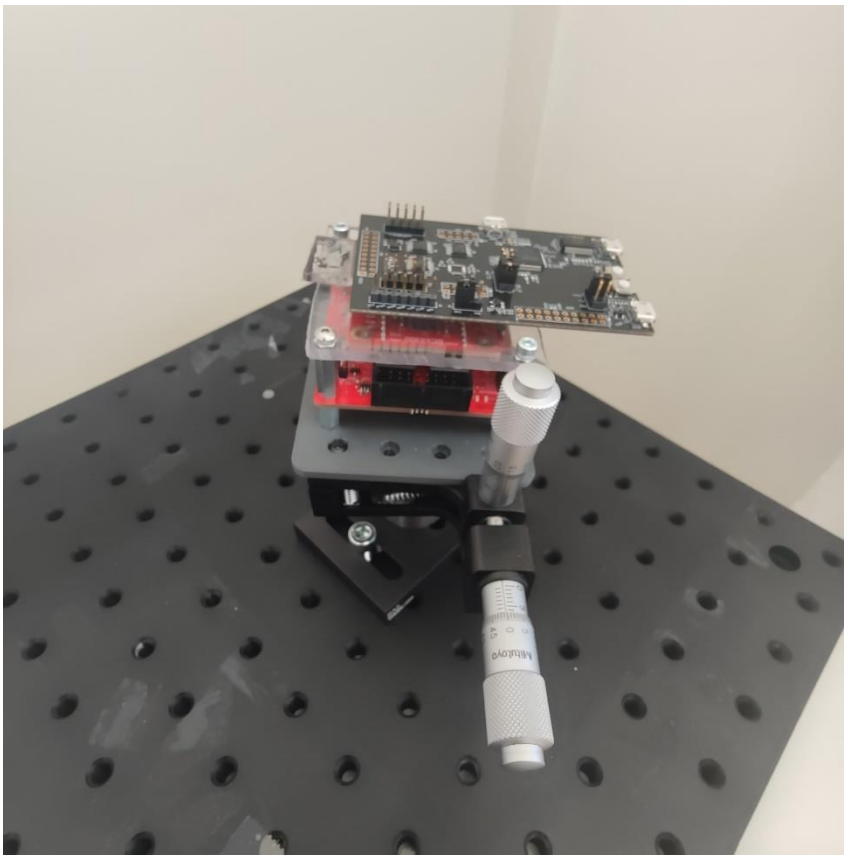


EVALUATION OF INERTIAL MEUSURMENT UNIT FOR ANTENNA ALIGNMENT

Evaluation of MEMS-based inertial measurement unit

Bachelor's Thesis in Electrical Engineering

Joel Brofors

DEPARTMENT OF Electrical Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Bachelor's Thesis 2021

EVALUATION OF INERTIAL MEUSURMENT UNIT FOR ANTENNA ALIGNMENT

Evaluation of MEMS-based inertial measurement unit

Joel Brofors

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

ABSTRACT

The increasing demand of higher throughput in telecommunications has led to the development of backhaul network links which are working in the E-band region (70-80Ghz). A Backhaul link is a point to point network with a narrow laser lock alike beam which is sensitive for mast movements which causes misalignments of the antenna. Ericsson has solved the problem with misalignments with different beam-steering techniques, but the antenna does not know in which direction to steer. Therefore, this study evaluates the performance of two MEMS based acceleration/inclination systems (Murat SCL3300 and TDK IIM42652) which can be used for tracking mast movements.

Findings from this project suggest that the SCL3300 is best suited for the tasks. For future work in the subject, the report points out the importance of following a well-established standard on how testing procedures and simulations should be conducted.

Keywords: IMU, MEMS, Accelerometer, Allan Variance, Allan Deviation, Telecommunication, Backhaul Network Links,

ACKNOWLEDGEMENTS

This bachelor's thesis was written during the spring of 2021 at Chalmers University of Technology under the department of Electronic Engineering

I would like to thank Ericsson and Jonas Hansryd for providing me with such an interesting research topic and for all of the support, advices, and guidance during the study. I would also like to thank my supervisor and examiner Bertil Thomas for supporting me in writing the thesis.

Joel Brofors

Gothenburg 2021

Table of Content

LIST OF ABBREVIATIONS

IMU	Inertial measurement Unit
GPS	Global Positioning System
DOF	Degrees of freedom
MEMS	Micro-electro-mechanical system
FOG	Fiber Optic Gyroscope
RLG	Ring Laser Gyroscope
CRG	Cylindrical Resonator Gyroscopes
INS	Inertial Navigation System
MCU	Microprocessor Control Unit
PC	Personal Computer
Bias instability	Amount of deviation from sensors mean value
VRW	Velocity Random Walk
ARW	Angular Random Walk
RRW	Rate Random Walk
ARW	Acceleration Random Walk
PSD	Power Spectral Density
AVAR	Allan Variance
ADEV	Allan Deviation
FIFO	First-In-First-Out
SNR	Signal to Noise Ratio

1 INTRODUCTION

The increasing demand of higher network speeds, and the development of 5G, have led to the development of wireless backhaul links. The most effective solution to increase the throughput of a network is to increase the transmitting frequency and bring down the signals to the mmWave frequency range. One example of frequency ranges in this region is the E-band (which has a range between 70-80Ghz). Today, wireless backhaul links have a capacity of 10Gbps and have the benefits of a fiber in terms of speed and are less costly. Furthermore, setting up wireless communication networks does not need extensive digging as fibers do. Other benefits of backhaul links with high frequency is that the components inside the link are smaller compared with lower frequency transmitters. Smaller components reduce material consumption and allow a shrinking of the formfactor. The drawback of higher frequency is that the transmitted information is concentrated on a very narrow signal (almost as a laser beam). The narrow signals make the links sensitive to misalignments, vibrations, and oscillations of the antenna and mast. Severe oscillations of telecommunication masts will result in a lower network capacity and can in some cases bring down a whole network. Ericsson has solved the problem regarding misalignment through a usage of beam-steering techniques. However, the antennas do not know in what directions to steer the transmitted beam and how it should steer it to find the optimal alignment.

1.1 DESCRIPTIONS

The focus in this project was on designing, building, and evaluating a compensation system that makes it possible for a beam-steering antenna to know where it should steer the signal to achieve maximum performance (i.e., throughput and efficiency). The compensation system consists of an inclinometer, actuator, with onboard microcontrollers.

The evaluation of the sensor was performed both through a simulated test environment, using MATLAB, and through tests on two Inertial Measurement Units. Later, the results from the tests were compared to evaluate how precise the simulated tests where. Gathering of test data was conducted with a tilt table that had 2 degrees of freedom (DOF). To determine the performance of the sensor, following metrics was evaluated:

- Precision of the measurement
- Drift due to noise, bias
- Lower detectable limit

1.2 DELIMITATION

To minimize the trouble shooting and the size of the project, following points were ignored:

- How the antennas perform beam-steering
- Laboratory environment based on Covid-19 restriction
- Control and communication with the antenna
- How the communication with the evaluation boards and IMU worked
- Assume the evaluation software is correct (no fault in the settings)

If the aim is a professional work, following topics need to be read and understood to be able to correctly execute the level of perfection that is needed:

- IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology—Random Instabilities [1]
- IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Laser Gyros [2]
- G. Langfelder, "Microelectromechanical systems integrating motion and displacement sensors," in *Smart Sensors and MEMS Edition*, Second Edition ed., Milan, Italy, Elsevier Ltd, 2018, pp. 395-428. [3]

With this said, this project can be seen as a general approach on evaluation of Inertial Measurement Units and may give the reader some sources to examinate if a professional work must be done on the subject.

