) ...
           + k3*x(3) - (k3+Ke)*x(5) + Ke*x(7));
dxdt(7) = x(8);
dxdt(8) = (1/(Mf + Madded))*((c4+Cd)*x(6) - (c4 + Cd)*x(8) ...
           + Ke*x(5) - Ke*x(7));

end

```

1.4. ADXL355 Sensor

In this annex it is presented the code used to operate and obtain data or accelerometer ADXL355 using a Raspberry Pi 3B.

This code is written in Python 3.

1.4.1. ADXL355 Class

```

# Import necessary libraries
import spidev
# SPI configuration
SPI_MAX_CLOCK_HZ = 10000000
SPI_MIN_CLOCK_HZ = 100000
SPI_MODE = 0b00 # CPOL = 0, CPHA = 0
SPI_BUS = 0
SPI_DEVICE = 0
# ADXL355 addresses
DEVID_AD = 0x00
DEVID_MST = 0x01
PARTID = 0x02
REVID = 0x03
STATUS = 0x04
FIFO_ENTRIES = 0x05
TEMP2 = 0x06
TEMP1 = 0x07
XDATA3 = 0x08
XDATA2 = 0x09
XDATA1 = 0x0A
YDATA3 = 0x0B
YDATA2 = 0x0C
YDATA1 = 0x0D
ZDATA3 = 0x0E
ZDATA2 = 0x0F
ZDATA1 = 0x10
FIFO_DATA = 0x11
OFFSET_X_H = 0x1E
OFFSET_X_L = 0x1F

```

```

OFFSET_Y_H = 0x20
OFFSET_Y_L = 0x21
OFFSET_Z_H = 0x22
OFFSET_Z_L = 0x23
ACT_EN = 0x24
ACT_THRESH_H = 0x25
ACT_THRESH_L = 0x26
ACT_COUNT = 0x27
FILTER = 0x28
FIFO_SAMPLES = 0x29
INT_MAP = 0x2A
SYNC = 0x2B
RANGE = 0x2C
POWER_CTL = 0x2D
SELF_TEST = 0x2E
RESET = 0x2F
# Data Range
RANGE_2G = 0x01
RANGE_4G = 0x02
RANGE_8g = 0x03
# Values
WRITE_BIT = 0x00
READ_BIT = 0x01
DUMMY_BYTE = 0xAA
MEASURE_MODE = 0x06

class ADXL355:
    """
    Class to interact with ADXL355 device
    Allows user to read, write and obtain data
    from the accelerometer
    """

    def __init__(self, measure_range=RANGE_2G):
        """Initializes sensor and SPI bus.
        Args:
            measure_range (int): measure range selected
        Returns:
            None
        """
        # Initialize SPI
        self.spi = spidev.SpiDev()
        self.spi.open(SPI_BUS, SPI_DEVICE)
        self.spi.max_speed_hz = SPI_MAX_CLOCK_HZ
        self.spi.mode = SPI_MODE
        # Initialize sensor
        self._set_measure_range(measure_range)
        self._enable_measure_mode()

    def write_data(self, address, value):
        """Writes data on ADXL355 device address.
        Args:
            address (int): Address to write in ADXL355.
            value (int): Value to write in address.
        Returns:
            None
        """
        device_address = address << 1 | WRITE_BIT
        self.spi.xfer2([device_address, value])

    def read_data(self, address):

```

```

    """Reads data from ADXL355 device.
    Args:
        address (int): Address to read from ADXL355.
    Returns:
        int: Value in specified address in accelerometer
    """
    device_address = address << 1 | READ_BIT
    return self.spi.xfer2([device_address, DUMMY_BYTE])[1]

def _set_measure_range(self, measure_range):
    """Sets measure range on ADXL355 device.
    Args:
        measure_range (int): Measure range to set in ADXL355.
    Returns:
        None
    """
    # Write data
    self.write_data(RANGE, measure_range)

def get_measure_range(self):
    """Gets measure range
    Returns:
        measure_range (int): Measure range
    """
    # Read data
    raw_data = self.read_data(RANGE)
    # Split data
    measure_range = (raw_data - ((raw_data >> 2) << 2))
    # Return values
    return measure_range

def _enable_measure_mode(self):
    """
    Enables measure mode on ADXL355 device.
    Returns:
        None
    """
    # Write data
    self.write_data(POWER_CTL, MEASURE_MODE)

def set_measure_mode(self, drdy_off, temp_off, standby):
    """Sets measure mode
    Returns:
        None
    """
    # Read register before modifying it
    data = self.read_data(POWER_CTL)
    # Preserve reserved data, discard the rest
    data = ((data >> 3) << 3)
    # Add measure mode
    data = data + (drdy_off << 2) + (temp_off << 1) + standby
    # Write data
    self.write_data(POWER_CTL, data)

def get_axes(self):
    """
    Gets the current data from the axes.
    Returns:
        dict: Current value for x, y and z axis
    """
    # Reading data

```

```

x_data = [self.read_data(XDATA1), self.read_data(XDATA2), \
          self.read_data(XDATA3)]
y_data = [self.read_data(YDATA1), self.read_data(YDATA2), \
          self.read_data(YDATA3)]
z_data = [self.read_data(ZDATA1), self.read_data(ZDATA2), \
          self.read_data(ZDATA3)]
# Join data
x_data = (x_data[0] >> 4) + (x_data[1] << 4) \
        + (x_data[2] << 12)
y_data = (y_data[0] >> 4) + (y_data[1] << 4) \
        + (y_data[2] << 12)
z_data = (z_data[0] >> 4) + (z_data[1] << 4) \
        + (z_data[2] << 12)
# Apply two complement
x_data = twos_comp(x_data, 20)
y_data = twos_comp(y_data, 20)
z_data = twos_comp(z_data, 20)
# Return values
return [x_data, y_data, z_data]

def get_temperature(self):
    """Get current temperature.
    Returns:
        (int): Current temperature in 12bit
    """
    # Reading data
    temp_data = [self.read_data(TEMP1), self.read_data(TEMP2)]
    # Join data
    temp_data = (temp_data[0]) + ((temp_data[1] \
        - ((temp_data[1] >> 4) << 4)) << 8)
    # Return values
    return temp_data

def get_axes_and_temp(self):
    """Get current accelerations and temperature
    Returns:
        (dict): x, y, z, temperature
    """
    # Reading data
    x_data = [self.read_data(XDATA1), self.read_data(XDATA2), \
              self.read_data(XDATA3)]
    y_data = [self.read_data(YDATA1), self.read_data(YDATA2), \
              self.read_data(YDATA3)]
    z_data = [self.read_data(ZDATA1), self.read_data(ZDATA2), \
              self.read_data(ZDATA3)]
    temp_data = [self.read_data(TEMP1), self.read_data(TEMP2)]
    # Join data
    x_data = (x_data[0] >> 4) + (x_data[1] << 4) \
            + (x_data[2] << 12)
    y_data = (y_data[0] >> 4) + (y_data[1] << 4) \
            + (y_data[2] << 12)
    z_data = (z_data[0] >> 4) + (z_data[1] << 4) \
            + (z_data[2] << 12)
    temp_data = ((temp_data[1] - ((temp_data[1] >> 4) << 4)) \
        << 8) + (temp_data[0] << 0)
    # Apply two complement
    if (x_data & (1 << 19)) != 0:
        x_data = x_data - (1 << 20)
    if (y_data & (1 << 19)) != 0:
        y_data = y_data - (1 << 20)
    if (z_data & (1 << 19)) != 0:

```

```

        z_data = z_data - (1 << 20)
    # Return values
    return [-x_data, -y_data, -z_data, temp_data]

def set_ODR_and_filter(self, odr_lpf, hpf_filter):
    """Set Low Pass Filter, Output Data Rate (ODR) and
        High Pass Filter (HPF)
    Returns:
        None
    """
    # Read register before modifying
    data = self.read_data(FILTER)
    # Preserve reserved data
    data = ((data >> 7) << 7)
    # Join data
    data = data + (hpf_filter << 4) + odr_lpf
    # Write data
    self.write_data(FILTER, data)

def get_ODR_and_filter(self):
    """Gets Low Pass Filter, Output Data Rate (ODR) and
        High Pass Filter (HPF)
    Returns:
        (int): odr_lpf (ODR and LPF bits),
            hpf_filter (HPF bits)
    """
    # Read data
    raw_data = self.read_data(FILTER)
    odr_lpf = raw_data - ((raw_data >> 4) << 4)
    hpf_filter = (raw_data >> 4)
    hpf_filter = hpf_filter - ((hpf_filter >> 3) << 3)
    # Return values
    return [odr_lpf, hpf_filter]

def set_sync(self, ext_clk, ext_sync):
    """Sets type of synchronization
    Returns:
        None
    """
    # Read data on register before modifying
    data = self.read_data(SYNC)
    # Preserve reserved bits
    data = ((data >> 3) << 3)
    # Join data, not modifying reserved data
    data = (data) + (ext_clk << 2) + ext_sync
    # Write data
    self.write_data(SYNC, data)

def get_sync(self):
    """Gets type of synchronization
    Returns:
        (int): synchronization type
    """
    # Read data
    raw_data = self.read_data(SYNC)
    # Split bits
    ext_sync = raw_data - ((raw_data >> 2) << 2)
    ext_clk = (raw_data >> 2)
    ext_clk = ext_clk - ((ext_clk >> 1) << 1)
    # Return values
    return [ext_clk, ext_sync]

```

```

def get_status(self):
    """Get status
    Returns:
        (int):
    """
    # Read data
    raw_data = self.read_data(STATUS)
    # Split bits
    data_rdy = raw_data - ((raw_data >> 1) << 1)
    raw_data = (raw_data >> 1)
    fifo_full = raw_data - ((raw_data >> 1) << 1)
    raw_data = (raw_data >> 1)
    fifo_ovr = raw_data - ((raw_data >> 1) << 1)
    raw_data = (raw_data >> 1)
    activity = raw_data - ((raw_data >> 1) << 1)
    raw_data = (raw_data >> 1)
    nvm_busy = raw_data - ((raw_data >> 1) << 1)
    # Return values
    return [data_rdy, fifo_full, fifo_ovr, activity, \
            nvm_busy]

def get_offsets(self):
    """Get X, Y, Z offsets
    Return:
        ():
    """
    # Read data
    x_data = [self.read_data(OFFSET_X_H), \
              self.read_data(OFFSET_X_L)]
    y_data = [self.read_data(OFFSET_Y_H), \
              self.read_data(OFFSET_Y_L)]
    z_data = [self.read_data(OFFSET_Z_H), \
              self.read_data(OFFSET_Z_L)]
    # Join data
    x_data = x_data[1] + (x_data[0] << 8)
    y_data = y_data[1] + (y_data[0] << 8)
    z_data = z_data[1] + (z_data[0] << 8)
    # The significance of OFFSET[15:0] matches
    # the significance of DATA[19:4]
    x_data = (x_data << 4)
    y_data = (y_data << 4)
    z_data = (z_data << 4)
    # Apply two's complement
    x_data = twos_comp(x_data, 20)
    y_data = twos_comp(y_data, 20)
    z_data = twos_comp(z_data, 20)
    # Return values
    return [x_data, y_data, z_data]

def reset_settings(self):
    """Reset all settings
    Returns:
        None
    """
    # Reset command
    self.write_data(RESET, 0x52)

```

1.4.2. Data acquisition

Here is the code used for data acquisition. First the main program used is show. This program starts the sensor and saves the output data onto a text file. Then a file containing the various defined functions used is presented.

1.4.2.1. *MainProgram.py*

```
import time
import datetime
import sys
from defined_functions import *
from adxl355 import ADXL355

# Open file
filename = '{0:%Y%m%d-%H.%M.txt}'.format(datetime.datetime.now())
output_file = open(filename, 'w')
print(filename)

# Constants
temp_scale = -9.05 # LSB/°C
temp_bias = 1852 # LSB

# Start sensor
sensor = ADXL355()
time.sleep(0.2)

# Reset all settings
sensor.reset_settings()
time.sleep(0.2)

# Set measure mode
sensor.set_measure_mode(drdy_off = 1, temp_off = 0, standby = 0)

# Set settings
f_s = 3 # [Hz]
dt = 1/f_s # [s]

range_g = 0x01 # 2g range
sensor.set_measure_range(range_g) #Set range
odr_lpf = 0xA # ODR: 3.906Hz, LPF: 0.977Hz
#odr_lpf = 0x04 # ODR: 500Hz, LPF: 125Hz
hpf = 0x00 # No high pass filter
sensor.set_ODR_and_filter(odr_lpf, hpf)
time.sleep(0.2)

# Display settings
range_g = sensor.get_measure_range()
range_g = bits2range(range_g)
scale = range_g[1]
odr_filter = sensor.get_ODR_and_filter()
odr_lpf_hpf = bits2odr_filter(odr_filter) # Interpret bits
offsets = sensor.get_offsets()
sync_bits = sensor.get_sync()
syncs = bits2sync(sync_bits) # Interpret bits
status = sensor.get_status()
print('Measure range: ', range_g[0], file=output_file)
print('Output Data Rate: ', odr_lpf_hpf[0], file=output_file)
print('Low pass filter: ', odr_lpf_hpf[1], file=output_file)
```

```

print('High pass filter: ', odr_lpf_hpf[2], file=output_file)
print('Data on twos_comp method', file=output_file)
print('Data sampled at: ', f_s, file=output_file)
print('\n', file=output_file)
# Display settings on terminal
print('Measure range: ', range_g[0])
print('Output Data Rate: ', odr_lpf_hpf[0])
print('Low pass filter: ', odr_lpf_hpf[1])
print('High pass filter: ', odr_lpf_hpf[2])
print('Data on twos_comp method')
print('Data sampled at: ', f_s)
print('\n')
print('Sensor running...')
print('\n')
time.sleep(0.01)

n = 1
t = 0.0
# Infinite loop
while (1):
    startTime = time.time()
    axes_temp = sensor.get_axes_and_temp()
    accel = bits2accel(axes_temp[0:3], [scale, scale, scale], offsets)
    temp = bits2temp(axes_temp[3], temp_scale, temp_bias)
    print('ID: %d t: %f x: %f y: %f z: %f Temp: %f' % (n, t, \
        accel[0], accel[1], accel[2], temp), file=output_file)
    n = n + 1
    time.sleep(dt)
    t = t + time.time() - startTime

```

1.4.2.2. *defined_functions.py*

```

"""
    Functions for use in the main program for ADXL355 Sensor
"""

def bits2range(bits):
    """Interprets the bits to know the measure range and scale
    Return:
        (str): measure range
        (float): scale
    """
    # Interpret range
    if bits == 0x01:
        meas_range = '2g'
        scale = 256000 #LSB/g
    elif bits == 0x02:
        meas_range = '4g'
        scale = 128000 #LSB/g
    else:
        meas_range = '8g'
        scale = 64000 #LSB/g
    # Return value
    return [meas_range, scale]

def bits2odr_filter(bits):
    """Interprets the bits to know the correspondig ODR,
    LPF and HPF
    Return:

```

```

        (str): ODR, LPF, HPF
    """
    # Interpret ODR and LPF
    if bits[0] == 0x00:
        odr = '4000 Hz'
        lpf = '1000 Hz'
    elif bits[0] == 0x01:
        odr = '2000 Hz'
        lpf = '500 Hz'
    elif bits[0] == 0x02:
        odr = '1000 Hz'
        lpf = '250 Hz'
    elif bits[0] == 0x03:
        odr = '500 Hz'
        lpf = '125 Hz'
    elif bits[0] == 0x04:
        odr = '250 Hz'
        lpf = '62.5 Hz'
    elif bits[0] == 0x05:
        odr = '125 Hz'
        lpf = '31.25 Hz'
    elif bits[0] == 0x06:
        odr = '62.5 Hz'
        lpf = '15.625 Hz'
    elif bits[0] == 0x07:
        odr = '31.25 Hz'
        lpf = '7.813 Hz'
    elif bits[0] == 0x08:
        odr = '15.625 Hz'
        lpf = '3.906 Hz'
    elif bits[0] == 0x09:
        odr = '7.813 Hz'
        lpf = '1.953 Hz'
    else:
        odr = '3.906 Hz'
        lpf = '0.977 Hz'
    # Interpret HPF
    if bits[1] == 0x00:
        hpf = 'No high pass filter'
    elif bits[1] == 0x01:
        hpf = '247e-3 x ODR'
    elif bits[1] == 0x02:
        hpf = '62.084e-3 x ODR'
    elif bits[1] == 0x03:
        hpf = '15.545e-3 x ODR'
    elif bits[1] == 0x04:
        hpf = '3.862e-3 x ODR'
    elif bits[1] == 0x05:
        hpf = '0.954e-3 x ODR'
    else:
        hpf = '0.238e-3 x ODR'
    # Return values
    return [odr, lpf, hpf]

def bits2sync(bits):
    """Interprets the bits to know the type of synchronization
    Return:
        (str): external synchronization type
        (str): external clock enabled or not
    """
    # Interpret external clock

```

```

    if bits[0] == 0x00:
        ext_clk = 'External clock disabled'
    else:
        ext_clk = 'External clock enabled'
    # Interpret external sync control
    if bits[1] == 0x00:
        ext_sync = 'Internal sync'
    elif bits[1] == 0x01:
        ext_sync = 'External sync, no interpolation filter'
    else:
        ext_sync = 'External sync, interpolation filter'
    # Return value
    return [ext_clk, ext_sync]

def bits2accel(bits, scale, offset):
    """Transforms the accelartion value from bits to [g]
    Return
        (float): acceleration
    """
    x_data = bits[0] / scale[0] - offset[0]
    y_data = bits[1] / scale[1] - offset[1]
    z_data = bits[2] / scale[2] - offset[2]
    return [x_data, y_data, z_data]

def bits2temp(bits, scale, offset):
    """Transforms the temperature value from bits to [°C]
    Return:
        (float): temperature in [°C]
    """
    temperature = ((bits - offset) / scale) + 25.0
    return temperature

def bits2accel_temp(bits_accel, scale_accel, offset_accel, \
    bits_temp, scale_temp, offset_temp):
    """Transforms the accelerations and temperature bits to
    [g] and [°C]
    Return:
        (floats): acceleration and temperature
    """
    accel = bits2accel(bits_accel, scale_accel, offset_accel)

```

1.5. Allan Variance

This is the code used to analyse the signal of AXL355 accelerometer. A file containing the output of sensors, the accelerations and the temperature. For each axis the time signal is shown and the Allan Deviation is computed and analysed. The temperature signal is also plotted.

The code consists of three functions. The main program, mainAllan.m; Allan_AVAR_ADEV.m, which calculates the AVAR, ADEV and cluster times; and AnalyseAllanDev.m, which analyses the ADEV. These three functions are presented in the following subsections.

This code is written in Matlab 2017.

1.5.1. mainAllan.m

```

%% Analyses the noise of a triaxial accelerometer

clear all
close all

% Load file containing data of X, Y, Z axes and Temperature
file = 'data';
load(file);

% Classify data and convert units from [g] to m/s2
x_data = 9.81*data(:,1);
y_data = 9.81*data(:,2);
z_data = 9.81*data(:,3);
temp = data(:,4);

% Stablish sampling frequency and time vector
fs = 2; % Hz
dt = 1/fs;
timevec = dt*(0:(length(x_data)-1));

%% X axis
% Plot the signal
figure('Units', 'Normalized', 'OuterPosition', ...
    [0, 0.3, 0.7, 0.58]);
plot(timevec/3600,x_data);
title('X-axis','FontSize',20);
ylabel('Acceleration (^m/{s^2})','FontSize',16);
xlabel('Time (Hours)','FontSize',16);
xticks([0 6 12 18 24 30 36 42 48 54 60])
xlim([timevec(1)/3600 timevec(end)/3600]);
% Compute the Allan Variance and Deviation and Cluster times
[T_x, AVAR_x, ADEV_x] = Allan_AVAR_ADEV(x_data, dt);
% Interpret Allan Deviation
[noise_errors_x, Terrors_x, slopes_x, ADEV_x_found] = ...
    AnaliseAllanDev(T_x, ADEV_x, 'X');