1.3 CLARIFICATION OF THE ISSUE

Misalignment between point-to-point links will result in signal quality reduction and lower throughput. Hence, this project will focus on evaluation system for sensors to detect the movements that causes misalignment. In the future the sensor system can be combined with a beam steering antenna that cancels out the misalignment caused by the movements. To design, build, and evaluate the evolution system, this project aims to answer following questions:

Can simple trigonometry and normal equipment be used to determine the precision of the sensor

Is it possible to simulate an IMU and get almost same values as the sensor would generate in reality?

How to determine bias and drifts of the sensors?

Is it possible to measure the lowest limit only using the evolution board and cheap tools from a normal metal store?

Can a too sensitive sensor cause problem?

2 THEORY

In this section theory regarding Inertial Measurements Units (IMU), sensors, noise terms, stability analysis, and constraints simulations of IMUs will be presented. Today, almost all new electrical systems that can be found in smartphones, cars, drones, and airplanes have one or several sensors that are used for navigation, stability control and motion detections [3] [4].

2.1 INERTIAL MEASUREMENT UNIT

An electronical system that needs to know about its orientation or be able to measure an inertial motion are often equipped with one or many IMUs. An IMU contains sensors that can measure acceleration (relative the gravity), angular velocity (rotational force), and in some cases magnetic field (magnetometer). The magnetometer is used for determining its heading (horizontal reference) and it is commonly referred to the Magnetic North pole.

It is common for a IMU to have three to six degrees of freedom (DOF) [3] [5]. An IMU with three DOF are usually equipped with one axis gyroscope and two axis accelerometers, while a three axis IMU has three axes on both its gyroscope and accelerometer. It is also possible to get nine DOF, by using a six DOF IMU plus a three-axis magnetometer in the same packet. Some IMUs can also be equipped with a three-axis accelerometer and is in that case just an accelerometer.

In this section the accelerometer and gyroscope will be presented since the sensors that were evaluated did not have any magnetometer. First, the optical high-performance (and most expensive) gyroscopes Ring Laser Gyros (RLG) and Fiber Optic Gyros (FOG) will be presented [2] [3] [6]. After that, another solution will be presented that is based on the so called Micro-Electro-Mechanical Systems (MEMS) [3]. The MEMS chapter will present both gyroscopes and accelerometer sensors.

2.1.1 Optical Gyroscopes

As stated before, gyroscopes are normally used to measure angular velocities. Todays most used gyroscopes are Ring Laser Gyros (RLG), Fiber Optic Gyros (FOG), and gyroscopes based on Micro-Electro-Mechanical Systems (MEMS) [4]. The most accurate variants, the RLG and FOG, are commonly used in applications where highest possible accuracy and the smallest drift over time is needed. An example of usage areas for FOG and RLG are unmanned aircrafts, autopilots, and higher level of autonomous systems (see Figure 1). According to [6] a low-level FOG, or a RLG (in the price class of 30 000$) can have more than 20 times the accuracy compared with a 1000$ MEMS for the same application. In other words, it is a trade of between redundancy, accuracy, and economics when choosing an IMU. But the MEMS technology has been better in the recent years see citation below.

"New precision inertial navigation system (INS) markets are materializing and MEMS technology is also entering markets that were previously dominated by FOG technology. An apparent transition from FOG to MEMS technology is in antenna array stabilization applications"(p.1 [6]*)*

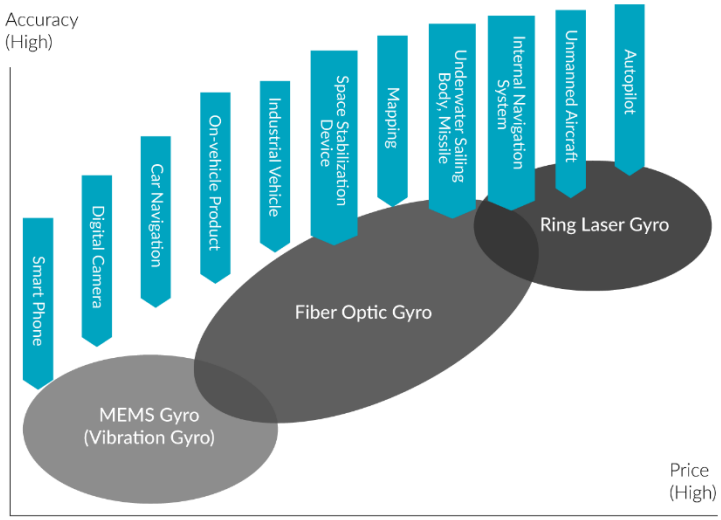


Figure 1: Depiction of accuracy verses price of three different gyroscopes and what market they are used for. Source of the picture [4].

Optical gyros are using the so-called Sagnac effect to detect angular velocity when a rotational force acting on its body. According to [7] [8] a FOG uses the interference of light to detect the angular velocity. One of the main parts of a FOG is a long coil made of thin fiberoptic fibers [8]. Into this fiber a single light source is fired, but before the light entering the fiber it is divided into two beams. These two beams will be led into the coil in the opposite direction to each other. The light beam who travels in the same direction as the rotation acting on the gyro will have a longer path. The other beam who travels against the rotation will instead experiences a shorter path to travel. This is known as the Sagnac effect and this phenomenon resulting in a phase shift between the two beams when they are exiting the coil [7] [6]. This phase shift can be measured, and it is proportional to the rotational speed acting on the gyroscope.
The RLG have a similar functionality as the FOG, the difference is that the RLG uses mirrors instead of a long fiber optic cable as the FOG dose.

2.2 Micro-Electronic-Mechanical Systems (MEMS)

The price of the optically based gyroscopes is too high to be used in consumer graded applications, even though the first FOG was demonstrated for more than 40 years ago [7] [6]. A cost-effective alternative to the optical IMUs (i.e., FOGs and RLGs) is the Micro-Electro-Mechanical Systems (MEMS). The definition for MEMS is following:

"The term MEMS stands for Microelectromechanical systems (MEMS) and is defined as miniaturized electromechanical units (i.e., structures) that are produced using micromachining technologies. This allows for a very compact format and cost-effectiveness." (p.1 [4])

MEMS sensors are partly made of silicon because it is well-established material in microelectronic industry, and the main reason manufacturers use silicon is the different crystallographic structures silicon can have, which affects the sensors electrical and mechanical

properties [3]. Hence, using a well-known material that is used in almost every electric product on the market has led to a good understanding how the MEMS should be produced and is one of the contributing factors of the rapidly increases of performance in today's MEMS. The MEMS market is not only for IMU applications, and it can also be used in telecommunication for oscillators and programable filters.

2.2.1 Gyroscopes Based on MEMS

Different types of MEMS gyroscopes that use micromachining technologies exists. Commonly used MEMS gyroscopes are: Cylindrical Resonator Gyroscopes (CRG), Piezoelectric Gyroscopes, Turning fork gyroscopes, Wine-glass resonator gyroscopes, and Vibrating wheel gyroscopes [4]. Despite the different mechanical structures that exists, they all are using the Coriolis Force to detect whether a sensor is affected by a rotational force. The Coriolis force states that an object with a certain mass *m* that travels with a velocity *v* in a rotating reference frame with the angular velocity ω, will experience a force acting on it. This force is perpendicular to the rotational axis and the objects motion in the rotational frame and this force is the Coriolis forces.

1

$$F_c = -2m(\omega \times v)$$

A depiction of how it works is presented in figure 2:

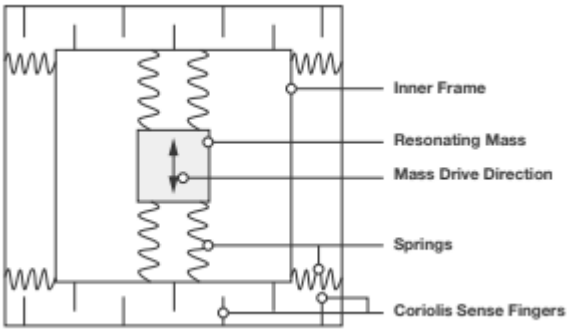


Figure 2: The picture illustrates a gyroscope which has one drive axis and one sense axis. Source: [9]

The gyroscope presented in Figure 2 is one of the simplest gyroscopes and consists of a given mass that hangs in a spring damped system. The working principles is as follow: The mass m oscillates along the drive axis (see Figure 3). When the sensor is rotating a perpendicular oscillation will be induced in the secondary axis, the so-called sense axis. The induced oscillation on the sense axis is caused by the Colonies effect and the oscillation is used to calculate the angular velocity *[8]*.

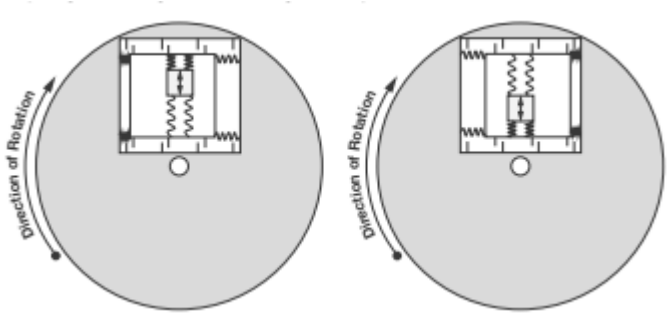


Figure 3: Working principle of the Gyroscope. Source: [9]

The main benefits of using MEMS based gyros over the optical alternative can be read in [6] [4] and some of the main advantages are:

- smaller footprint
- rugged construction
- cheaper
- faster startup time
- low weight
- high shock tolerant

2.2.2 Accelerometers Based on MEMS

As previously stated, an accelerometer is a sensor that is used to measure acceleration with respect to the gravitational forces. The traditional way to measure acceleration is to use an enforced spring system, that gives an output that is proportional to the acceleration given by Newtons second Law of motion and combining this knowledge with Hooke's law of tension a model of an accelerometer can be constructed.

Newtons law says that the acceleration a of an object with a given mass m is directly proportional to the net force F acting on it, and the force is inversely proportional to the mass of the object. In other words, if the force that is acting on the mass increases, the acceleration on the mass increases, because the acceleration is directly proportional to the net force acting on it (see equation 2).

2

$$F = m * a$$

Hooke's law of spring states; the fore F_c used to extend a spring is proportional to the extension distance x times a spring constant k (see equation 3).

3

$$F_c = k * x$$

Combining (Newton's) and (Hooke's law) gives the relationship (see equation 4) that shows how the acceleration is inversible proportional to the mass and proportional to the distance (k is fixed).

4

$$\alpha = \frac{k}{m} * x$$

Another type of MEMS accelerometer is the differential capacitance accelerometer. This type of accelerometer consists of a flexible spring system in which a given mass is attached. Parallel connected capacitors are placed around the mass and so-called electrode fingers are attached on the mass [10]. The purpose with the electrodes is to make it possible to change the distances between the capacitive plates on the parallel connected capacitors which is visualized in 4.

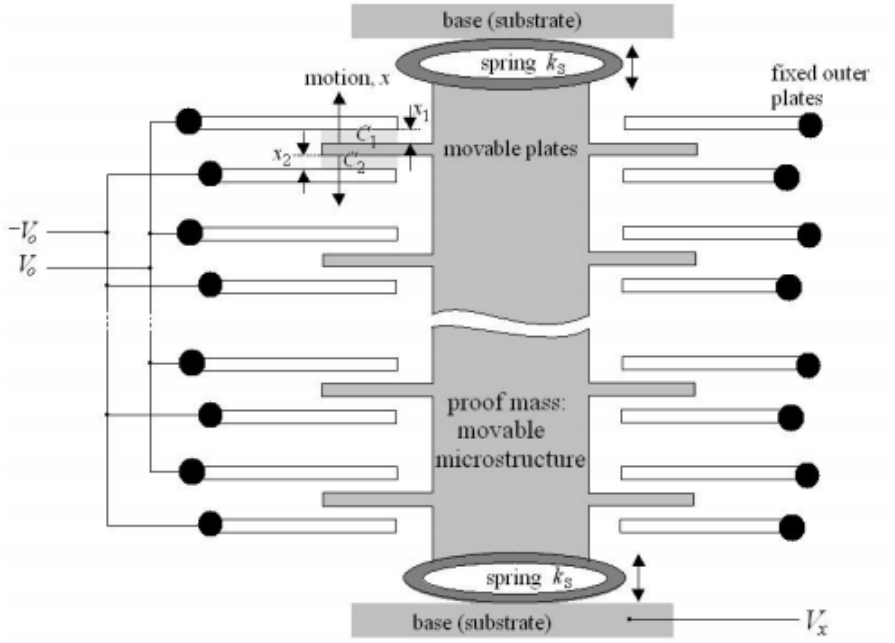


Figure 4:The above figure illustrates a differential capacitive MEMS accelerometer. Source:(p.5 [11])

When an acceleration is acting on the mass the springs will bend, and the electrodes will change the distances between the electrode plates, and this will change the capacitive differences in the circuit. This change will result in a change of voltage which can be measured and is proportional to the acceleration.

The capacitance C_0 is inversible proportional to the distance d between two parallel plates, and the capacitance depends on the plates area A, relative permittivity of free space ε_0 (between plates and electrode) and the relative permittivity of dielectric ε_r (in this case, it is the permittivity of the electrode) [10]. Halving the distance between the two plates will double the capacitance. Using the formula of capacitance between two capacitive parallel plates gives the following relationship (see equation 5).

5

$$C_0 = \frac{\varepsilon_0 \varepsilon_r A}{d} = \varepsilon_A \frac{1}{d}$$

Where $\varepsilon_A = \varepsilon_0 \varepsilon_r A$ is the permeability constant between a capacitive plate and the electrode. According to [12] and [10] the displacements x_1 and x_2 between the plates is proportional to the change of capacitance and can be formulated as following:

6

$$C_1 = \frac{\varepsilon_A}{x_1} = \frac{\varepsilon_A}{d + x} = C_0 - \Delta C$$

7

$$C_2 = \frac{\varepsilon_A}{x_2} = \frac{\varepsilon_A}{d - x} = C_0 + \Delta C$$

Where the capacitance difference between the plates becomes:

8

$$C_2 - C_1 = 2\Delta C = 2\frac{\varepsilon_A x}{d^2 - x^2}$$

Rewrite the nonlinear function equation 8 gives:

9

$$\Delta C x^2 + \varepsilon_A x - \Delta C d^2 = 0$$

And solve for displacement x with the assumption that a small displacement $\Delta C x^2$ is negligible, which leads to the following approximation:

10

$$x \approx \Delta C \frac{d^2}{\varepsilon_A} = d\frac{\Delta C}{C_0}$$

Equation 10 states that the displacement is approximately proportional to the change in capacitance between the plates (i.e.., ΔC).

A circuit of a differential capacitive accelerometer be simplified as following (see Figure 5) [10]. In Figure 5, the two adjustable capacitive plates and the capacitors are connected to a square wave voltage source and a phase shifter.