%% Y axis
% Plot the signal
figure('Units', 'Normalized', 'OuterPosition', ...
    [0, 0.3, 0.7, 0.58]);
plot(timevec/3600,y_data);
title('Y-axis', 'FontSize',20);
ylabel('Acceleration (^m/{s^2})','FontSize',16);
xlabel('Time (Hours)','FontSize',16);
xticks([0 6 12 18 24 30 36 42 48 54 60])
xlim([timevec(1)/3600 timevec(end)/3600]);
% Compute the Allan Variance and Deviation and Cluster times
[T_y, AVAR_y, ADEV_y] = Allan_AVAR_ADEV(y_data, dt);
% Interpret Allan Deviation
[noise_errors_y, Terrors_y, slopes_y, ADEV_y_found] = ...
    AnaliseAllanDev(T_y, ADEV_y, 'Y');

%% Z axis
% Plot the signal
figure('Units', 'Normalized', 'OuterPosition', ...
    [0, 0.3, 0.7, 0.58]);
plot(timevec/3600,z_data);
title('Z-axis','FontSize',20);
ylabel('Acceleration (^m/{s^2})','FontSize',16);

```

```

xlabel('Time (Hours)', 'FontSize', 16);
xticks([0 6 12 18 24 30 36 42 48 54 60])
xlim([timevec(1)/3600 timevec(end)/3600]);
% Compute the Allan Variance and Deviation and Cluster times
[T_z, AVAR_z, ADEV_z] = Allan_AVAR_ADEV(z_data, dt);
% Interpret Allan Deviation
[noise_errors_z, Terrors_z, slopes_z, ADEV_z_found] = ...
    AnalyseAllanDev(T_z, ADEV_z, 'Z');

%% Temperature
% Plot the signal
figure('Units', 'Normalized', 'OuterPosition', ...
    [0, 0.3, 0.7, 0.58]);
plot(timevec/3600, temp);
title('Temperature', 'FontSize', 20);
ylabel('Temperature (°C)', 'FontSize', 16);
xlabel('Time (Hours)', 'FontSize', 16);
xticks([0 6 12 18 24 30 36 42 48 54 60])
xlim([timevec(1)/3600 timevec(end)/3600]);

```

1.5.2. Allan_AVAR_ADEV.m

```

function [T, AVAR, ADEV] = Allan_AVAR_ADEV(acc, dt)
% Computes the Allan Variance, Deviation and averaging times
% for an accelerometer output
%
% Inputs:
%   - acc: vector containing the output acceleration.
%   - dt: sampling period of the signal
% Outputs:
%   - T: vector containing the averaging times
%   - AVAR: Allan Variance
%   - ADEV: Allan Deviation
%
% Determine total number of points
N = length(acc);
% Integrate acceleration
vel = cumsum(acc)*dt;

% Compute the clusters size
max_n = N/2 - 1; % Maximum cluster size
max_n_log = log10(max_n); % log10 of the maximum cluster size
% Vector of cluster sizes, spaced logarithmically
n = unique(floor(logspace(0, max_n_log, N)))';

% Compute clusters
T = n.*dt;
% Preallocate AVAR vector
AVAR = zeros(length(T), 1);

% Compute AVAR for every cluster size
for i = 1:length(T)
    for k = 1:(N-2*n(i))
        AVAR(i) = AVAR(i) + (vel(k+2*n(i)) - 2*vel(k+n(i))...
            + vel(k))^2;
    end
end
AVAR = AVAR./(2.*(T.^2) .* (N - 2.*n));

```

```
ADEV = sqrt(AVAR);
```

```
end
```

1.5.3. AnalyseAllanDev.m

```
function [noise_errors, Terrors, slopes, ADEVb] = ...
    AnalyseAllanDev(T, ADEV, axis)
%
% Analyses the Allan Deviation of an accelerometer
%
% Inputs:
%   - T: vector containing averaging times
%   - ADEV: vector containing the Allan Deviation values
%   - axis: string with the axis name
%
% Compute the logarithmic slopes of the ADEV curve
logT = log10(T);
logADEV = log10(ADEV);
slopes = diff(logADEV)./diff(logT);
% Smooth slopes curve
for i = 1:20
    slopes = smooth(slopes);
end
% Slopes at which Noise errors are found
slopes_of_int = [-1, -1/2, 0, +1/2, +1];
% Averaging times at which Noise errors are found
T_of_int = [sqrt(3), 1, NaN, 3, sqrt(2)];
% Preallocate vectors
noise_errors = zeros(length(slopes_of_int), 1);
b = zeros(length(slopes_of_int), 1);
Terrors = zeros(length(slopes_of_int), 1);
ADEVb = zeros(length(slopes_of_int), 1);
% Find the Noise errors
n = 1;
i = 1;
while (i <= length(slopes) && n <= length(slopes_of_int))
    if (slopes(i) >= slopes_of_int(n))
        if isnan(T_of_int(n))
            % Determine Noise error value
            b(n) = log10(ADEV(i));
            noise_errors(n) = ADEV(i)/sqrt(2*log(2)/pi);
        else
            % Determine Noise error value
            b(n) = logADEV(i) - slopes_of_int(n)*logT(i);
            noise_errors(n) = 10.^(slopes_of_int(n)*...
                log10(T_of_int(n)) + b(n));
        end
        % Value at which the Noise error is found
        ADEVb(n) = ADEV(i);
        % Averaging time at which the Noise error is found
        Terrors(n) = T(i);
        n = n + 1;
    end
end
```

```

    end
    i = i + 1;
end

% Plot the logarithmic slope of the Allan Deviation
figure();
semilogx(T(1:end-1),slopes);
hold on;
semilogx(Terrors,slopes_of_int,'ks','MarkerSize',12);
semilogx([T(1) T(end-1)],[-1 -1],'Color',[0.7 0.7 0.7]);
semilogx([T(1) T(end-1)],[-1/2 -1/2],'Color',[0.7 0.7 0.7]);
semilogx([T(1) T(end-1)],[0 0],'Color',[0.7 0.7 0.7]);
semilogx([T(1) T(end-1)],[1/2 1/2],'Color',[0.7 0.7 0.7]);
semilogx([T(1) T(end-1)],[1 1],'Color',[0.7 0.7 0.7]);
title(['Slopes of Allan Deviation',axis,'-axis']);
xlabel('Cluster Time T (s)');
ylabel('^{\sigma}/_{dT} (^m/_{s^2})');
xlim([T(1) T(end-1)]);
ylim([-5 5]);

% Determine points to plot slopes
slope_points(:,1) = 10.^(slopes_of_int'.*log10(T(1)) + b);
slope_points(:,2) = 10.^(slopes_of_int'.*log10(T(end)) + b);
T_lines = repmat([T(1) T(end)],length(slopes_of_int),1);

% Plot Allan Deviation with the slopes
figure('Units', 'Normalized', 'OuterPosition', ...
    [0, 0.3, 0.7, 0.58]);
loglog(T,ADEV,'LineWidth',2);
hold on;
loglog(T_lines',slope_points');
xlim([T(1) T(end-1)]);
ylim([min(ADEV)/2, 2*max(ADEV)]);
title(['Allan Deviation of ',axis,'-axis'],'FontSize',20);
xlabel('Cluster time T (s)','FontSize',20);
ylabel('\sigma (^m/_{s^2})','FontSize',16);
grid on

end

```

Annex 2. Seismic signals

In this annex all seismic signals obtained from CGAR, VILA and BAIN seismographs are presented.

2.1. CGAR seismograph

2.1.1. 1st of February 2018

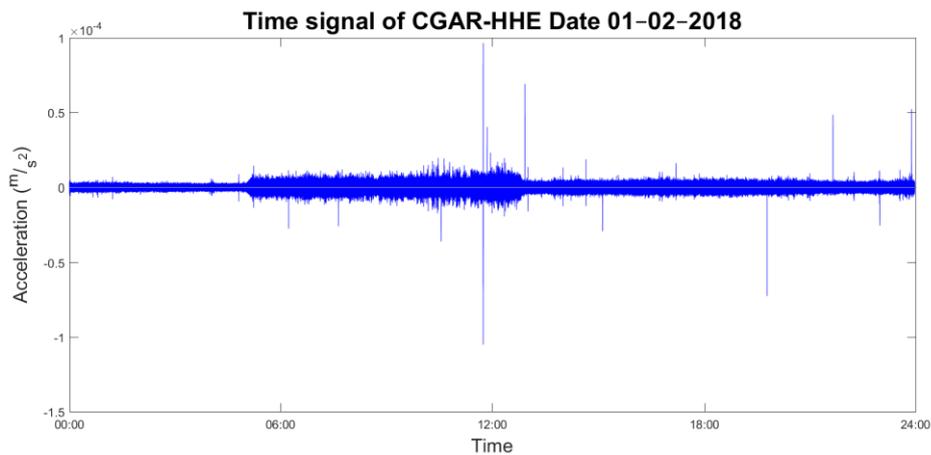


Figure A2.1. Seismic time signal for the HHE axis of CGAR seismograph during the 1st of February 2018

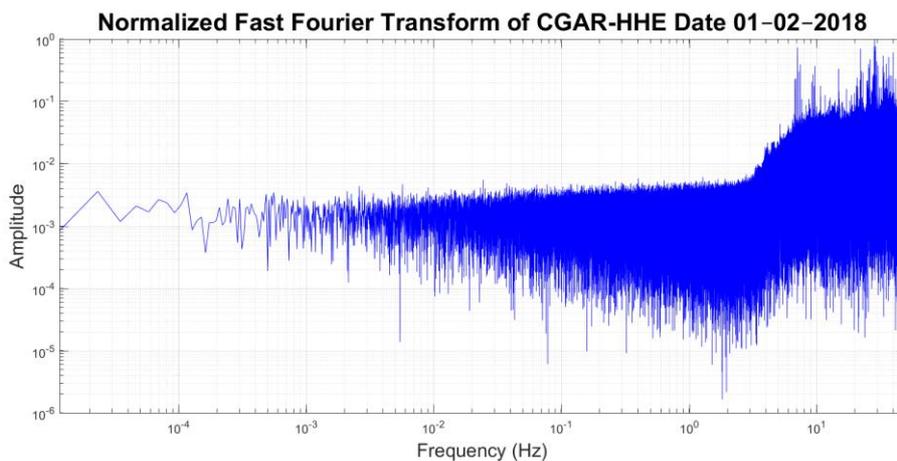


Figure A2.2. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 1st of February 2018

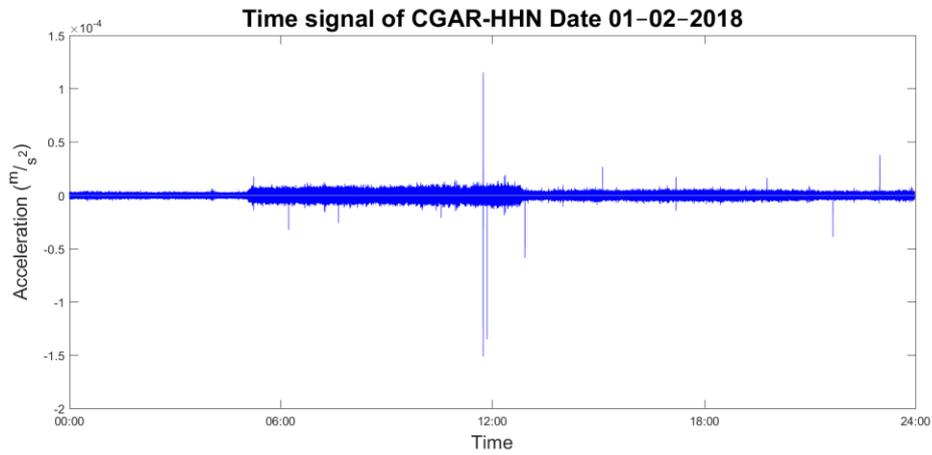


Figure A2.3. Seismic time signal for the HHN axis of CGAR seismograph during the 1st of February 2018

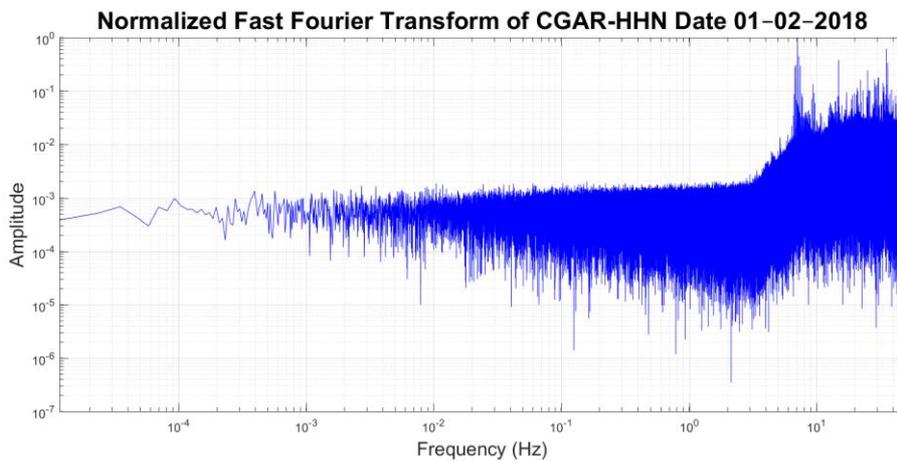


Figure A2.4. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 1st of February 2018

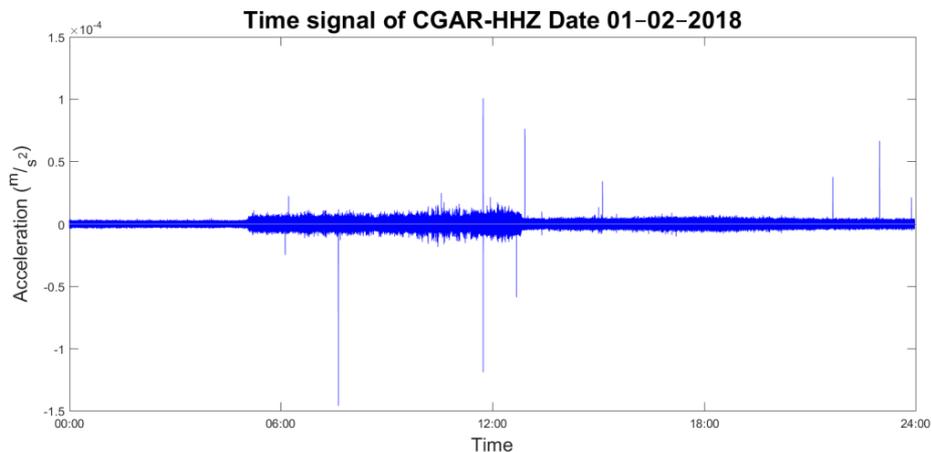


Figure A2.5. Seismic time signal for the HHZ axis of CGAR seismograph during the 1st of February 2018

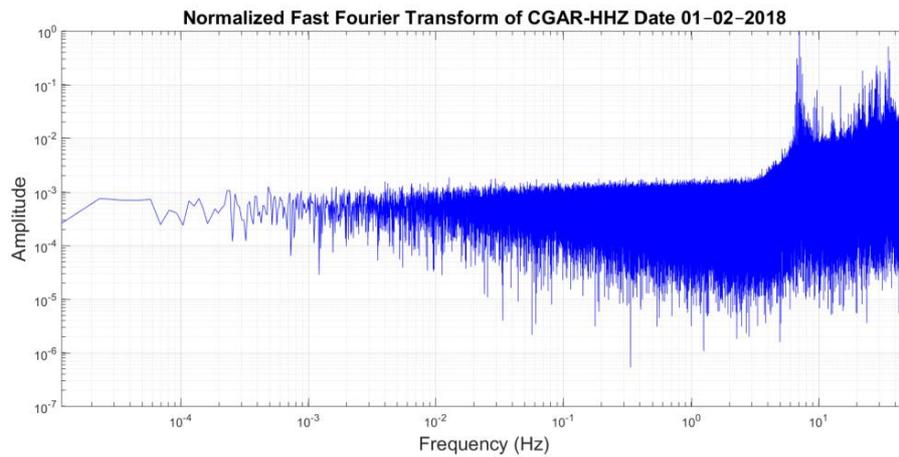


Figure A2.6. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 1st of February 2018

2.1.2. 11th of February 2018

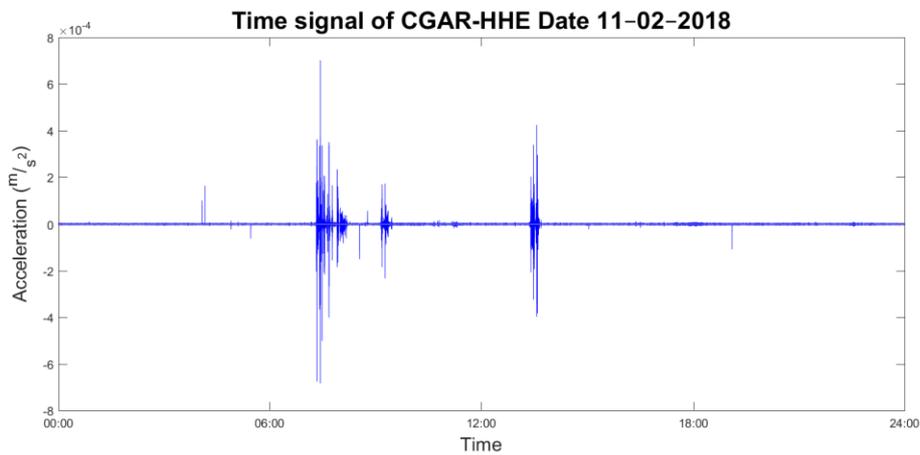


Figure A2.7. Seismic time signal for the HHE axis of CGAR seismograph during the 11th of February 2018

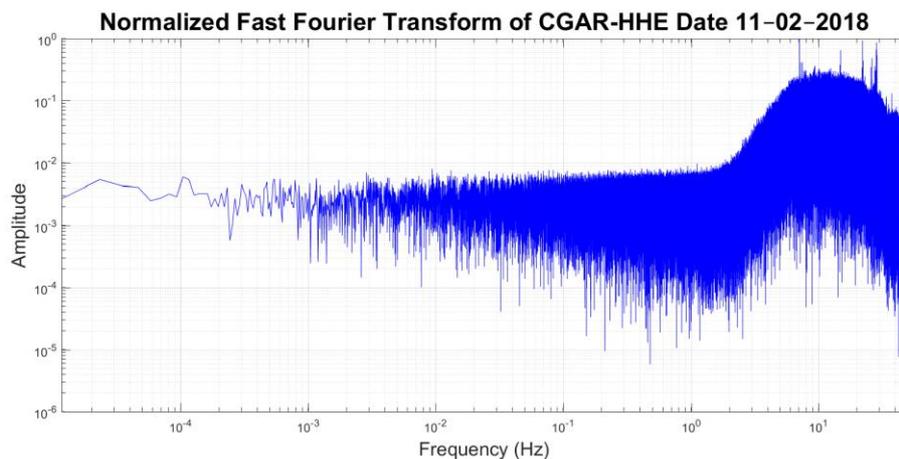


Figure A2.8. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 11th of February 2018

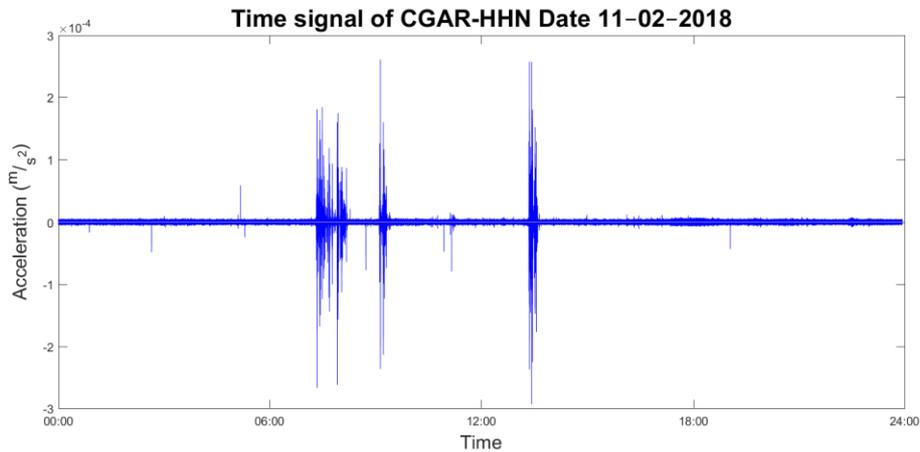


Figure A2.9. Seismic time signal for the HHN axis of CGAR seismograph during the 11th of February 2018

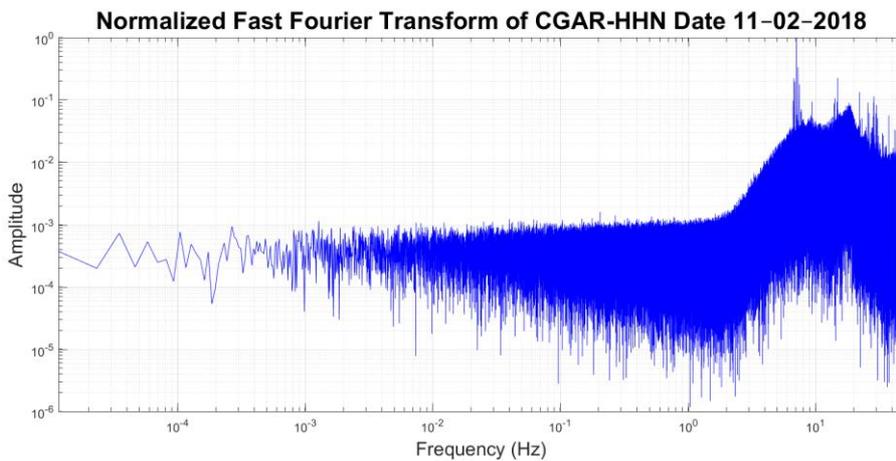


Figure A2.10. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 11th of February 2018

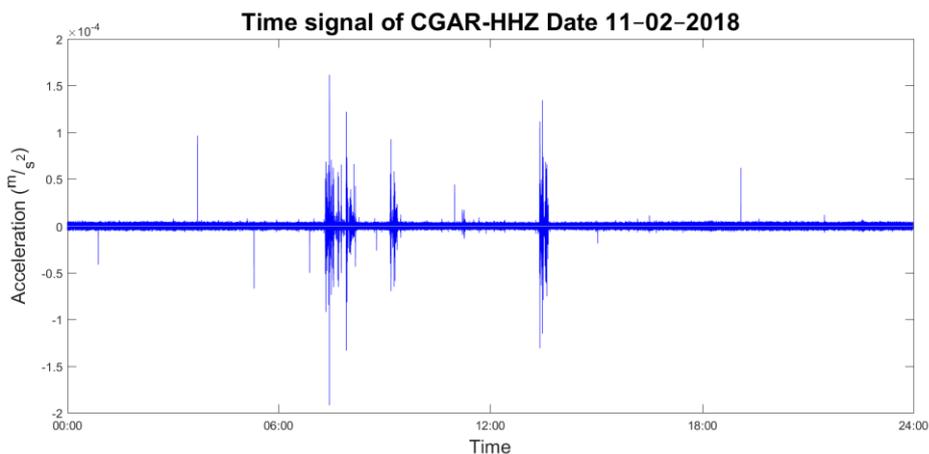


Figure A2.11. Seismic time signal for the HHZ axis of CGAR seismograph during the 11th of February 2018

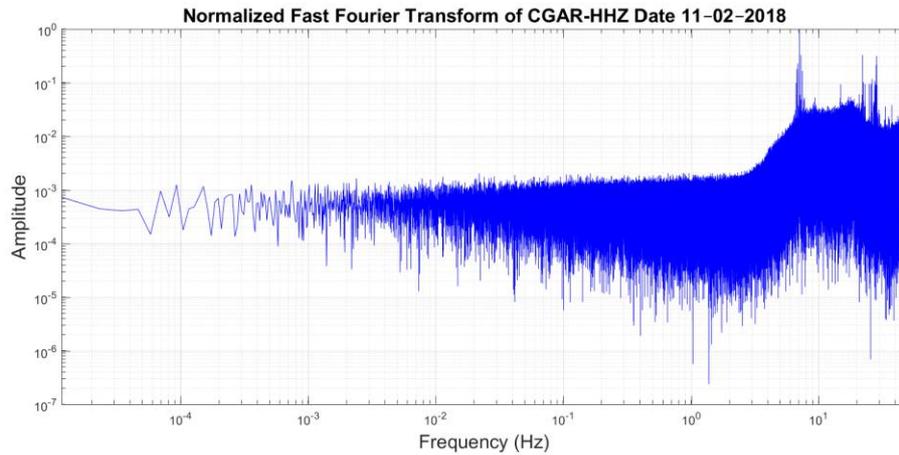


Figure A2.12. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 11th of February 2018

2.1.3. 21st of February 2018

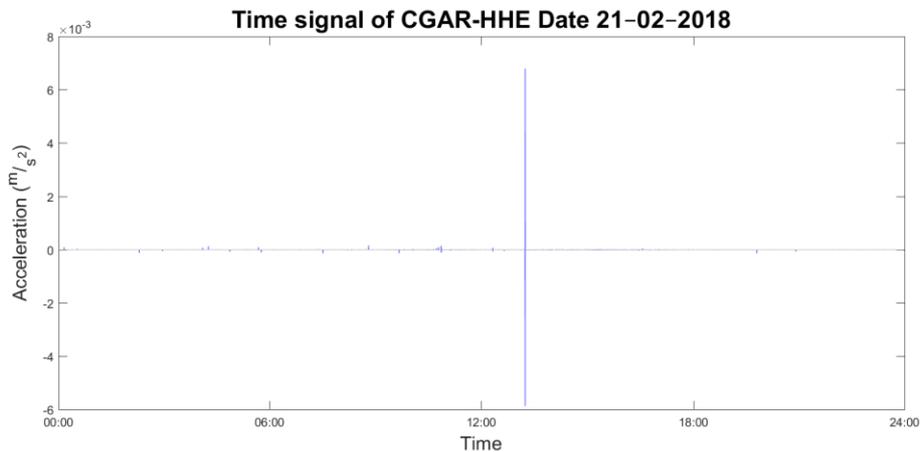


Figure A2.13. Seismic time signal for the HHE axis of CGAR seismograph during the 21st of February 2018

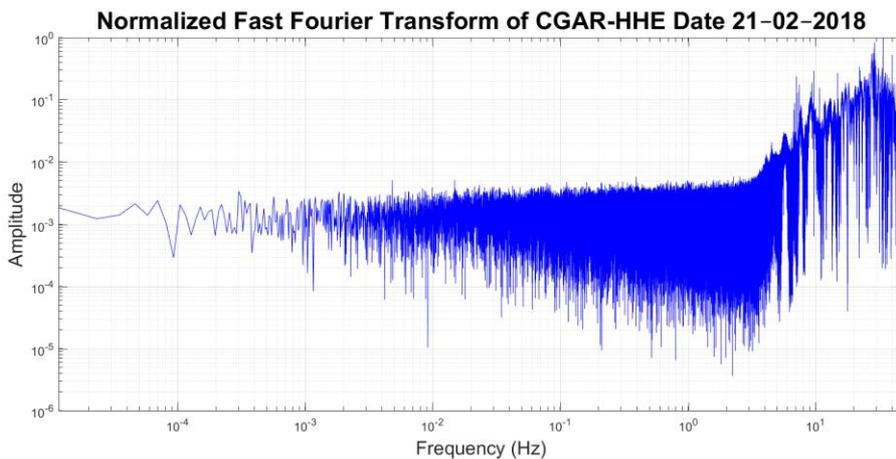


Figure A2.14. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 21st of February 2018

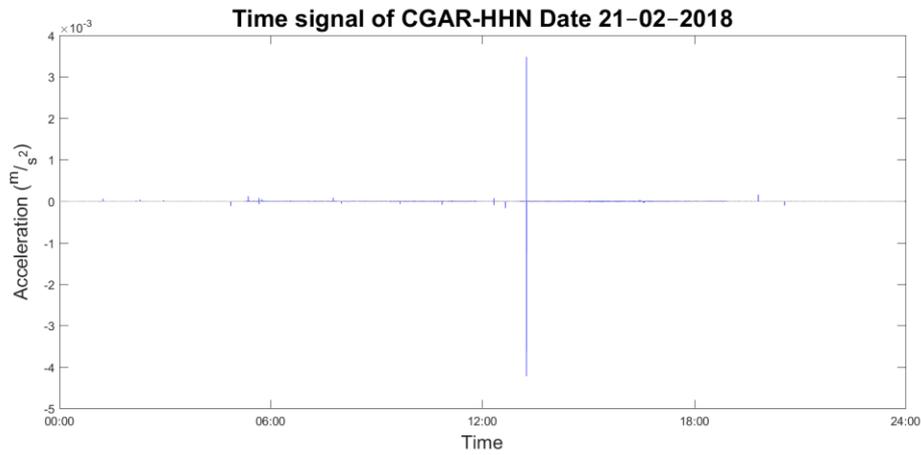