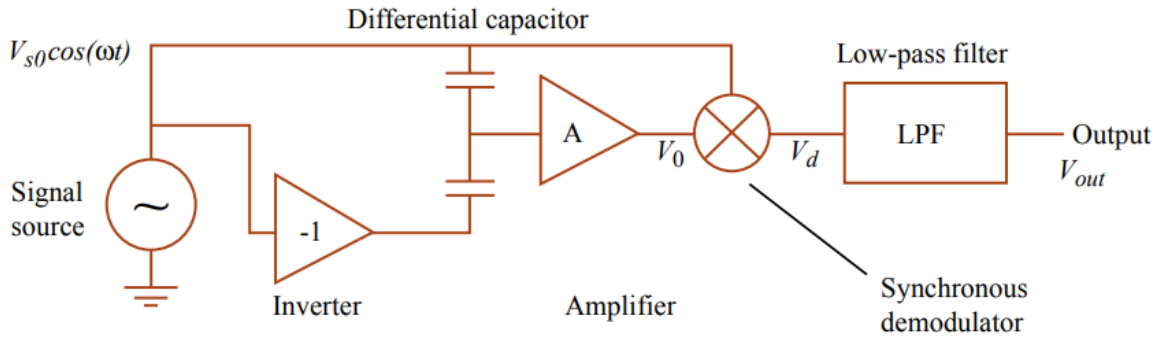


Figure 5: The above figure illustrates a differential capacitive accelerometer. Source: [13]

The voltage over C_2 is 180 degrees phase shifted in relation to the voltage over C_1. Hence, only two working conditions are possible in the circuit [10] [11]. The first gives a positive voltage V+ at point A, while the voltage at point B are 0 volt. The second state gives 0 volt at A and 5

volts at B. Therefore, a change of distance between the plates will result in a voltage difference which can be measured and can be written as a voltage divider between the capacitor plates:

11

$$(V_x + V_0)C_1 + (V_x - V_0)C_2 = 0$$

Using equation 10,7,and 6 gives the output voltage [12]:

12

$$V_x = V_0 \frac{C_2 - C_1}{C_2 + C_1} = \frac{x}{d} V_0$$

Now using the equations 2,3,4,12 set ma = kx and solve for the acceleration a give the following:

13

$$a = \frac{kdV_x}{mV_0}$$

The above equation, 13, gives the simplest explanation on of how a capacitive accelerometer work. For example, the Murata SCL3300 used in this thesis has a high precision three axis capacitive micromachined accelerometer [14]. One of the advantages of using a capacitive based MEMS is its ability to self-compensate for temperature drifts caused by self-heating and fluctuation in the ambient temperature. More information about errors and drifts in MEMS is presented in later**.**

2.2.3 Smart MEMS

The MEMS that were evaluated in this thesis, are categorized as smart sensors [3]. A smart senor is a sensor that has an extra set of features over the necessary functions required for represent the physical measurement of interests [3].The author also points out different sorts of features that are needed to be categorized as a smart sensor. The sensors should be able to be reconfigured between different settings and have some level of on-board signal processing capabilities. It should therefore have different detection ranges, resolution ranges, and have basic signal processing capabilities such as low pass filtering. One important feature is self-diagnostic; it lower trouble shooting time and can be used to estimate the lifetime the sensor will have. For example, if the sensor is undergoing hard accelerations and extreme temperatures its lifetime will be decreased. The springs inside a MEMS will be stressed if suddenly high accelerations are acting on it. It can be seen in the sensor's data sheet what states the maximum stress the device can handle. The springs within the MEMS sensors can be compared to the dampers spring system in a car. Running over a speed bump in 120 km/h may not break the dampers the first time but doing it many times will shorting the life cycles of the damping systemin.

The SCL3300 and the IIM42652 are the two sensors that were evaluated in this project. Booth sensors are characterized as smart, making them more versatile and adjustable for a wider market as well as higher performance over a wider operation area. The described features are just a few examples of features that are characteristic for a smart MEMS. Extra information

about MEMS and sensor types can be found in “Microelectromechanical systems integrating motion and displacement sensors” [3].

2.2.4 MEMS Error Characteristic

In this section, three different types of noise will be presented where all resulting into uncertainties in the IMU measurement readings. According to [2] [15] [16] instability of most frequency sources can be modeled. This can be done due to that the power-law noise processes depending on its spectral density’s functional dependency on Fourier frequency [16].

“It has been showed that a Powe law noise processes are characterized for their functional dependence on Fourier frequency. Modeling of noise can therefore be done with the knowledge of that power-law noises have a spectral density of their fractional frequency fluctuations of the form Sy(f) $\propto f^{\alpha}$, where f is the Fourier or sideband frequency in hertz, and α is the power law exponent.” [p.5 [15]]

More information on the subject and how to model stochastic processes, noise of different kinds, and other types of analytical tool are named and referenced in the ”IEEE Standard Specification Format Guide and Testing Procedure for Sigle axis Laser Gyros” [2] and also in [1] [16] [15].

Later in this report, the relationship between two methods commonly used for noise analysis will be presented: the power spectral density (PSD) and the Allan variance. It has been showed that noise can be characterized because the noise will be treated differently by the PSD and Allan variance, based on what sorts of noise it is. The data will produce slopes that are typical for just a specific sort of noise when it is plotted on a log-log plot [16]. According to [2] PSD is it well suited for analyzing and characterizing data and stochastic modeling and is commonly used for representation of spectral decomposition of time series data. However, this method will not be presented in this project.

2.2.5 White Noise and Random Walk

White noise, also known as thermal noise, is caused by randomly motion of charge carriers and it is one of the most common noise types in today’s electronics [17]. White noise is generated in all passive components which has a loss, and the noise fluctuates at a rate that is much faster than the sensors sampling time and it is proportional to the temperature of the device [17] . The name white noise comes from its analogy with white light, due to the white light including all frequencies [17]. It can be seen as a flat noise floor on a PSD plot because the signal will have same intensity distributed over different frequencies. Both a gyroscope and an accelerometer are affected by this white noise, and it will be propagated to the devise’s output value and will be seen as uncorrelated random variables in the output data. The phenomenon is called angular random walk (ARW) for a gyroscope, given in $°/\sqrt{h}$, or as $°/\sqrt{s}$. While for an accelerometer it is called velocity random walk (VRW) and is given in $g/\sqrt{h}$, or as $\frac{m}{s^2}\frac{1}{\sqrt{Hz}}$.

2.2.6 Bias Instability (Flicker Noises)

Flicker noise is a common noise type in semiconductive material and can both be found in conductors and resistive material. The noise is a result from a small random variation in the resistive material inside the semiconductors. The noise is directly proportional to the current and temperature in the semiconductor, while its power is inversely proportional to the frequency and is therefore sometimes referred as *1/f* noise [15]. The noise is not pure white, as it does not have any flat power spectrum over all frequencies which thermal noise has, instead it has bigger

impact on lower frequencies. Thanks for its low frequency characteristics the 1/f noise will be seen as a bias fluctuation in the recorded data [2].

The bias instability is a measurement on how the bias will drift over time at a constant temperature and sample rate. Finding the bias instability parameter is the same as finding the spot where the optimal accuracy is given. This can be performed when averaging many data points over a finite time and will later be seen in coming sections. The bias for an accelerometer is written as $\frac{m}{s^2}$ or as g.

2.2.7 Random Walk

It might seem attractive to averaging data over a long-time span to get a good average value, but doing so will instead lead to problems with another noise phenomena, the so-called acceleration random walk (ARW) for accelerometer, and rate random walk (RRW), for an gyroscope [2]. The nose has no certain origin and can be seen when averaging of data over a longer time period. After a while will the noise be seen as a drift of the output data. This sort of noise exists in both mechanical and optical IMUs [2]. The RRW for an accelerometer can be written as $\frac{m}{s^3}\frac{1}{\sqrt{Hz}}$.

2.2.8 Other Noise Types

Not all sorts of noises that exists and affects IMU will be presented in this thesis. It exists more complex methods to find them as well as have more degree of freedom on the measurement data. The following sources give good information about how the different sources of noise can be distinguished, modeled, and what types of tools that exists to processing them [1] [2] [15] [16].

In the next section, one of the less restricted methods for analysis of stochastic models, the so-called Allan variance, will be presented.