Figure A2.15. Seismic time signal for the HHN axis of CGAR seismograph during the 21st of February 2018

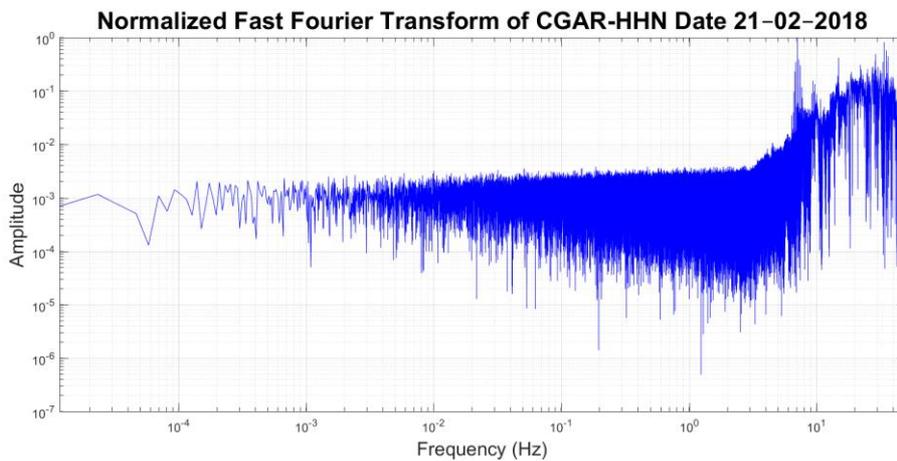


Figure A2.16. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 21st of February 2018

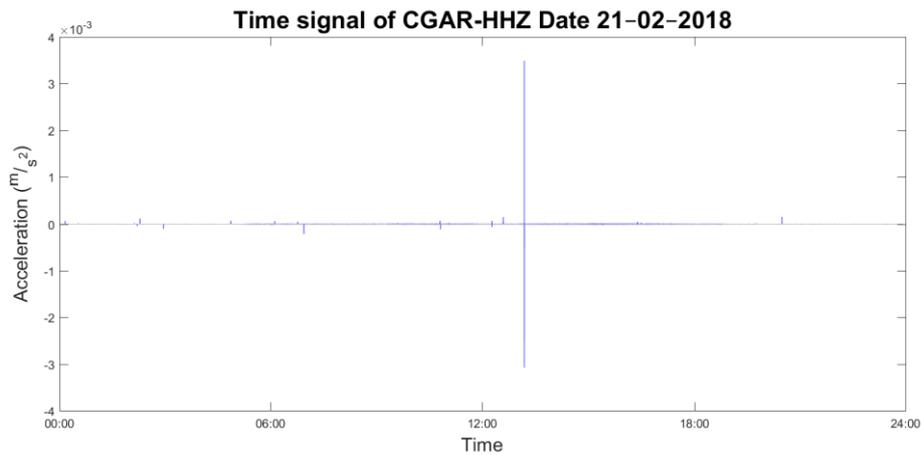


Figure A2.17. Seismic time signal for the HHZ axis of CGAR seismograph during the 21st of February 2018

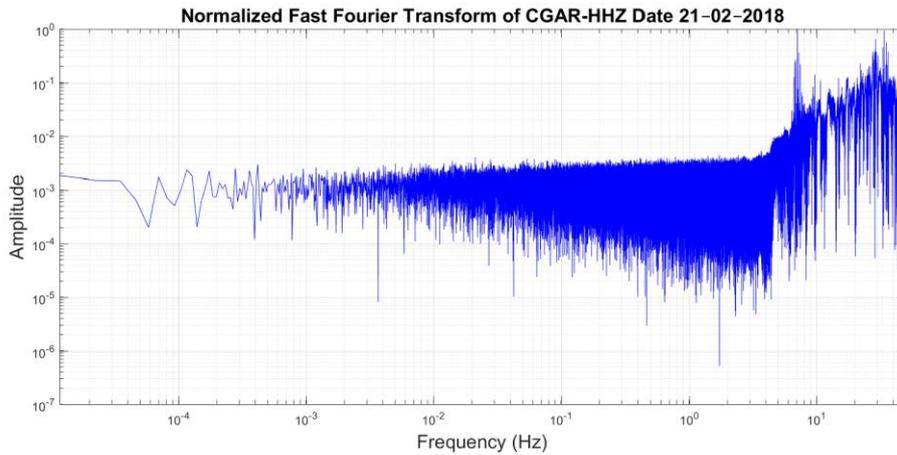


Figure A2.18. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 21st of February 2018

2.1.4. 1st of March 2018

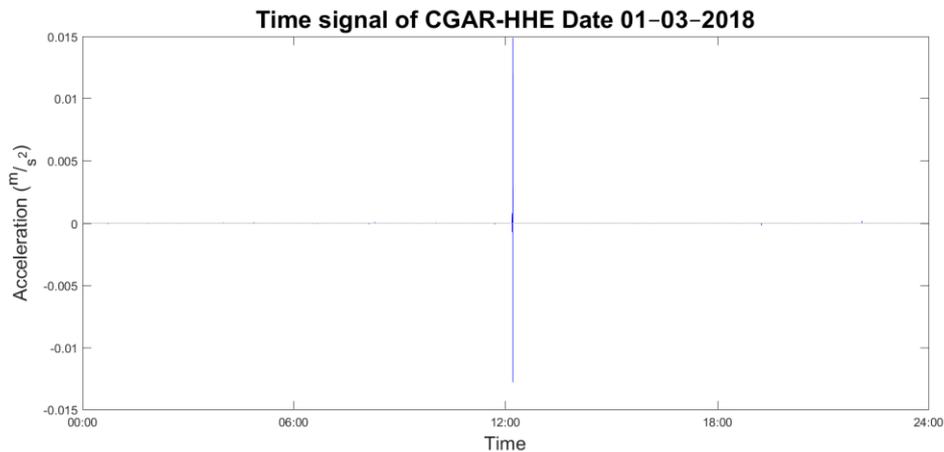


Figure A2.19. Seismic time signal for the HHE axis of CGAR seismograph during the 1st of March 2018

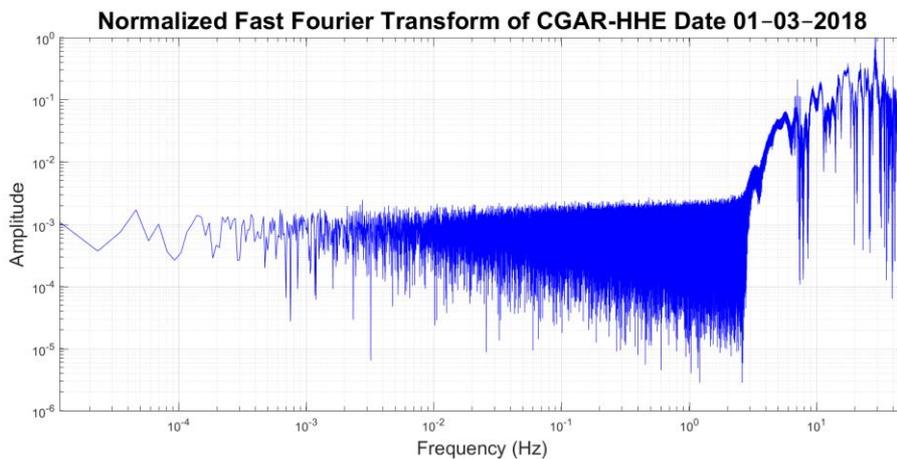


Figure A2.20. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 1st of March 2018

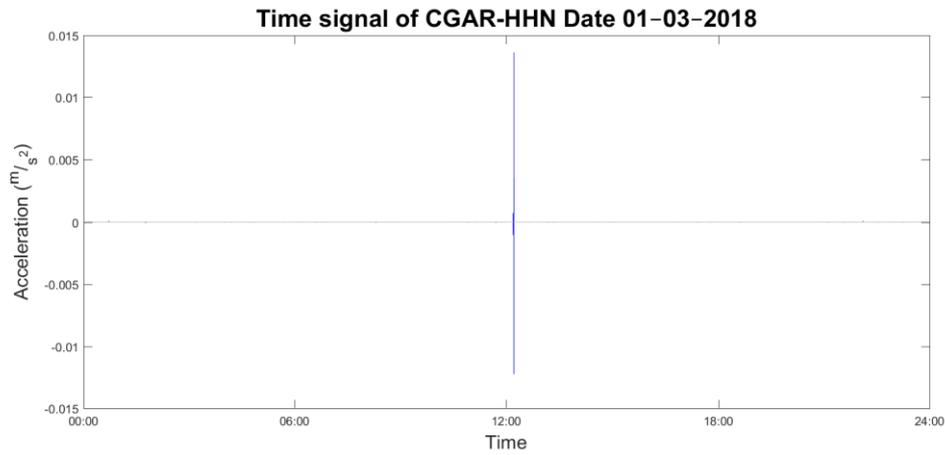


Figure A2.21. Seismic time signal for the HHN axis of CGAR seismograph during the 1st of March 2018

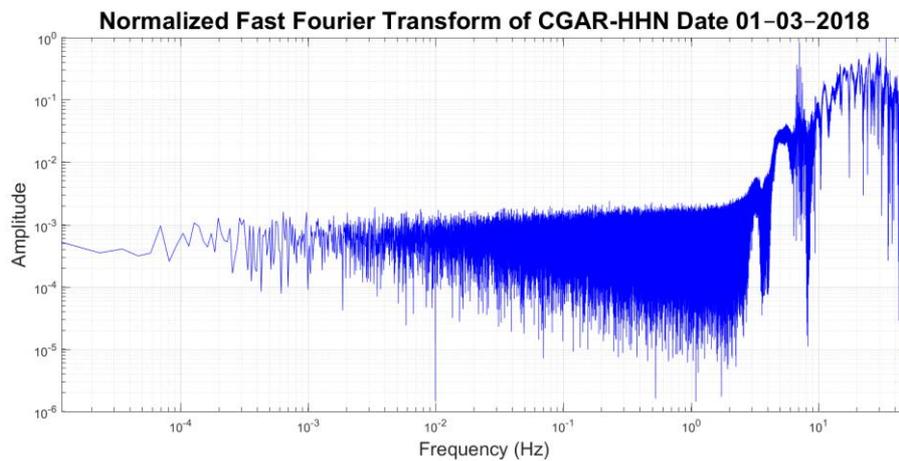


Figure A2.22. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 1st of March 2018

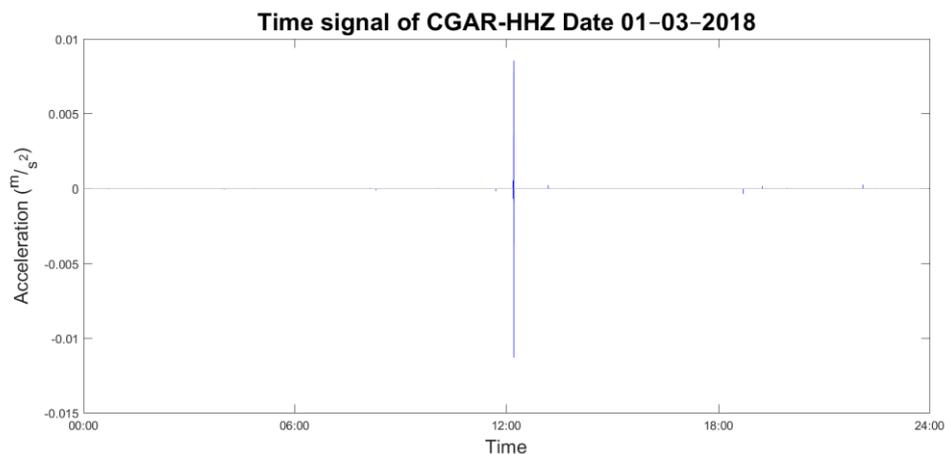


Figure A2.23. Seismic time signal for the HHZ axis of CGAR seismograph during the 1st of March 2018

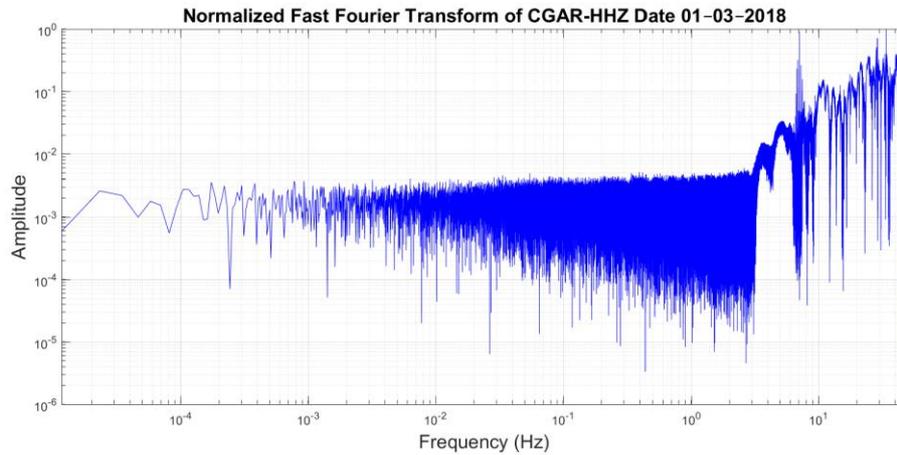


Figure A2.24. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 1st of March 2018

2.1.5. 11th of March 2018

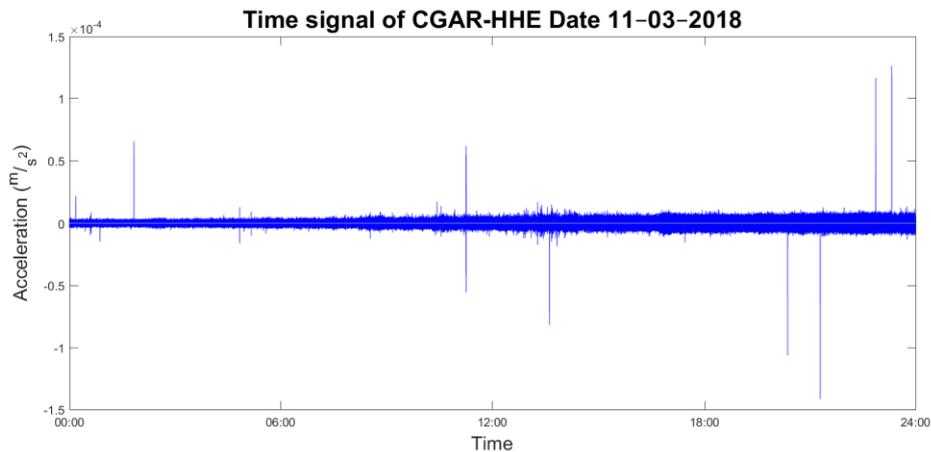


Figure A2.25. Seismic time signal for the HHE axis of CGAR seismograph during the 11th of March 2018

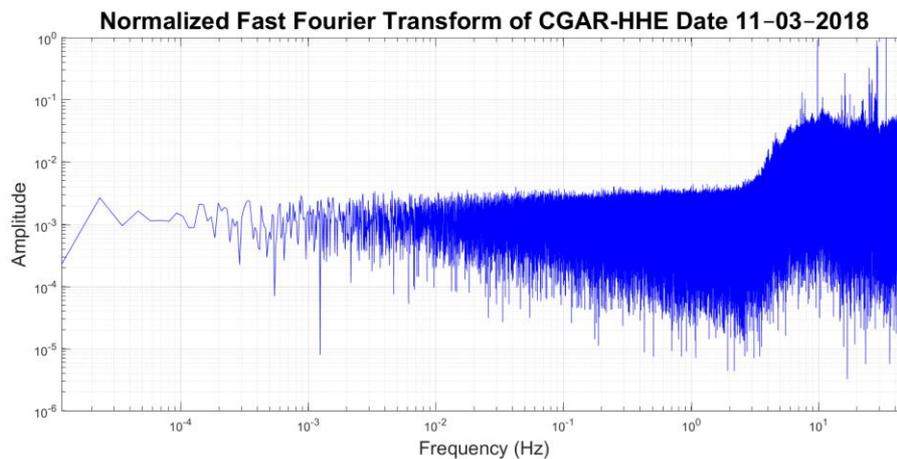


Figure A2.26. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of CGAR seismograph during the 11th of March 2018

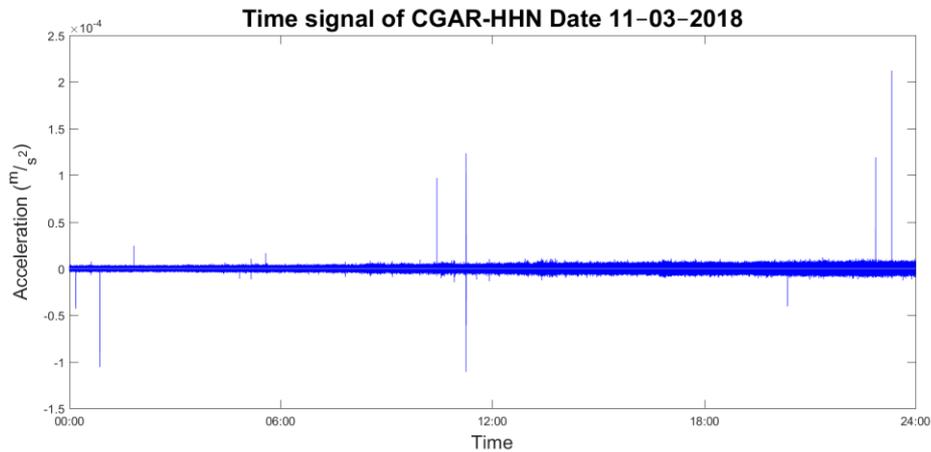


Figure A2.27. Seismic time signal for the HHN axis of CGAR seismograph during the 11th of March 2018

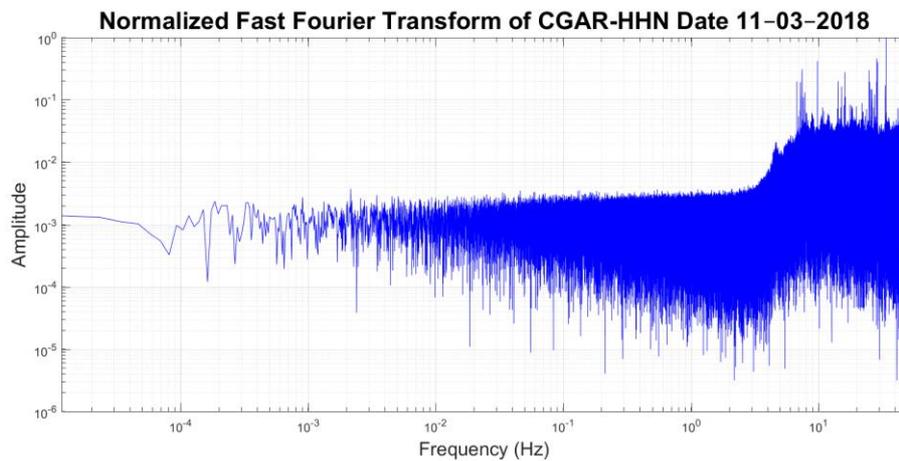


Figure A2.28. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of CGAR seismograph during the 11th of March 2018

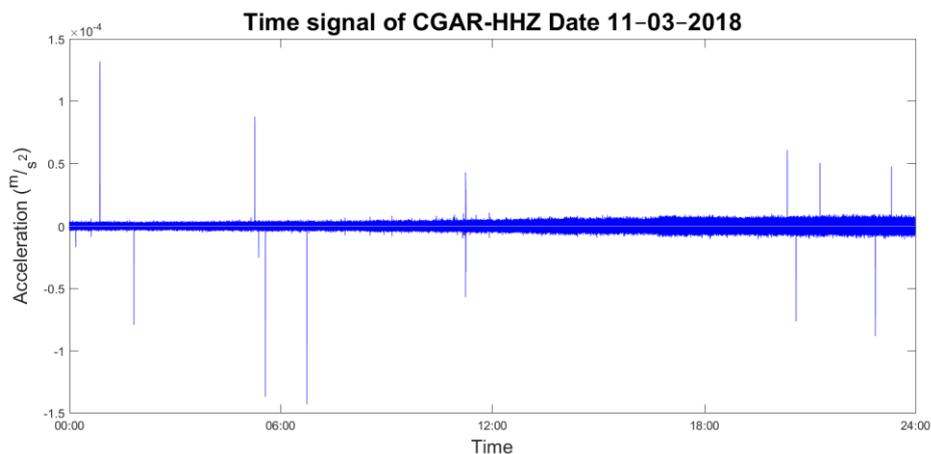


Figure A2.29. Seismic time signal for the HHZ axis of CGAR seismograph during the 11th of March 2018

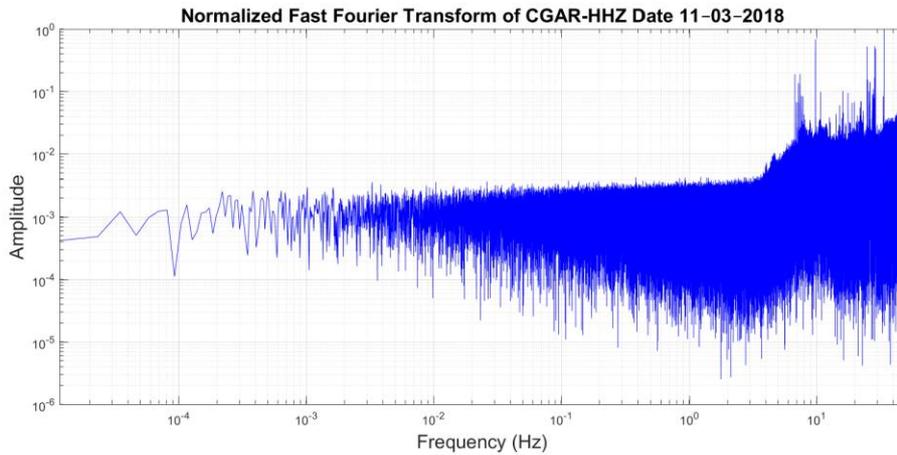


Figure A2.30. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of CGAR seismograph during the 11th of March 2018

2.2. VILA seismograph

2.2.1. 1st of February 2018

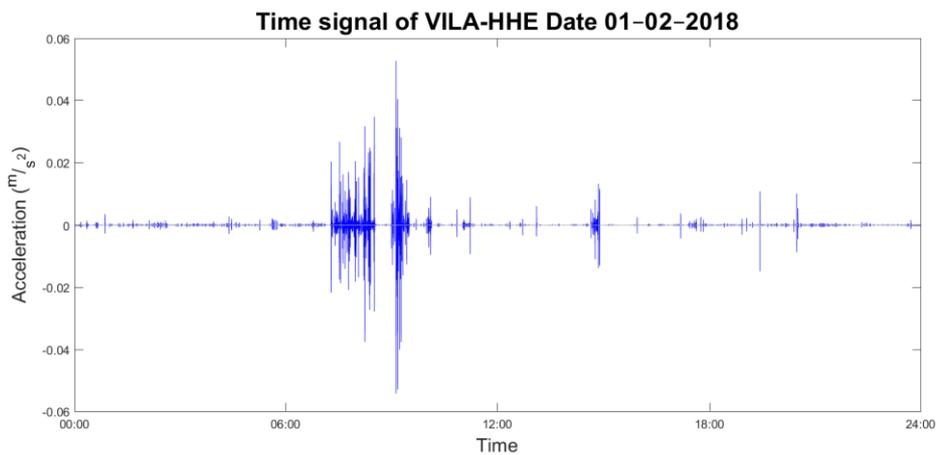


Figure A2.31. Seismic time signal for the HHE axis of VILA seismograph during the 1st of February 2018

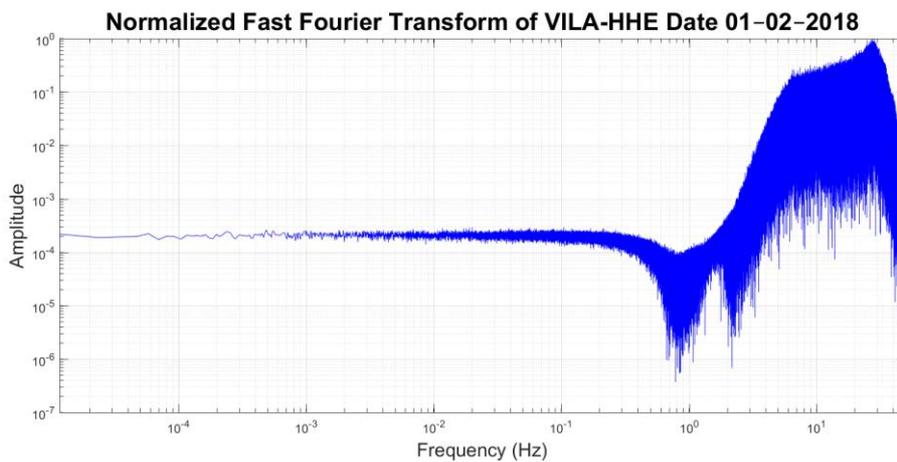


Figure A2.32. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 1st of February 2018

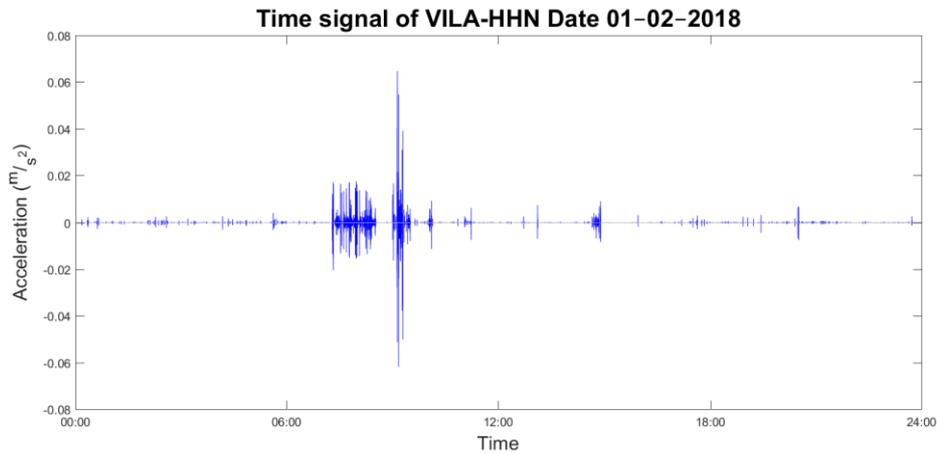


Figure A2.33. Seismic time signal for the HHN axis of VILA seismograph during the 1st of February 2018

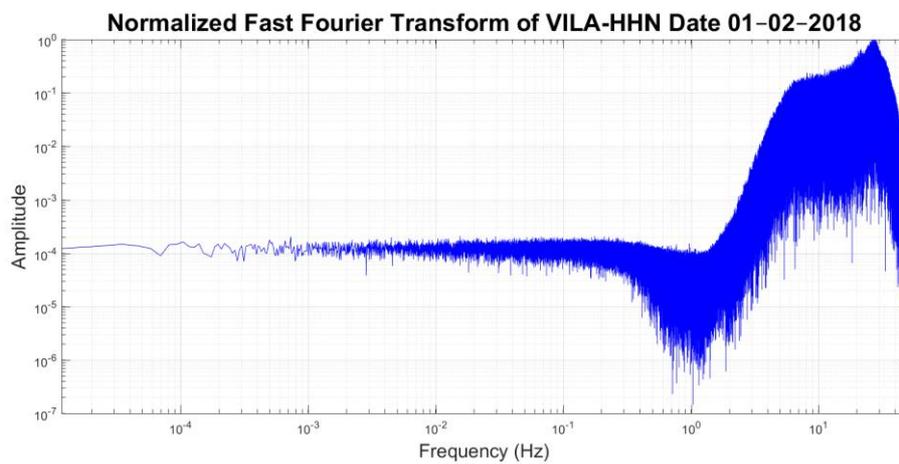


Figure A2.34. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 1st of February 2018

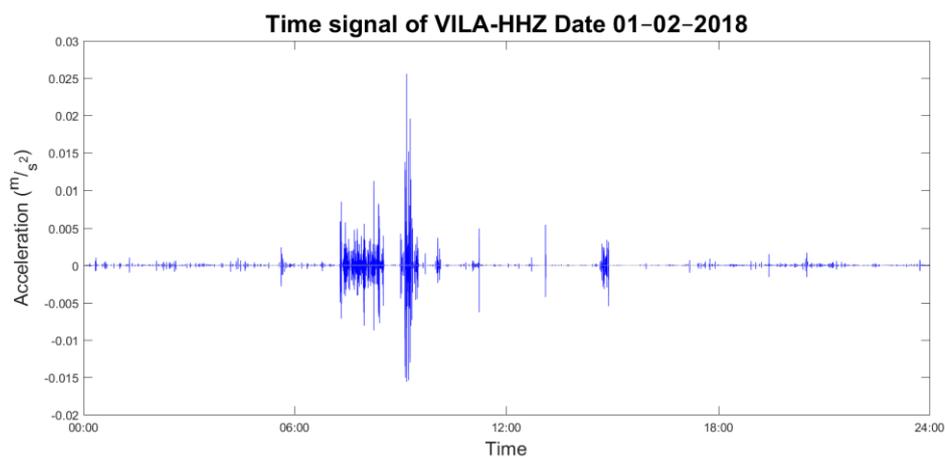


Figure A2.35. Seismic time signal for the HHZ axis of VILA seismograph during the 1st of February 2018

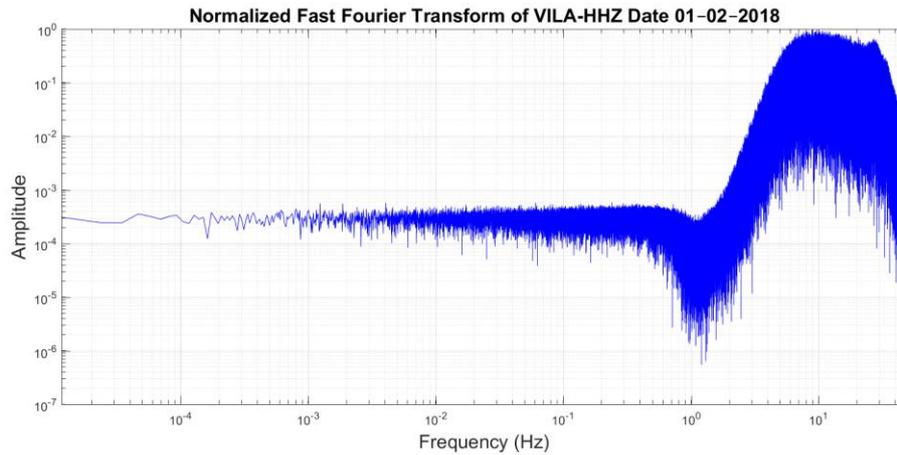


Figure A2.36. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 1st of February 2018

2.2.2. 11th of February 2018

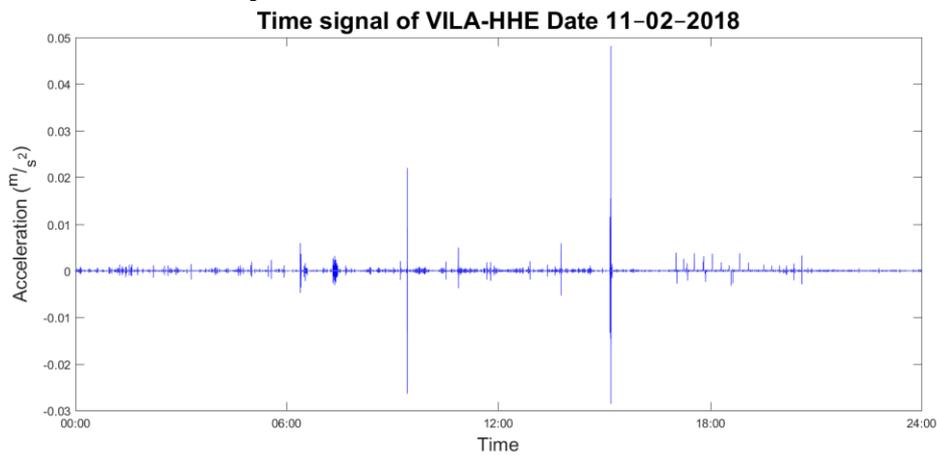


Figure A2.37. Seismic time signal for the HHE axis of VILA seismograph during the 11th of February 2018

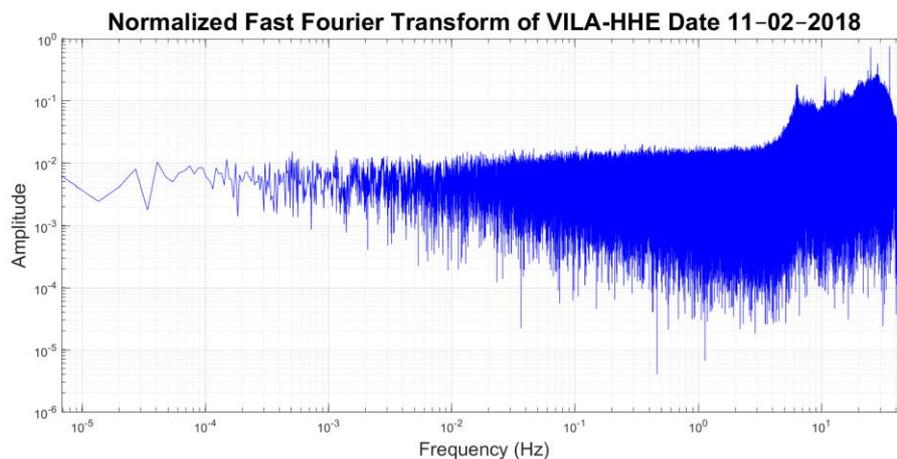


Figure A2.38. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 11th of February 2018

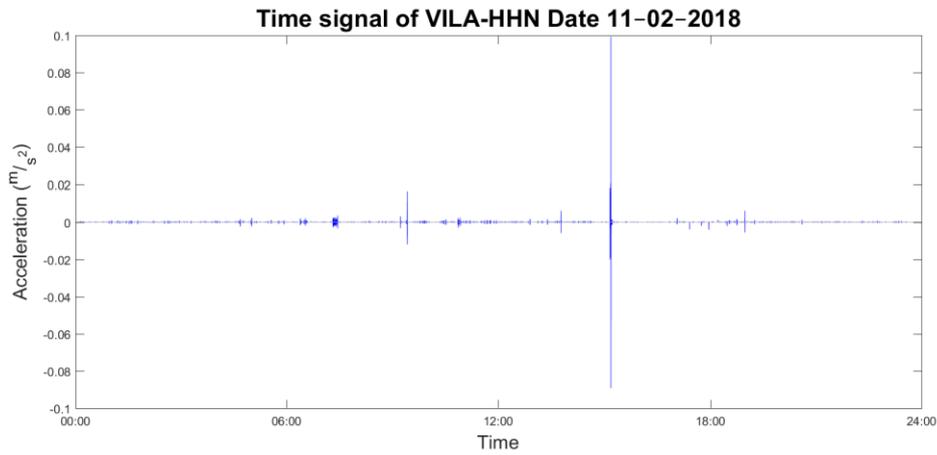


Figure A2.39. Seismic time signal for the HHN axis of VILA seismograph during the 11th of February 2018

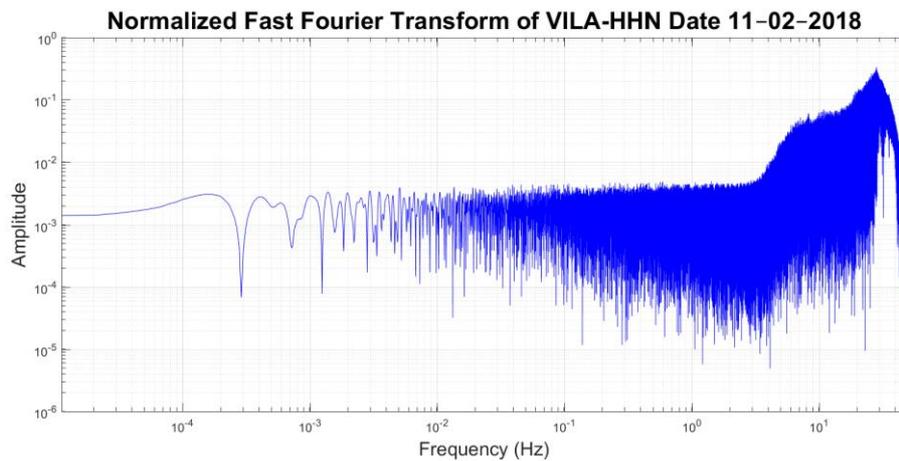


Figure A2.40. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 11th of February 2018

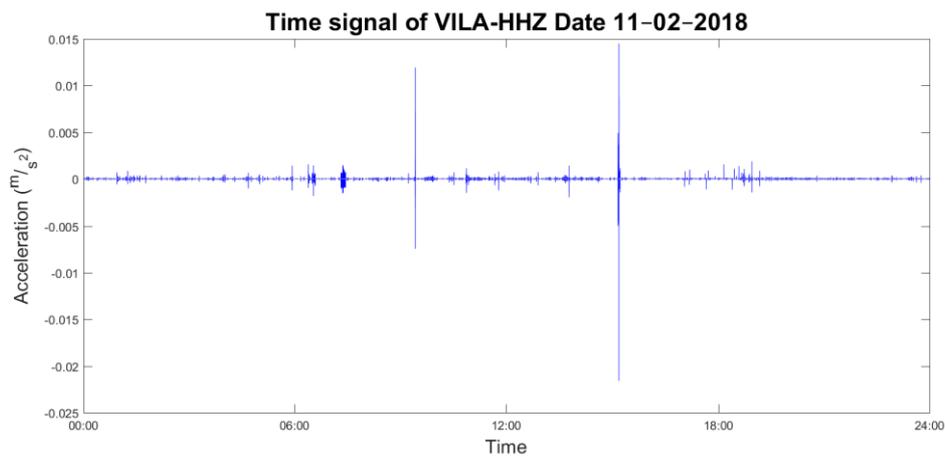


Figure A2.41. Seismic time signal for the HHZ axis of VILA seismograph during the 11th of February 2018

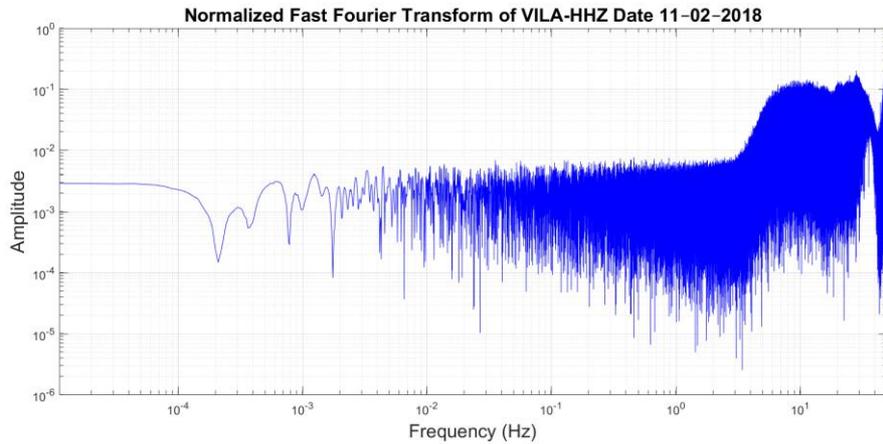


Figure A2.42. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 11th of February 2018

2.2.3. 15th of February 2018

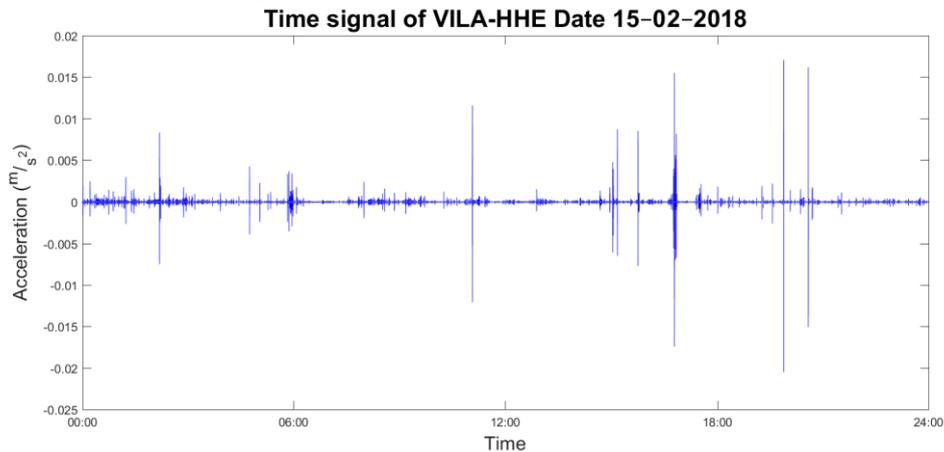


Figure A2.43. Seismic time signal for the HHE axis of VILA seismograph during the 15th of February 2018

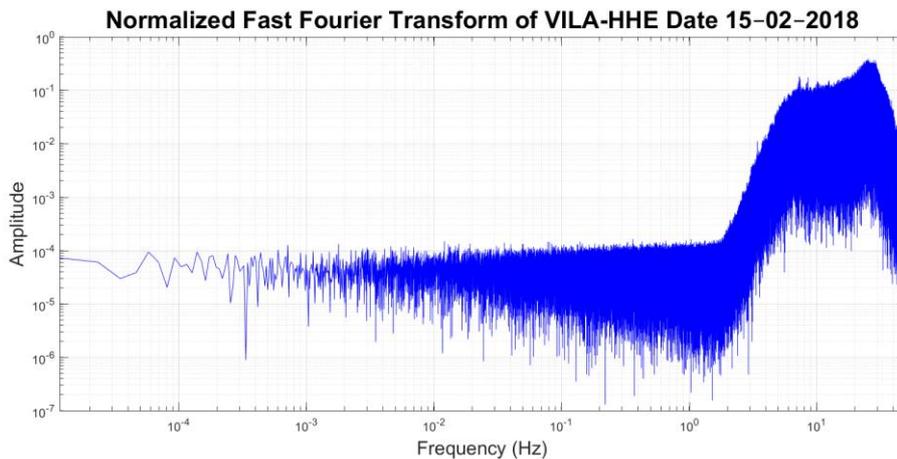


Figure A2.44. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 15th of February 2018

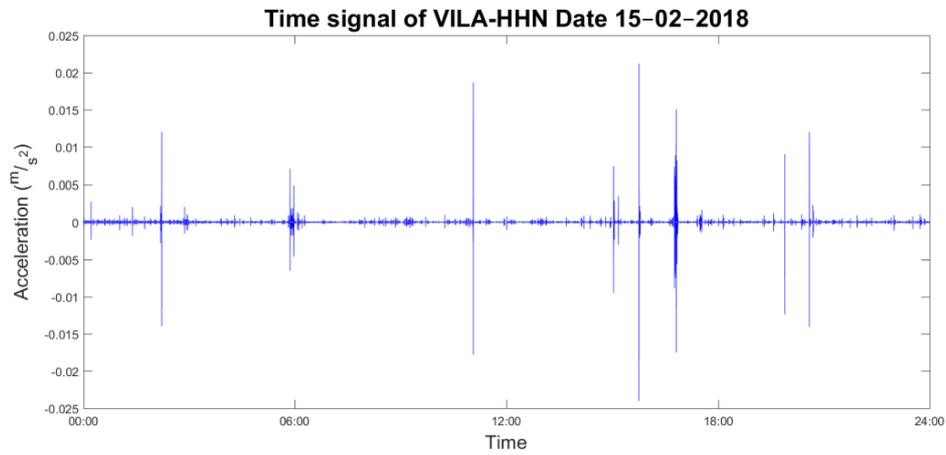


Figure A2.45. Seismic time signal for the HHN axis of VILA seismograph during the 15th of February 2018

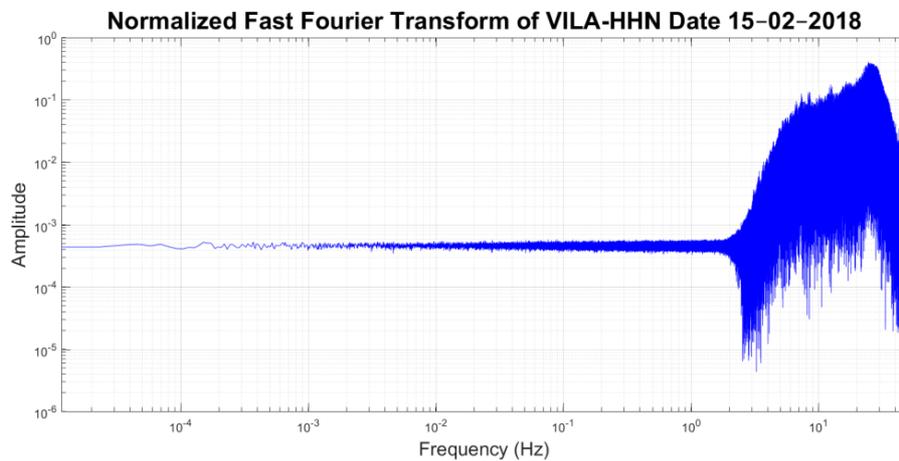