2.3 ALLAN VARIANCE

The original Allan variance (AVAR) was developed in 1966, as a result of D. W Allan's master thesis studies [15]. AVAR was originally developed as a statistic method for analyzing frequency stability on oscillators, clocks, and amplifiers. Since then, newer types of the AVAR methods have emerged, which have better performance and wider field of use. Both the AVAR and PSD are of the less restricted methods today used for investigation and analysis of stochastic models. Hence, they are both used and preferred means for analysis in the inertial measurement community [2] [15]. The AVAR can be used for determining the stability of IMUs and for characterizing and estimating random stabilities coefficients (the coefficients will be presented in a later section). The method is used in data analysis, where an estimation of uncertainty and characterizing of an underlying noise processes is needed [2] [18].

> *"It can be used to determine the character of the underlying random processes that give rise to the data noise. As such, it helps identify the source of a given noise term present in the data; whether it is inherently in the instrument or in the absence of any plausible mechanism within the instrument, its origin should be sought in the test setup" [p.68 [2]]*

The citation above states that AVAR can be used to analyze noise of any instrument of interest. With this said, trying to analyze a high precision instrument with an instrument with higher

level of intrinsic noise, will fail drastically. The analysis-instrument will induce uncertainties into the data (i.e., unwanted noise), and as a result, the AVAR will show as the precision clock is noisy, when the noise in fact comes from the testing instrument and not the clock.

2.3.1 Working Principles and Definition of Allan Variance

The AVAR is computed by taking N data samples, with FS samples per second, resulting in a sample time of τ_0. The data clusters will then be formed with a duration of τ_0, $2\tau_0$, $m\tau_0$,($m < (N - 1) / 2$) and obtain the average of the data points in each cluster over the length of the cluster itself [2] [19] [20]. Where m is the averaging factor for each cluster.

"The Allan variance is defined as the two-sample variance of the data cluster averages as a function of cluster time." [p.68 [2]*]*

In this thesis an alternative variant of AVAR was used, the so-called overlapped Allan variance. The overlapped AVAR does as name suggests, overlapping data clusters. The overlapped variant has superior performance over the non-overlapped variant, and it is especially good when it comes to bigger data sets [2].

The output data from a gyro, or accelerometer can therefore be written as

14

$$\theta(t) = \int^{t} \Omega(t')dt'$$

The data samples are taken with the times $t = k\tau_0$ *(τ_0, $2\tau_0$, $3\tau_0$...)* and *k* varies from *1* to *N* samples [20]. Another possible way is to calculate the cumulative sum of the discrete data samples Ω times the sample period τ_0. After this is done the Allan variance can be calculated as follow:

15

$$\sigma^2(\tau) = \frac{1}{2\tau^2} < (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2 >$$

Where the < > is the ensemble average and is a function of $\tau = m\tau_0$. When expanding the ensemble average the result will be following [2]:

16

$$\sigma^2(\tau) = \frac{1}{2\tau^2(N-2m)} \sum_{k=1}^{N-2m} (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2$$

The above equation (Equation 16) is the equation used for calculating the overlapped Allan variance which is used in this thesis, and it is done for a particular value of τ using the IMUs data samples. This can only be done if we know the sample period and greatness measured. Example of how the overlapped Allan variance operates over a given data samples is depicted in Figure 6.

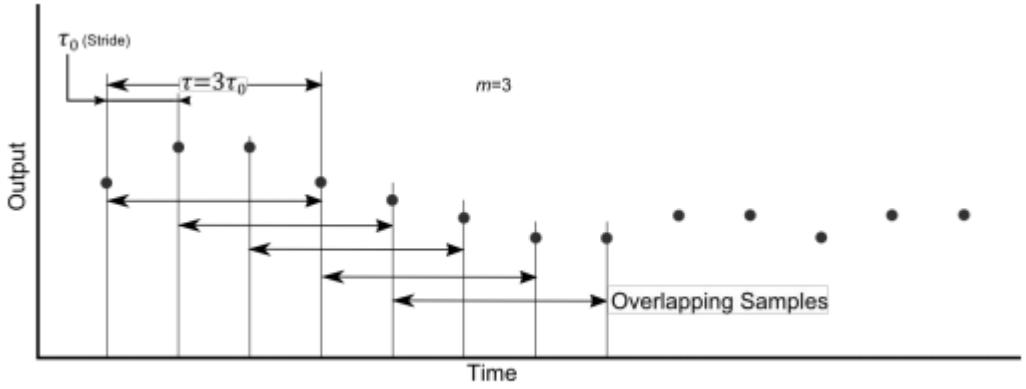


Figure 1. Overlapping Samples

Figure 6: The above picture illustrates the overlapped Allan variance, with an averaging factor m = 3 and the time duration of each cluster is averaged over is therefore $\tau = 3\tau_0$. Source [20]

The relationship between AVAR and the two side PSD $S_\Omega(f)$ is then given:

17

$$\sigma^2(\tau) = 4\int_0^\infty S_\Omega(f)\frac{sin^4(\pi f\tau)}{(\pi f\tau)^2}df$$

The above equation (Equation 17) is used when characterizing the different noise terms PSD when calculation of the AVAR has been carried out [2]. This function is used to operate on the different clusters describe above, and the AVAR is proportional to the total noise power when passed through a filter of the above characteristics (see equation 17). The filter is depending on τ and therefore it is possible to examine the stochastic processes just by changing the τ [2]. Hence, AVAR is used for quantifying different types of noises terms that could exist in each data set of interests. The most common way to extinguish between noises are to use the square root of AVAR which gives the *rms* value and is called Allan deviation (ADEV). An illustration of the noise terms are commonly depicted a on log-sigma log-tau plot and will produce slopes that is characteristic for the nose terms (see Figure 7).

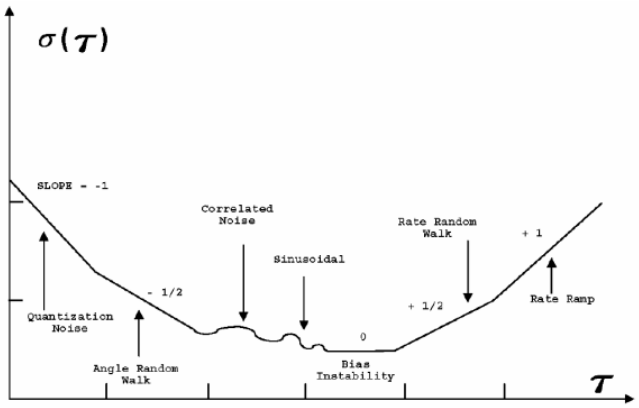


Figure 7:The Allan deviation of data which includes different power law noise terms. Source [15]

2.4 NOISE PARAMETRIC IDENTIFICATION

This section will present how noise identification of bias instability, rate/acceleration random walk, and random walk are performed by using (equation 17,18) and the respective PSD association to respective noise term. In the following source [2] answers are given on how the PSD associates with different power law noises and their characteristic slopes.

2.4.1 Angular/Velocity Random walk

The ARW / VRW is characterized by the PSD of white noise and can be written as equation 18, where *N* is the random walk coefficient [2]:

18

$$S_{\Omega}(f) = N^2$$

The equation 18 is then substituted into the formula 17 which is used for identification, integrating the formula gives the AVAR of the data:

19

$$\sigma^2(\tau) = \frac{N^2}{\tau}$$

Plotting the ADEV σ(τ) versus τ on a log-sigma log-tau plot gives the characteristic slope of -1/2 for Random Walk. The numerical value of the coefficient *N* can be read directly from the line at τ = 1 [2].