Figure A2.46. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 15th of February 2018

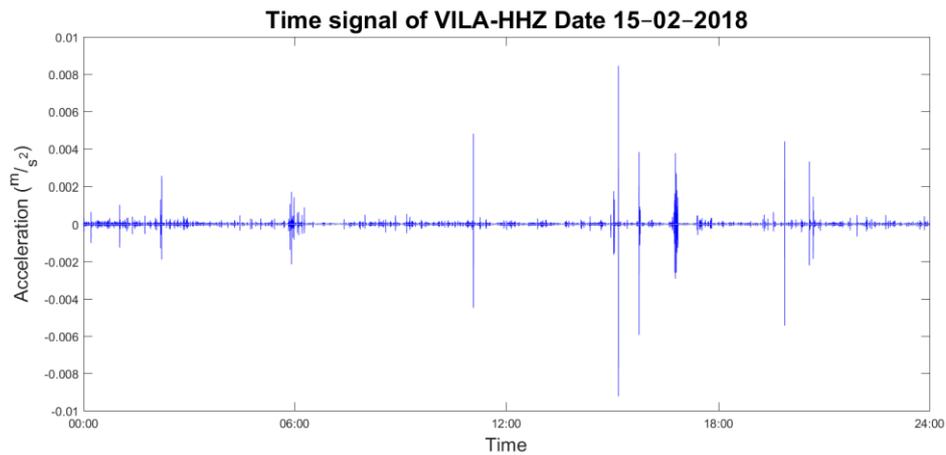


Figure A2.47. Seismic time signal for the HHZ axis of VILA seismograph during the 15th of February 2018

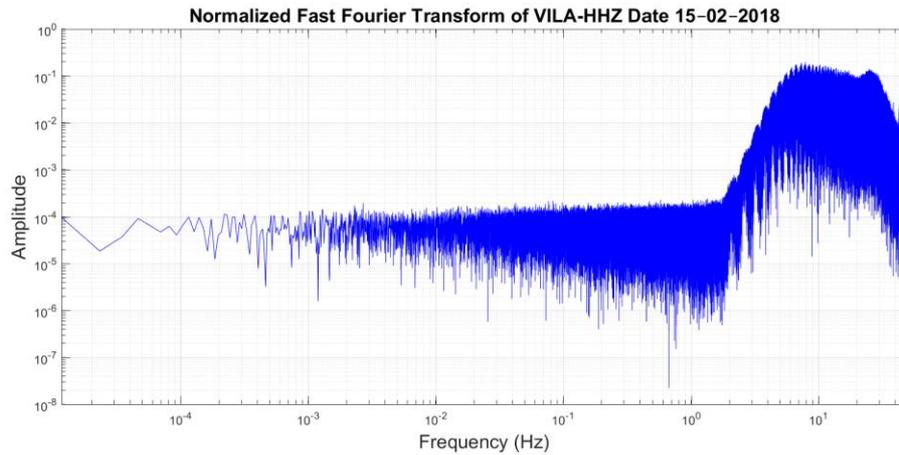


Figure A2.48. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 15th of February 2018

2.2.4. 6th of March 2018

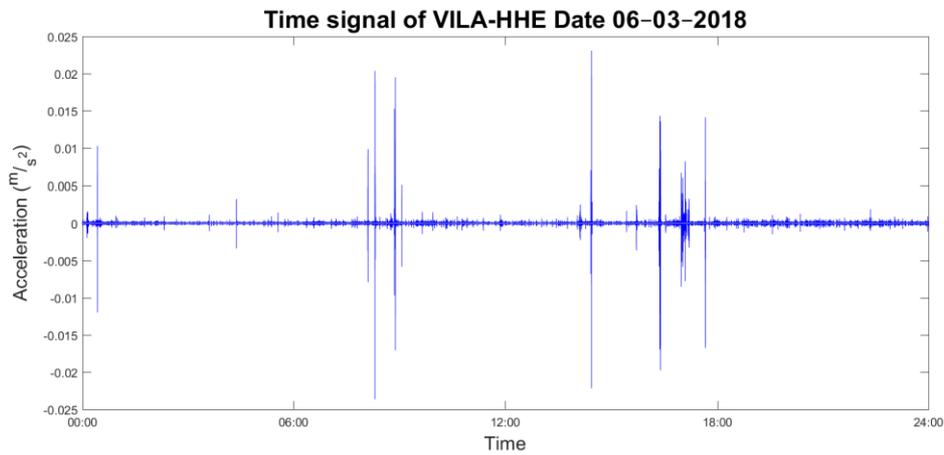


Figure A2.49. Seismic time signal for the HHE axis of VILA seismograph during the 6th of March 2018

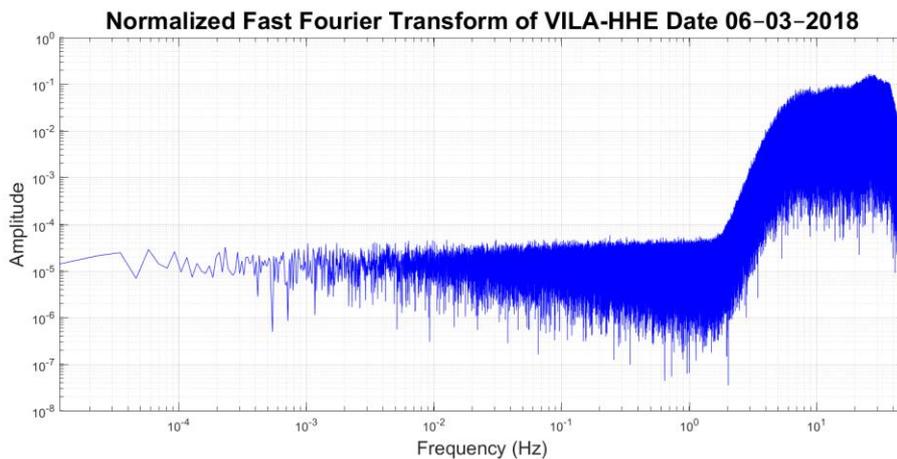


Figure A2.50. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 6th of March 2018

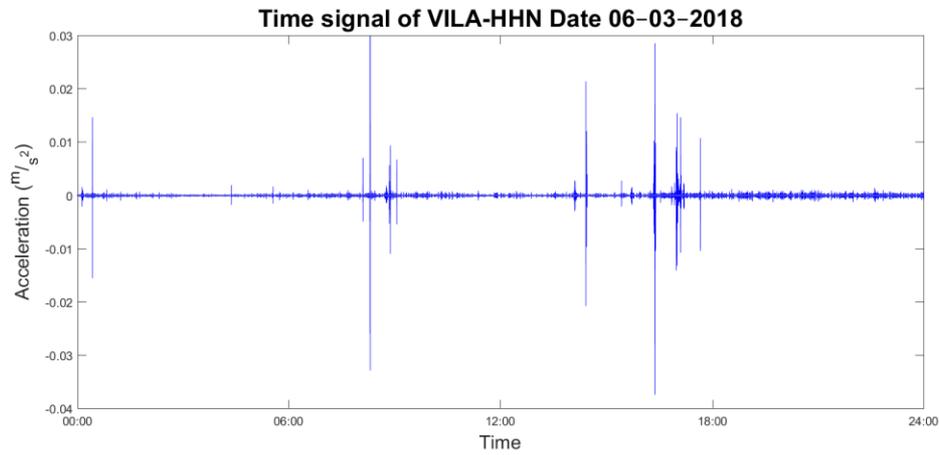


Figure A2.51. Seismic time signal for the HHN axis of VILA seismograph during the 6th of March 2018

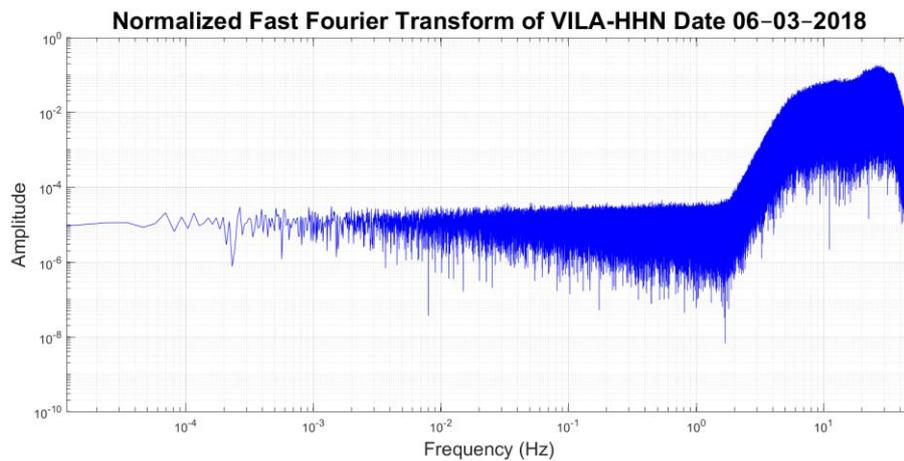


Figure A2.52. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 6th of March 2018

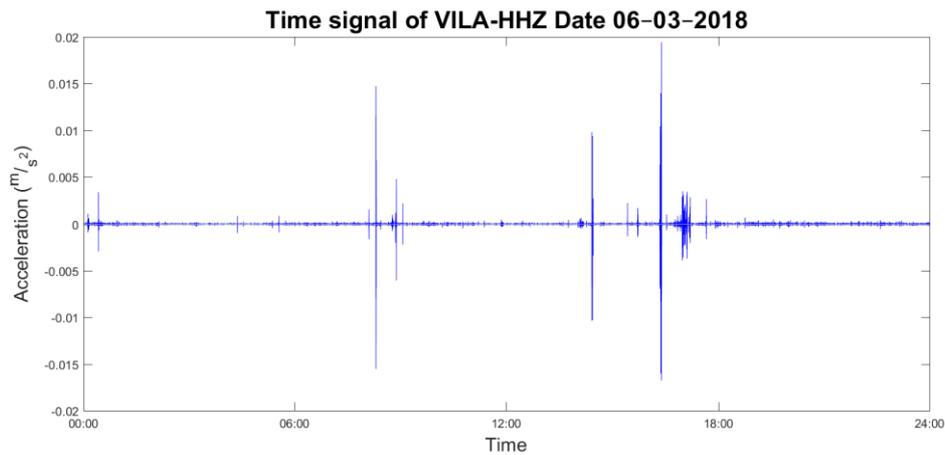


Figure A2.53. Seismic time signal for the HHZ axis of VILA seismograph during the 6th of March 2018

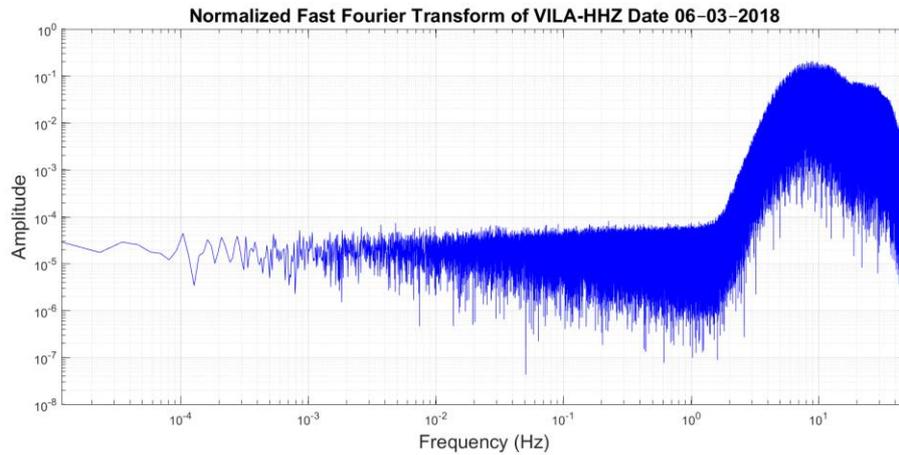


Figure A2.54. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 6th of March 2018

2.2.5. 11th of March 2018

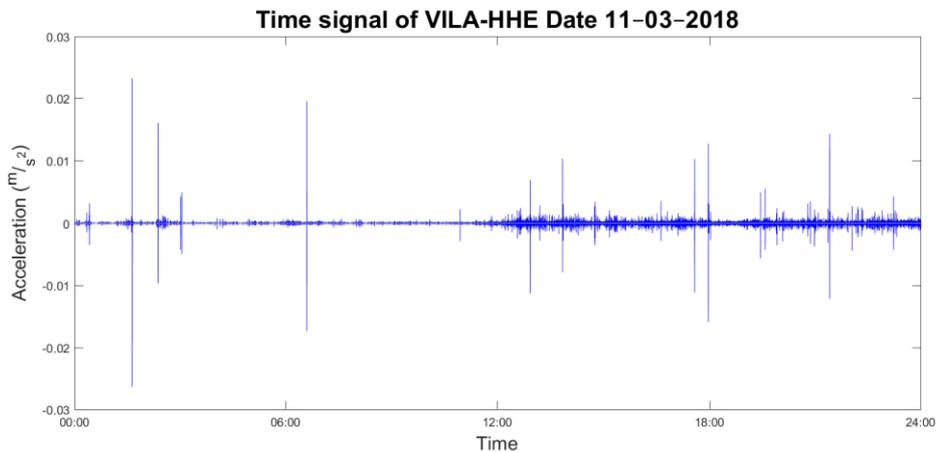


Figure A2.55. Seismic time signal for the HHE axis of VILA seismograph during the 11th of March 2018

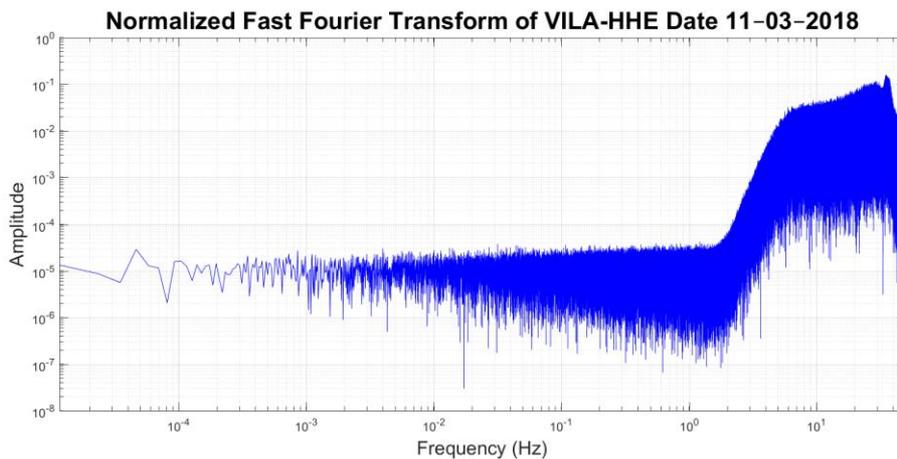


Figure A2.56. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of VILA seismograph during the 11th of March 2018

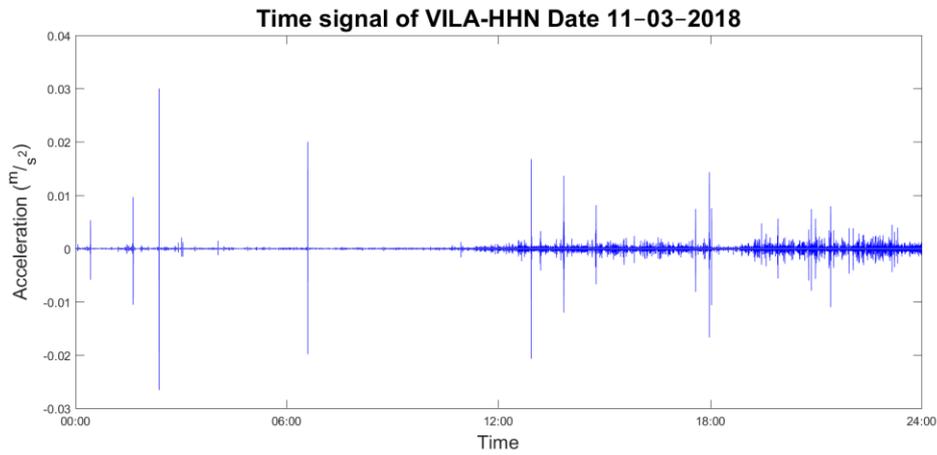


Figure A2.57. Seismic time signal for the HHN axis of VILA seismograph during the 11th of March 2018

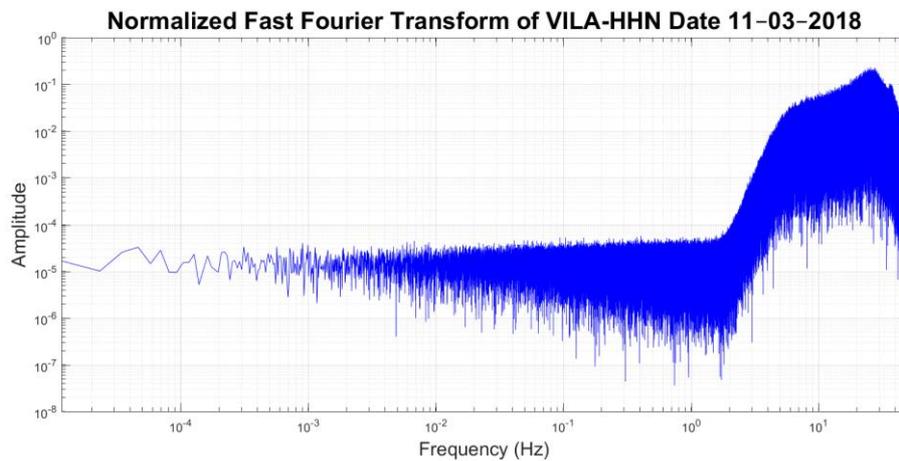


Figure A2.58. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of VILA seismograph during the 11th of March 2018

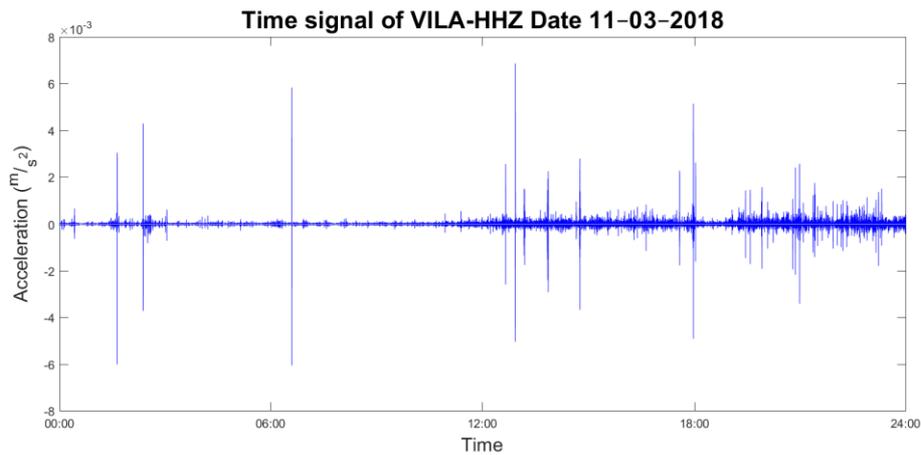


Figure A2.59. Seismic time signal for the HHZ axis of VILA seismograph during the 11th of March 2018

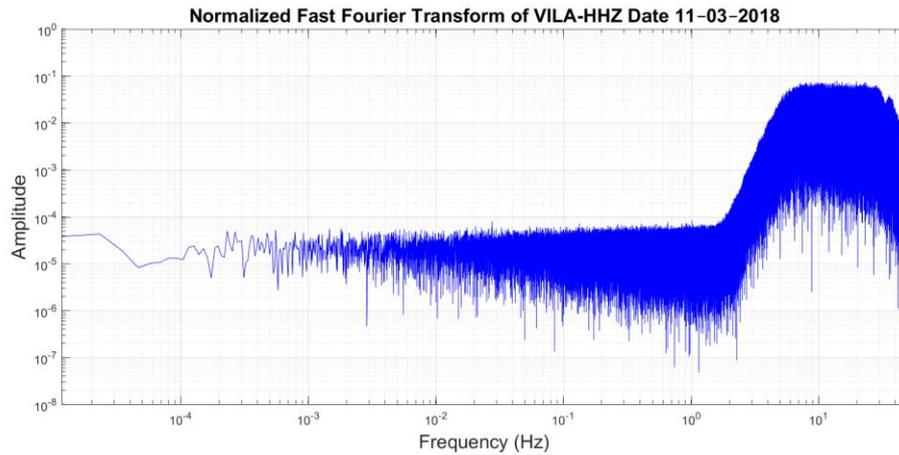


Figure A2.60. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of VILA seismograph during the 11th of March 2018

2.3. BAIN seismograph

3.2.1. 1st of February 2018

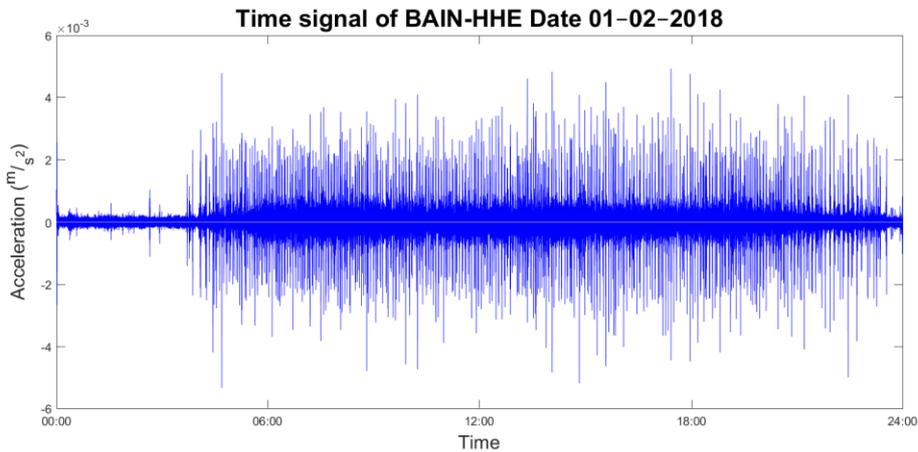