2.4.2 Bias Instability

The noise resulting in bias fluctuation in the IMU is caused by random flickering from the semiconducting material properties in the device. The noise is of a low-frequency characteristic and is depending on the frequency of the device under tests. The PSD following association to the flicker noise, see below in equation 20 [2]:

20

$$S_{\Omega}(f) = \begin{cases} \left(\frac{B^2}{2\pi}\right)\frac{1}{f}, & f \leq f_0 \\ 0, & f > f_0 \end{cases}$$

Where the constant *B* is the bias instability coefficient and the f_0 is the cut off frequency. The cut off frequency will, in this case when measuring an IMU, be dependent on following:

- Internal filters
- ADC and sampling bandwidth of the MEMS IMU itself
- Drop of frames between the IMU and host systems or overflowing on hardware registers, which in the long run will look like noise in the data
- External noise such causing slow oscillations, one example is AC-power grid sounds from power supplies.

The equation 20 is then substituted into equation 17, then by integrate the formula gives the $\sigma(\tau)$, see following equation 21:

21

$$\sigma^2(\tau) = \frac{2B^2}{\pi}\left[\ln(2) + -\frac{sin^3(x)}{2x^2}\big(\sin(x) + (4x)cos(x)\big) + Ci(2x) - Ci(4x)\right]$$

In this function, $x = \pi f_0 \tau$, while *Ci* is the cosine-integral function.
If τ is bigger than the invers of the cutoff frequency, the following simplification can be made:

22

$$\sigma^2(\tau) = \frac{2B^2}{\pi}\ln(2)$$

The above equation 22, will give us a line on a log-sigma log-tau plot with a slope of 0 for a $\sigma(\tau)$ [2] [19] .

2.4.3 Rate / Acceleration Random Walk

As stated, before in section 2.2.7 the problem caused by rate random walk can be seen when averaging data over a longer period of time. Suddenly, the power noise level will start to increase. This occurs due to a randomly process of uncertain origin and it is stated to have some few cases of exponentially correlation noise that happens over a long correlation time [2].

The PSD has following association for the noise causing RRW [2], see in equation 23:

23

$$S_\Omega(f) = \left(\frac{K}{2\pi}\right)^2 \frac{1}{f^2}$$

Where the coefficient *K* is the rate random walk. Substitution of PSD equation 23 into over filter function and integrate, the result gives us the following equation:

24

$$\sigma^2(\tau) = \frac{K^2\tau}{3}$$

And the square root of the AVAR gives:

25

$$\sigma(\tau) = K\sqrt{\frac{\tau}{3}}$$

The slope produced by random walk is ½ on a log-sigma log-tau plot. The value for *K* can be directly read from this line at $\tau = 3$ [2].

3 HARDWARE AND SOFTWARE

In this section, the hardware and software used in this thesis will be presented. First, the two IMUs (SCL3300 and IIM42652) with their key features. After the IMS, a Pitch and Yaw table for fiber optics and last two different evaluation software will be presented and discussed.

3.1 HARDWARE TO EVALUATE

Today's smart MEMS sensors can come equipped with features as adjustable filters, internal signal processing units, self-calibration, self-diagnostic, digital interface, and in some cases even FIFO buffers, to name a few. This chapter will therefore be focusing on describing the hardware that was evaluated in this thesis. First the two MEMS IMUs will be presented followed by a presentation of the fiber optic precision pan tilt table.

3.1.1 Murata SCL3300 Inclinometer

The SCL3300 is 3 axis smart MEMS accelerometer from the manufacture Murata. The sensor is an accelerometer with inbuilt processing capabilities for inclination calculations. The sensor is built on Muara's 3D MEMS technology, and it is Murata's third generation MEMS technology, which come packed with extra features (several of them are out of this study's scope and is not presented). The main smart features for the SCL3300 are:

- Signal processing
- Different modes to adjust sensitivity and dynamic range
- Self-diagnostic
- Self-calibration on command
- Low pass filtering
- Digital interface

To get the full specification of the SCL3300 visit the manufactures home page [21].

3.1.2 TDK IIM42652 Gyroscope

The IIM42652 gyroscope is a new six axis solution from TDK portfolio. Despite its small size the IMU comes packed with more features than the SCL3300. The main advantage by using this IMU is the capability to use the onboard First-In-First-Out (FIFO) register. The FIFO can be used to letting the host system to be in sleep mode while the IMU is filling up its own register, allowing reduction of power consumption. The FIFO will be usable for long time measurements that relays on battery power. The main smart features for the IIM42652 are:

- Signal processing
- Different modes to adjust sensitivity and dynamic range
- Self-diagnostic
- Self-calibration on command
- Low pass filtering
- Digital interface
- Trip and fall detection
- FIFO

The full specification of the IIM42652 can be found visiting the manufactures home page [22] and a picture of the IMU can been seen below in Figure 8.

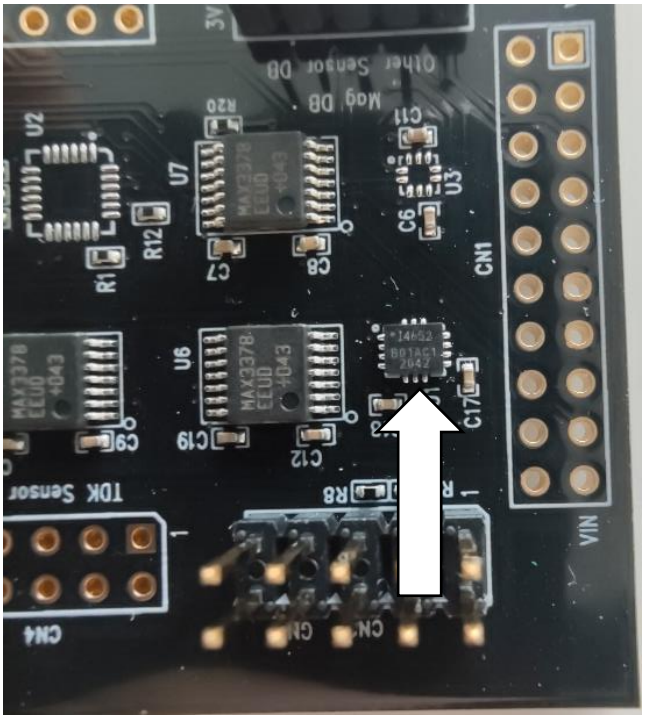


Figure 8: In the above picture is a pointer pointing on a 6 DOF IMU named IIM42652 made by the manufacture TDK

3.1.3 Pitch and Yaw Table

The APY002/M is a pitch and yaw table made by Thorlabs, its primary made for application as alignment of fiber optic and laser. Hench it has the accuracy that is necessary for conduct the test of interests, and with a known reference point could the sensors be evaluated. The table has a motion range of ±4 degrees with an accuracy on 10 arcsec. During testing of the sensors was the manual knobs used for adjusting the axis. Hence uncertainties caused by human error could occur. The table has the potential to change from manual knobs to electrical driven actuators and result in minimizing the human error and the motion can be programmed. The table and the two evaluation boars with respective sensors on, can be seen below in Figure 9.