Figure A2.61. Seismic time signal for the HHE axis of BAIN seismograph during the 1st of February 2018

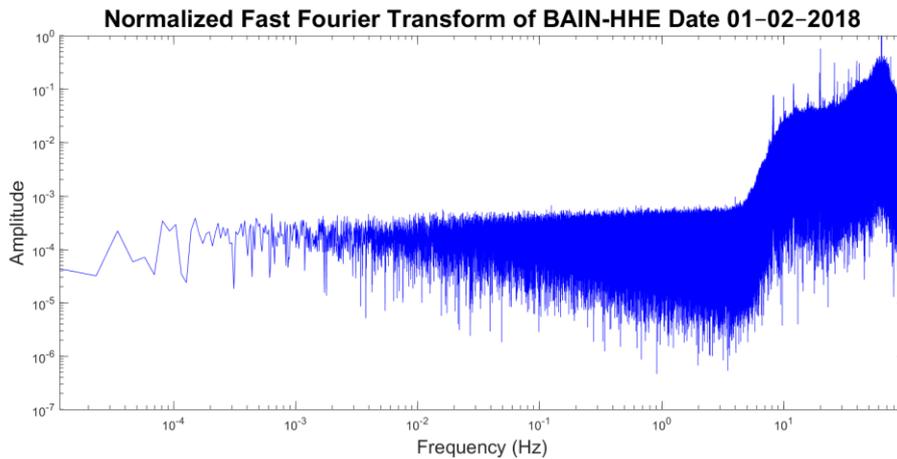


Figure A2.62. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 1st of February 2018

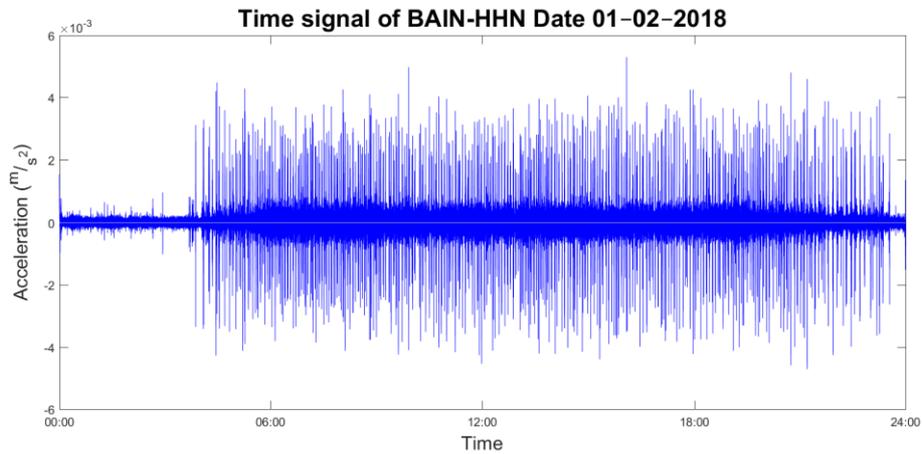


Figure A2.63. Seismic time signal for the HHN axis of BAIN seismograph during the 1st of February 2018

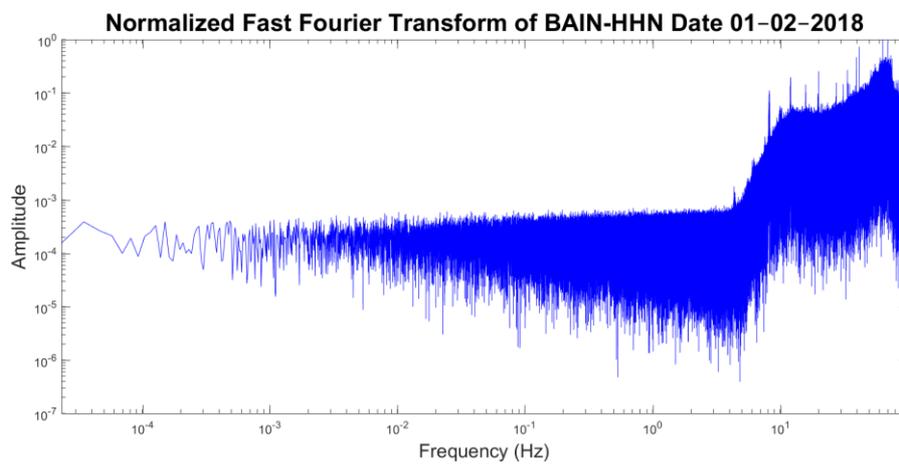


Figure A2.64. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 1st of February 2018

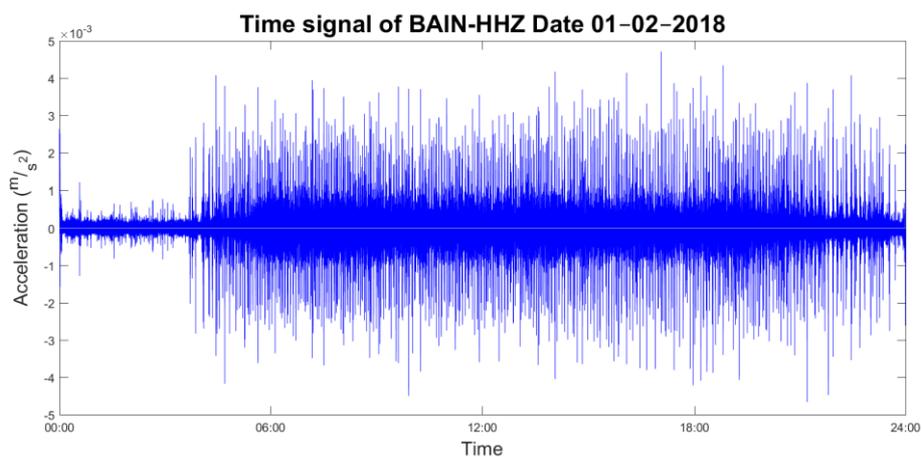


Figure A2.65. Seismic time signal for the HHZ axis of BAIN seismograph during the 1st of February 2018

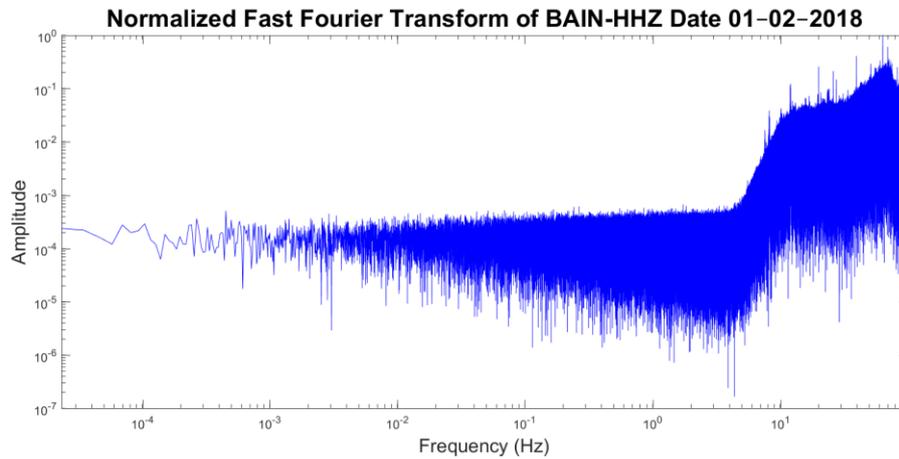


Figure A2.66. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 1st of February 2018

3.2.2. 11th of February 2018

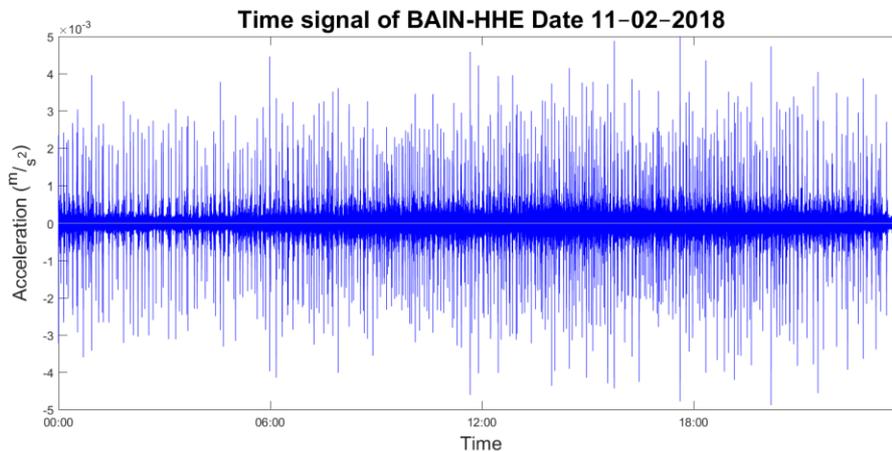


Figure A2.67. Seismic time signal for the HHE axis of BAIN seismograph during the 11th of February 2018

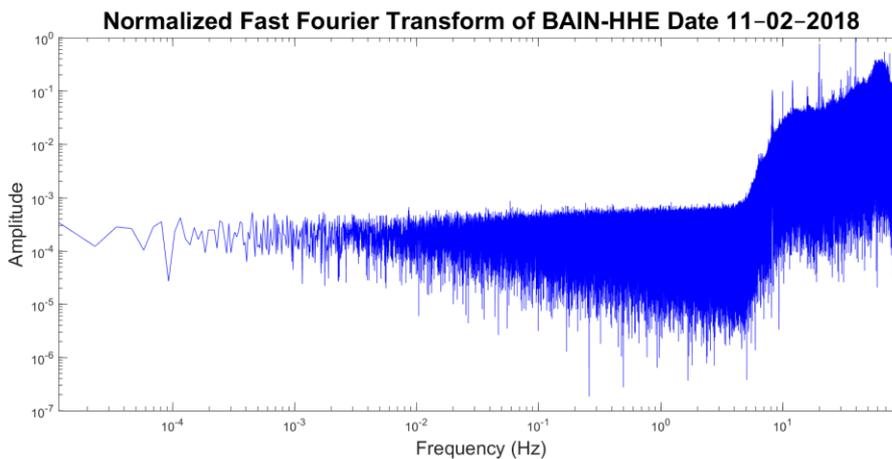


Figure A2.68. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 11th of February 2018

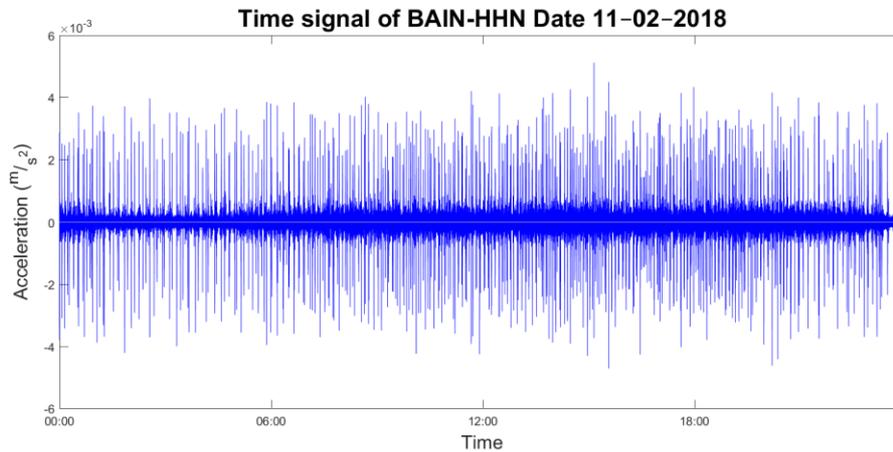


Figure A2.69. Seismic time signal for the HHN axis of BAIN seismograph during the 11th of February 2018

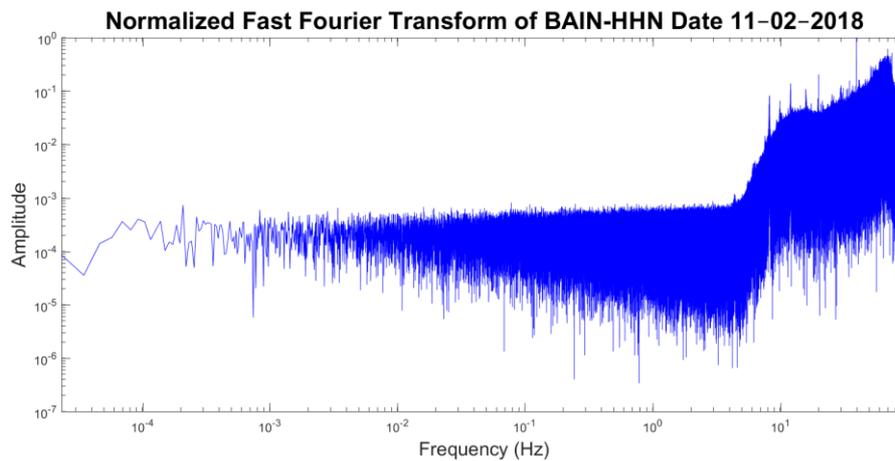


Figure A2.70. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 11th of February 2018

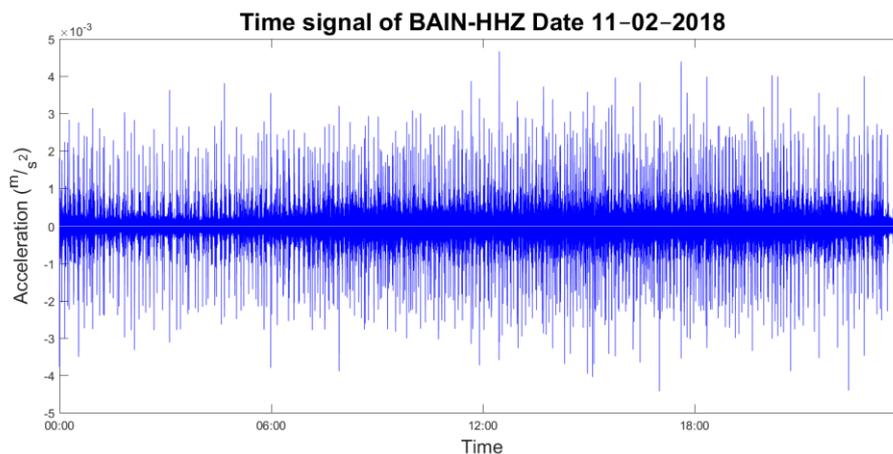


Figure A2.71. Seismic time signal for the HHZ axis of BAIN seismograph during the 11th of February 2018

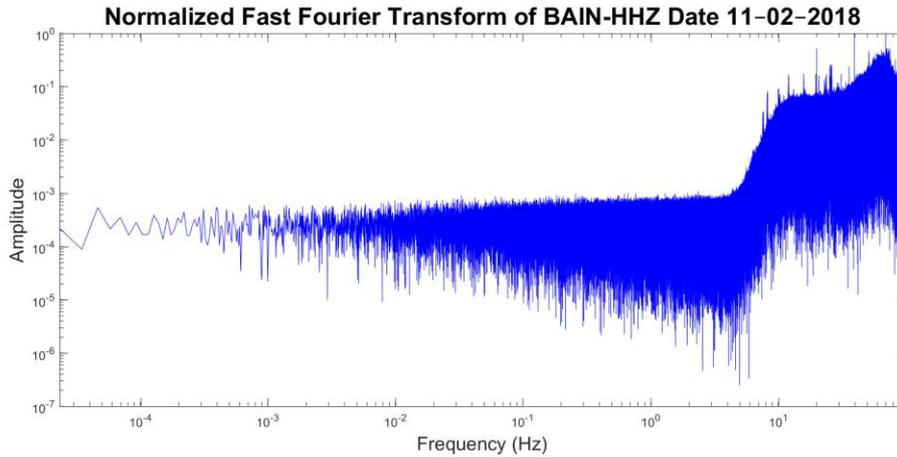


Figure A2.72. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 11th of February 2018

3.2.3. 21st of February 2018

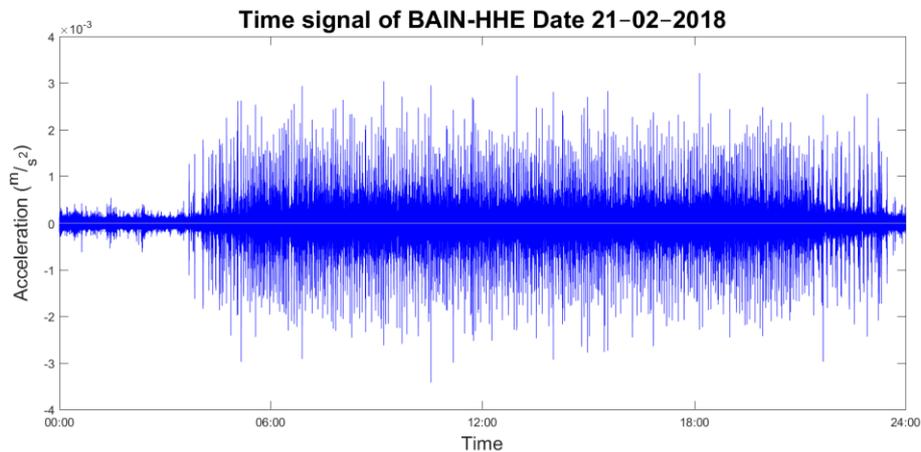


Figure A2.73. Seismic time signal for the HHE axis of BAIN seismograph during the 21st of February 2018

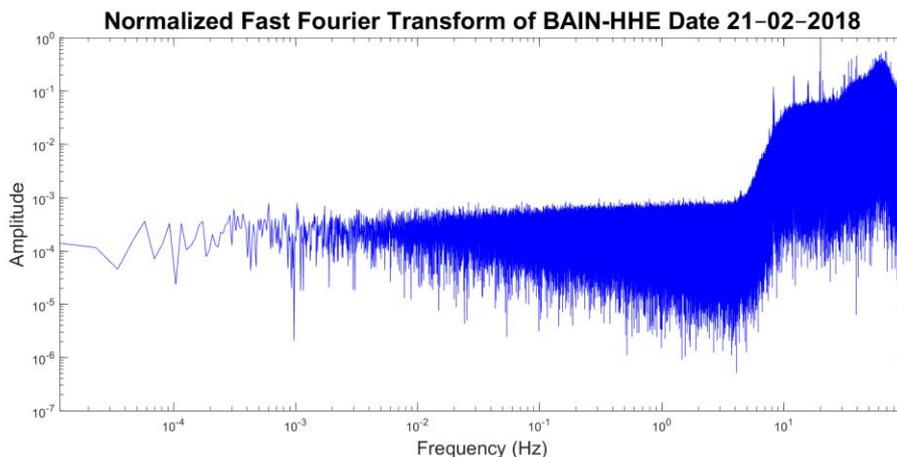


Figure A2.74. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 21st of February 2018

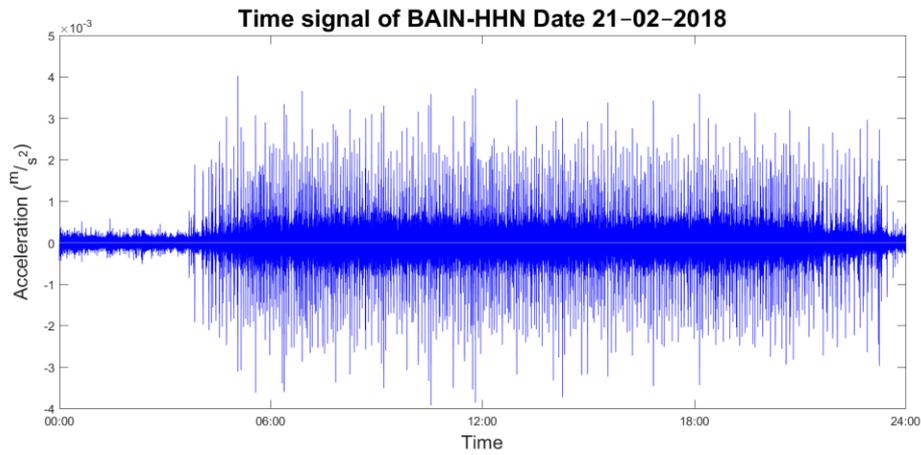


Figure A2.75. Seismic time signal for the HHN axis of BAIN seismograph during the 21st of February 2018

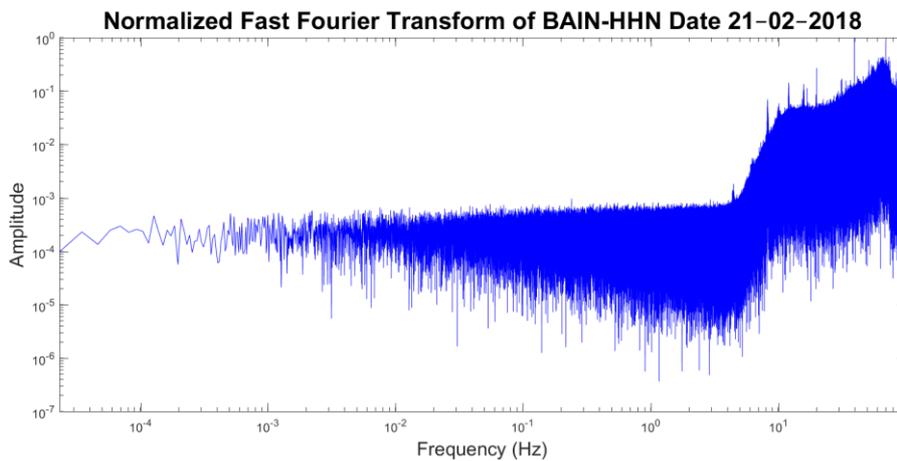


Figure A2.76. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 21st of February 2018

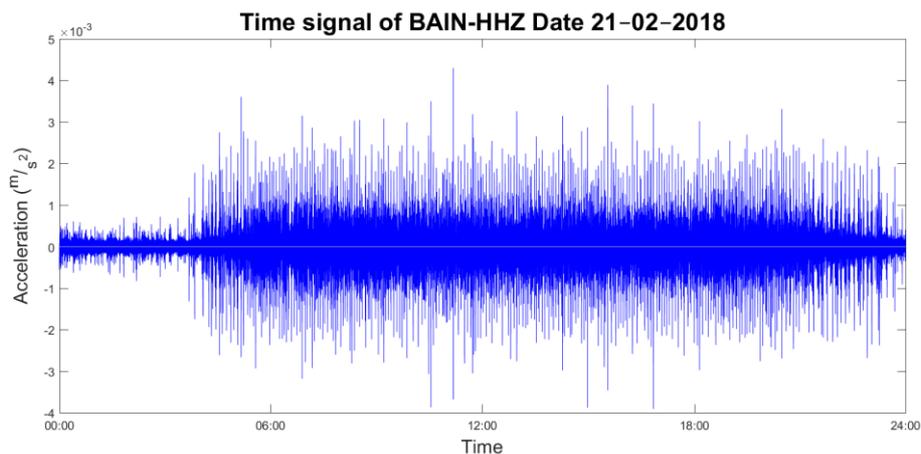


Figure A2.77. Seismic time signal for the HHZ axis of BAIN seismograph during the 21st of February 2018

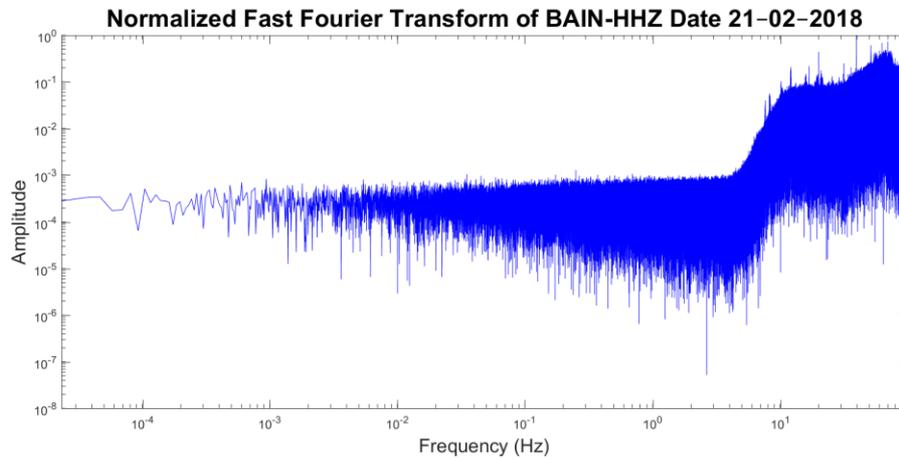


Figure A2.78. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 21st of February 2018

3.2.4. 1st of March 2018

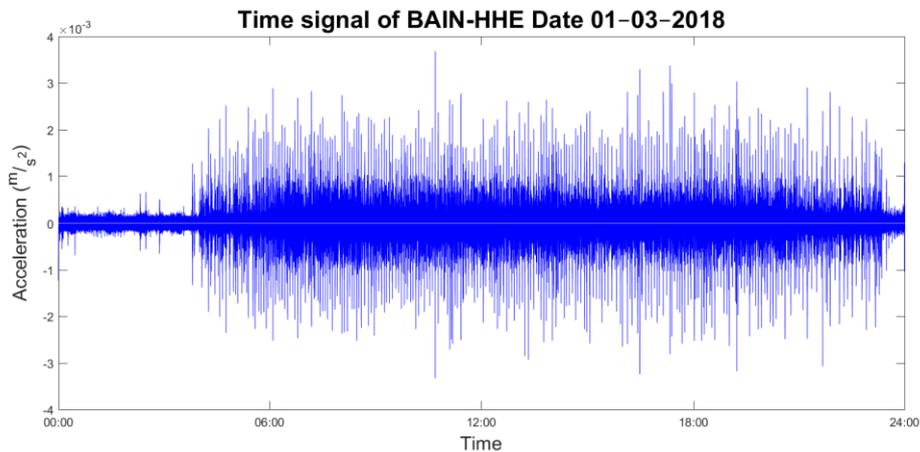


Figure A2.79. Seismic time signal for the HHE axis of BAIN seismograph during the 1st of March 2018

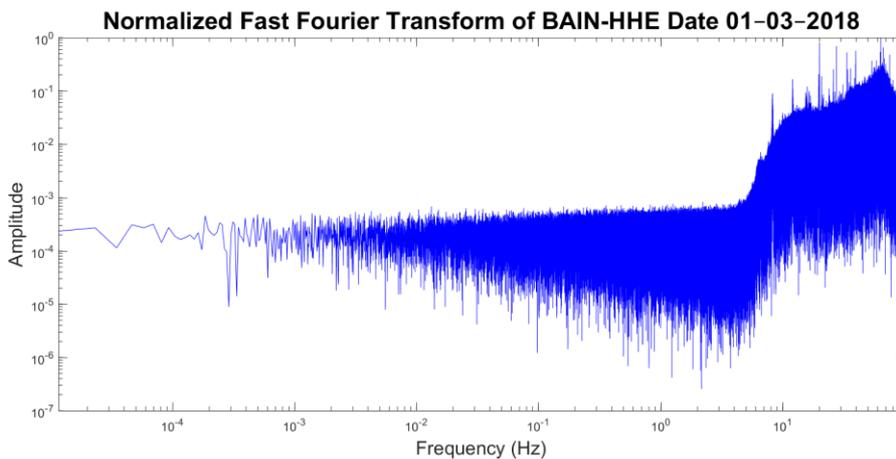


Figure A2.80. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 1st of March 2018

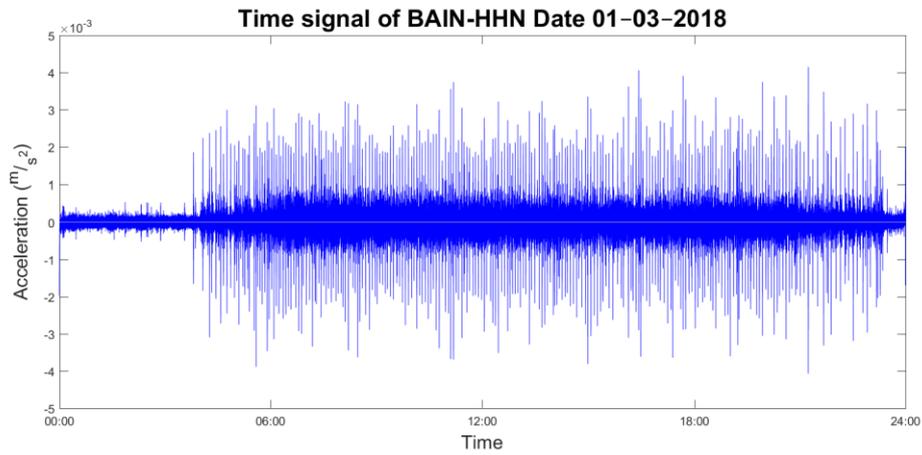


Figure A2.81. Seismic time signal for the HHN axis of BAIN seismograph during the 1st of March 2018

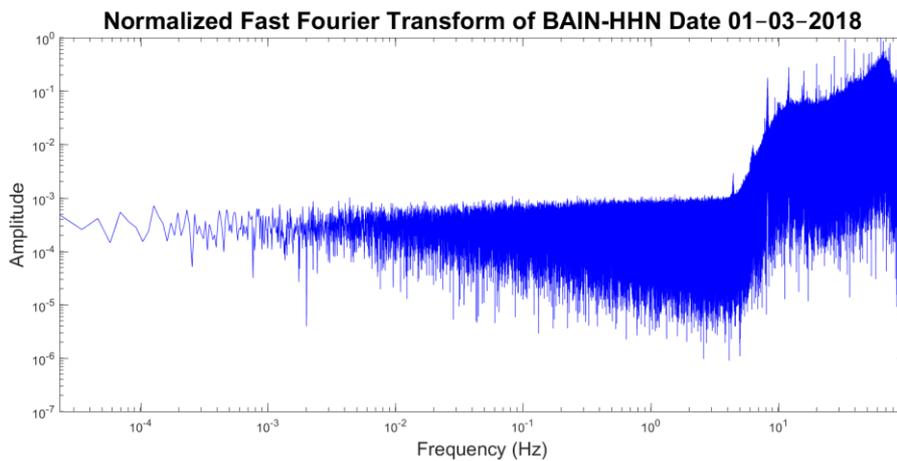


Figure A2.82. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 1st of March 2018

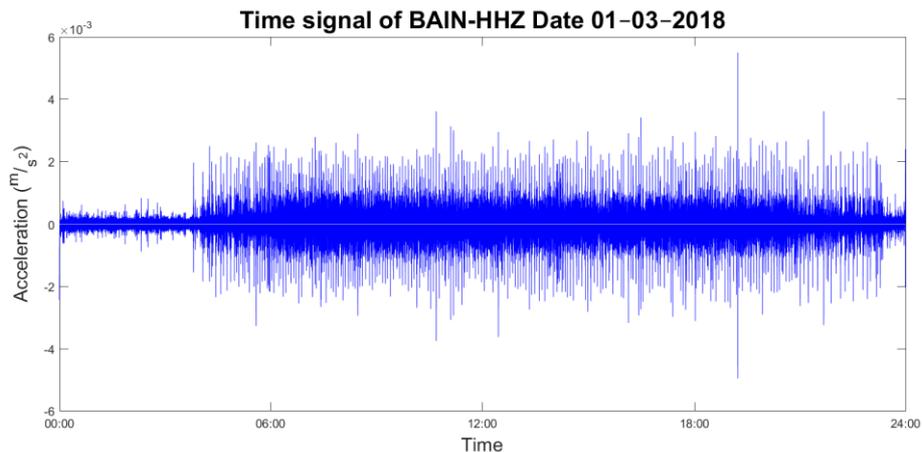


Figure A2.83. Seismic time signal for the HHZ axis of BAIN seismograph during the 1st of March 2018

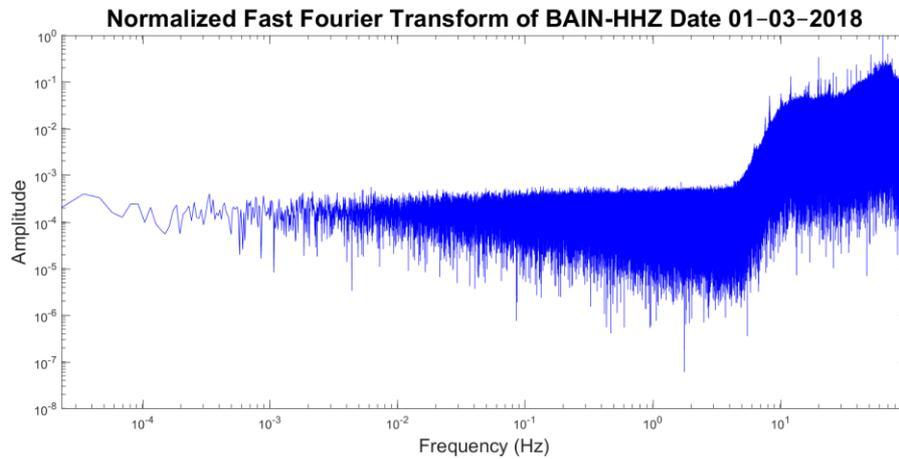


Figure A2.84. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 1st of March 2018

3.2.5. 13th of March 2018

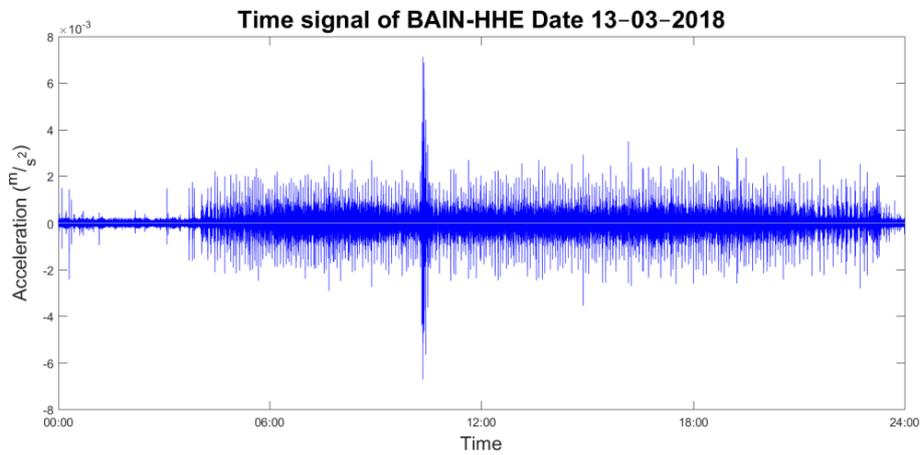


Figure A2.85. Seismic time signal for the HHE axis of BAIN seismograph during the 13th of March 2018

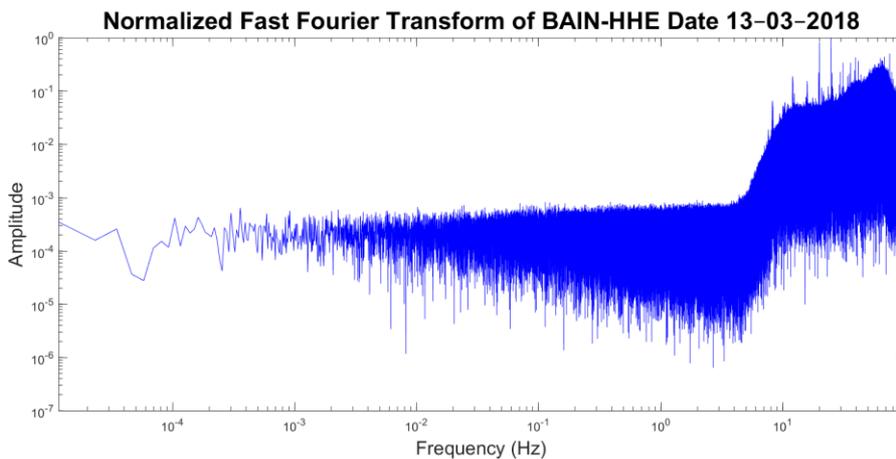


Figure A2.86. Normalized Fast Fourier Transform of the seismic signal for the HHE axis of BAIN seismograph during the 13th of March 2018

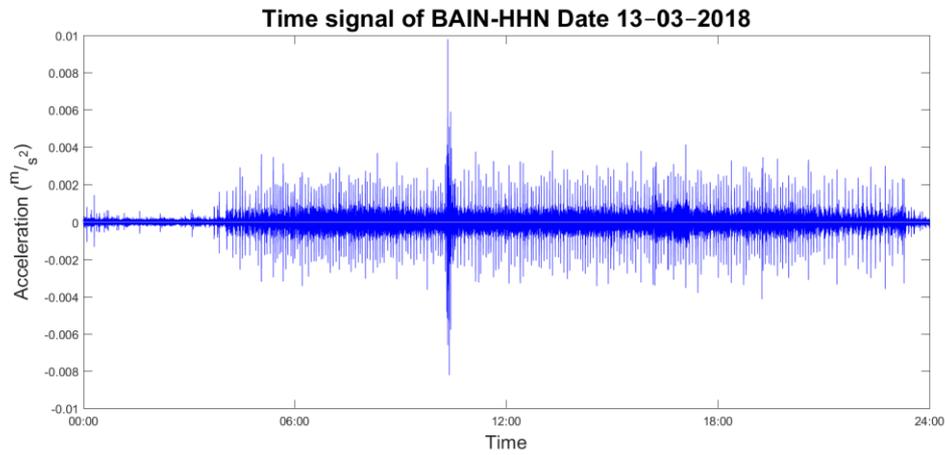


Figure A2.87. Seismic time signal for the HHN axis of BAIN seismograph during the 13th of March 2018

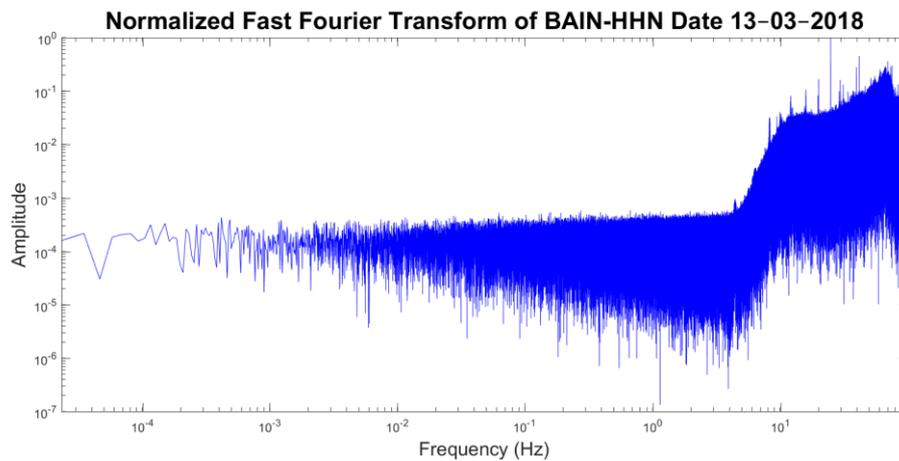


Figure A2.88. Normalized Fast Fourier Transform of the seismic signal for the HHN axis of BAIN seismograph during the 13th of March 2018

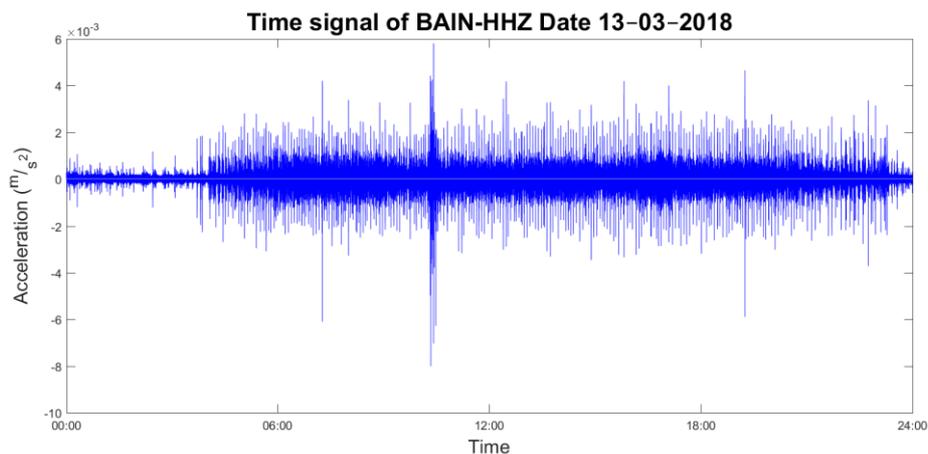


Figure A2.89. Seismic time signal for the HHZ axis of BAIN seismograph during the 13th of March 2018

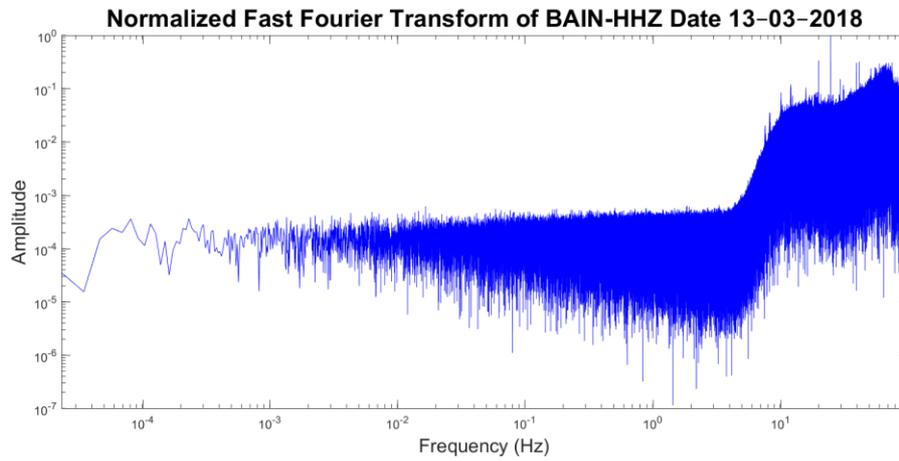


Figure A2.90. Normalized Fast Fourier Transform of the seismic signal for the HHZ axis of BAIN seismograph during the 13th of March 2018

Annex 3. ADXL355Z / Raspberry Pi 3B Connections

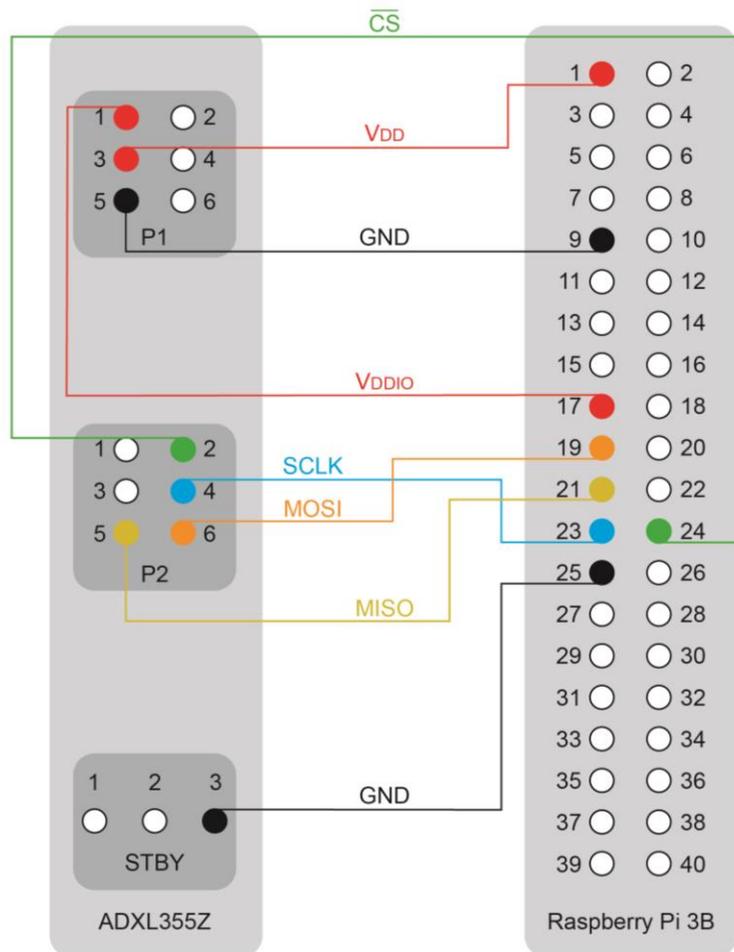


Figure A3.1. Connections between ADXL355Z and Raspberry Pi 3B