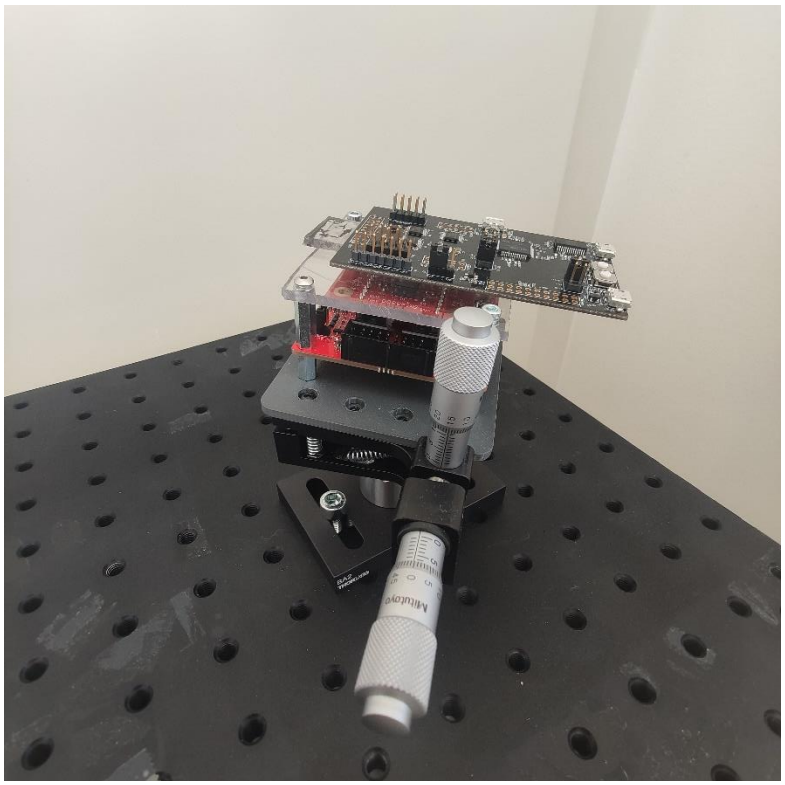


Figure 9: The picture includes two evaluation boards, pan tilt table, and a 3D printed anchoring plate. The tilt table are also screwed on a stable plate.

3.2 SOFTWARE FOR EVALUATION

To simplify and save time of the evaluation of the two sensors two evaluation softwares were used. The two IC chips manufacturers have made evaluation programs for their sensor chips which simplifies the evaluation of the sensors where no programming is necessary. The Murata SCL3300 inclinometer was used with *"Murata MEMS Demo"* software, which simplified the evaluation process [21] [14]. Instead of building up a data logging system from scratch, this evaluation kit will handle time sampling and communication with the SCL3300. This is done with an Atmel SAM4S MCU that acts as a middleman between the host PC and the MEMS inclinometer. This gives the user the ability to change between different mode of operations, data recording and error handling.

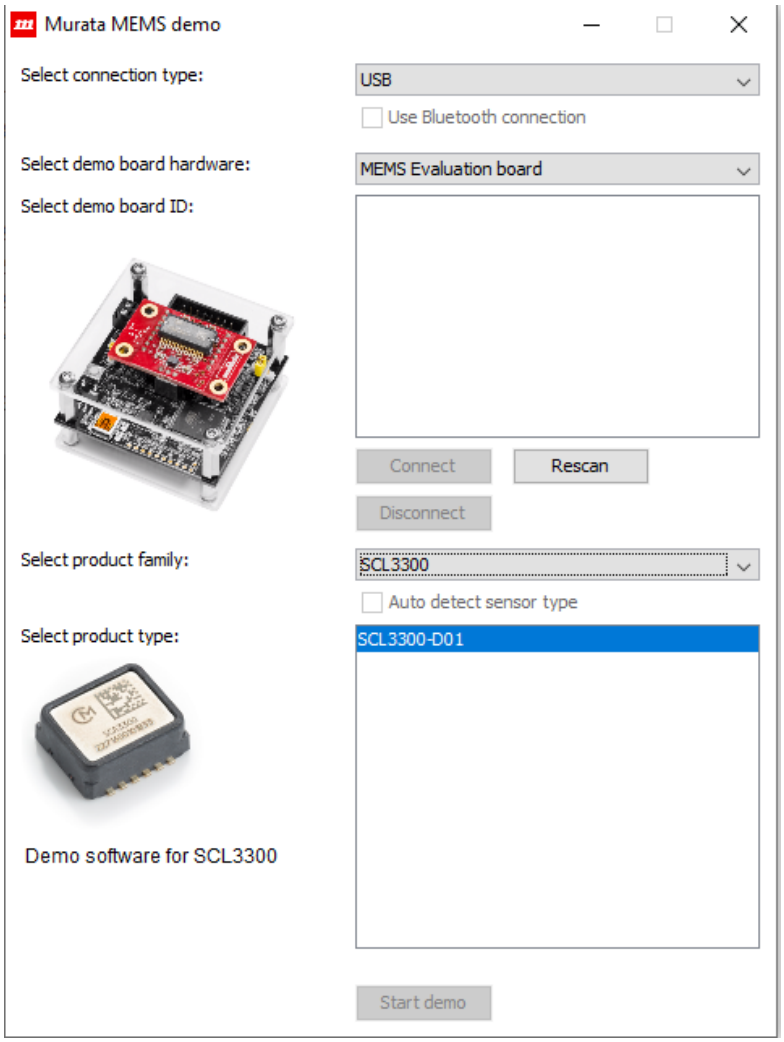


Figure 10: The above figure illustrates "Murata MEMS Demo" software. The same MCU evaluation board can be used to evaluate many different sorts of MEMS sensors; hence it is easy to evaluate different sensors with the same MCU and software.

Some of the features previously discussed in the section "smart MEMS" will be seen in the SCL3300. For instance, the possibility to change between operation mode (1-4 in this case) to suit different applications on the market. Mode 1 is the sensor's multi-functional operation mode for wider range of acceleration, inclination, and higher filter cut off frequency. Mode 4 is used purely as inclination mode and gives the highest possible resolution and a 10Hz Low Pass filter (see below in 11).

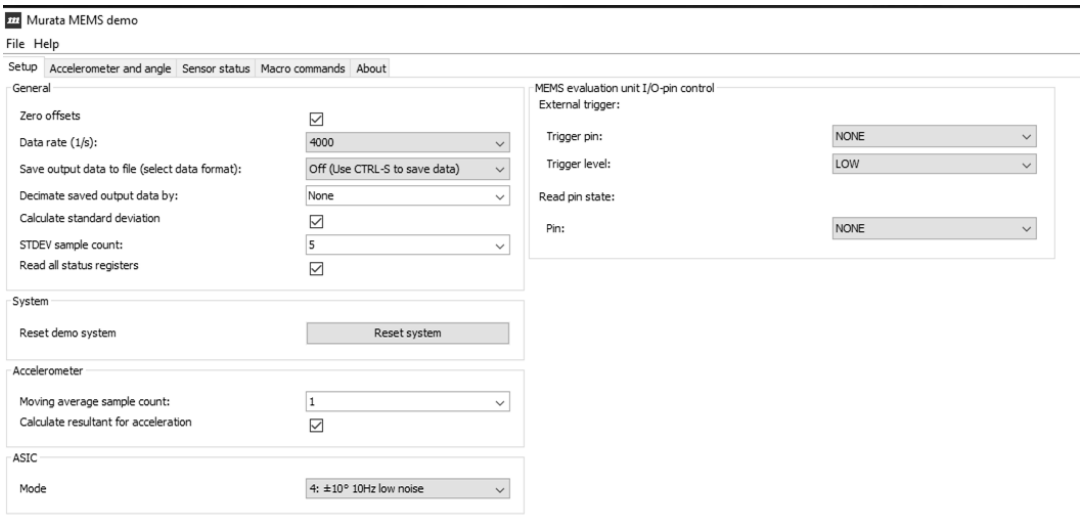


Figure 11: The above figure illustrates the different settings that are possible to select when running the SCL3300 on Murata's MEMS Demo software [14]

One of the most useful features in *Murata MEMS Demo* was the ability to store data with "time stamps". Time stamped data are time information when each data point was captured. Hence, it is possible to backtrack the data and see when special event has occurred or if the development MCU has failed to capture a time stamp at a given point.

Evaluation of the sensor from TDK IIM42652 was done with "SmartMotion Platform 1.9.6" software. It has almost identical functionalities as the software used for the Murata, but it is missing information about when the data was taken. It will only be returning the index number for each data point. Therefore, the user needs to know the sampling speed to be able to correct the time axis.

A potential problem with software made by a third part, is to know if the software is fully functional or not. Hence, this type of evaluation boards used in this thesis should not be expected to fully utilize the sensors full potential. This fact is based on a bus that was discovered, and possibilities to be more could have a noticeable impact on the evaluation of the IMUs. The bug found in "SmartMotion Platform 1.9.6" was found when the data plots did not have the same smooth shape on the Allan deviation plots and the files seems less dens on data. It was later observed that even the 25 samples per second generated more data than an option on 500 (The TDK support has been notified). The TDK chip IIM42652 is used by the "SmartMotion software", it can also record data, and select between different mode of operation.

4 EXPERIMENTAL TESTING

This chapter clarifies what kinds of experimental tests that were conducted during this thesis. It will answer why certain tests were conducted, what the hypothetical and/or expected results was, and how they were performed. First, the simulations (performed in Matlab) of the sensors are presented. After that, the real experimental tests on hardware are presented.

4.1 SIGNAL PROCESSING AND SIMULATIONS USING MATLAB

A cost-effective way for testing systems and applications, is to simulate them. The simulations were done in sensor fusion which can be found in Matlab. Sensor fusion has many inbuilt functions but for evaluating the sensors only a handful of them was used.

A cost-effective method for testing an IMU is to simulate the sensor in an artificial environment. The IMU itself will be modeled using the parameters given in the IMUs data sheet. To be able to do this the Sensor Fusion toolbox in Matlab was used. The toolbox includes functions that make it possible to simulate gyroscopes, accelerometers, magnetometers, GPS- sensors, and much more.

The name, sensor fusion, implies that multiple sensors can be fused together to improve the accuracy of the measurement. This can be seen in [6], where the paper compares two types of IMUs and a GPS for determine the positioning of a car. The highest accuracy and less drift of the position was established when booth the GPS and the IMU was working and used, while when the GPS lost its position less accurate results were given of the position. The same thing can be done in sensor fusion and different types of filters can be used to get even better final estimation of the system of interests.

The same approach of fusing multiple types of sensors and transmission performance parameters can be used for finding the optimal alignment of an antenna in telecommunication, and it will result in a better estimation in which direction the beam steering should concentrate its transmission signal. This could be done when combining the data from an IMU, with performance parameters given by the network. Typical parameters that are typically for performance in telecommunication are signal to noise ratio (SNR) and network speed.
Before conducting simulations on the sensors data was needed from each IMU's data sheet. Usually, the data sheets can provide enough information to make a model of the device with similar performance as the real one. The limiting factor is how complex the simulation model is and how well detailed the data sheet is.

The working principle in sensor fusion are following:

1) Create the characteristic parameters of the IMUs of interest by data from manufacture's datasheets.

```
%%%%%%%%%%% Murata %%%%%%%%%%%%%%%%%
%Data found in the data sheet
g = 9.819;
LSB_Murata = 12000;
Measurment_range = 1.*g;
Resolution_Murata = (g) / LSB_Murata;
Noise_Density = [20 15 20].*(g*10^(-6));
Constant_bias = [0 0 0].*(g*10^(-3));
Bias_instability =[4 6 4].*g*10^(-3);
Temperature_bias = [0.098 0.098 0.147].*(g*10^(-3));
Temperature_scale_factor = 0.3;
Axes_Misalignment = 0.2;
Random_Walk =  [20 15 20].*(g*10^(-6));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2) Pass the parameters to an IMU object, in other words, give the simulated IMU same parameters as the real example (if possible, not all manufactures give the required parameters and there is not a same standard used to name the parameters).

```
params = accelparams('MeasurementRange',Measurment_range,...
     'Resolution',Resolution_Murata,...
     'ConstantBias',Constant_bias,...
     'TemperatureBias',Temperature_bias,...
     'TemperatureScaleFactor',...
     Temperature_scale_factor,...
     'AxesMisalignment',Axes_Misalignment,...
     'NoiseDensity', Noise_Density,...
     'RandomWalk' , Random_Walk,...
     'BiasInstability',Bias_instability)

imu = imuSensor('SampleRate', Fs, 'Accelerometer', params);
orient = quaternion.ones(numSamples, 1);
acc = zeros(numSamples, 3);
angvel = zeros(numSamples, 3);

%Get simulated data from sensor object
accelData = zeros(numSamples, 3);
%for i = 1:numSamples
%imu.Temperature = temp(i);
accelData(:,:) = imu(acc(:,:), angvel(:,:));
```

3) Run the IMU object and store the generated data.
4) Now you are done with the generation of data, next step is to gathering data from the hardware itself and a comparison can be done if needed.

The data sheet provides the IMUs working characteristics and limits. The SCL3300 specification can be read here [21] and the TDK is given in [22].

4.1.1 Gathering IMU Parametric Data

One of the experiments was to gathering data from the IMU. This was performed with software from the evaluation boards made by respectable IMUs manufacturer (see section 3.2 for evaluation software). The following plot (Figure 12) illustrates an example of how the capturing window in Murata MEMS demo software looks like. The software that was used for gathering data from the TDK has nearly identical workflow as the Murata.

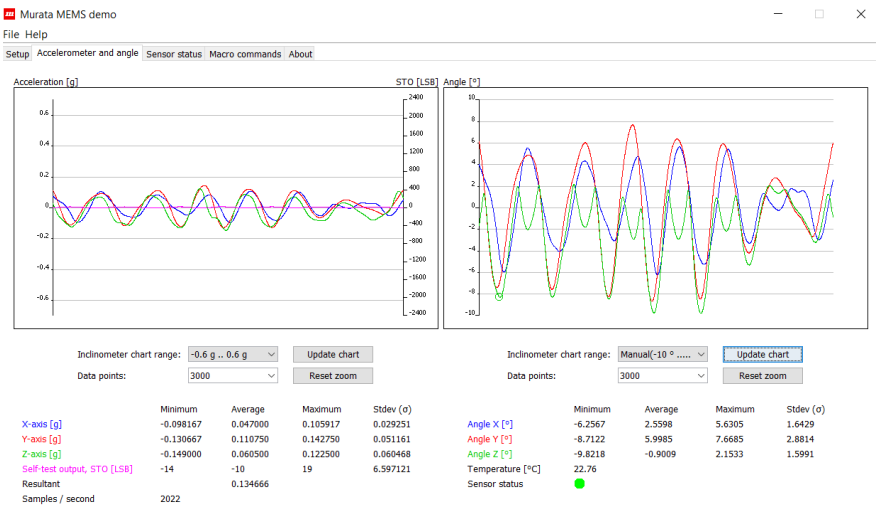


Figure 12: The capturing window of the Murata MEMS demo software.

The tests can be sensitive to the surrounding environment. Hence, it is important to plan accordingly. An example is static testing where it should be conducted with no temperature difference. The static test was important to quantify how noisy and accurate the sensor was. Knowing the margin of error makes it possible to determine if the sensor is of interests to investigate or if it has so large noise level that it cannot be used. The last test was performed to quantify the inclination performance of both sensors.

4.2 Static Tests

In the static tests, the goal was to estimate the stability of the two MEMS sensors and what the long time and short time noises have for impact. Therefore, the Allan deviation was used to find how much different noise sources impacted the sensors' performance, as well using the normal standard variance.

This test was conducted over an eight-hour time in a static environment. It is important that the environment is static due to the method's requirements. Therefore, it is important to have as fixed temperature as possible and as little external noise as possible. Otherwise, the measurement will be affected by temperature and external noise changes and if so, it will instead be more suitable to do a dynamic test [1] [2]. The eight-hour long static test was conducted in an empty apartment. However, in the apartment the temperature was not static which drives one uncertainty in this experiment. To reduce this uncertainty, a temperature regulated laboratory room should be used to which will minimize errors caused by temperature changes. Both sensors were used with their respective high-performance mode, which yields the best possible accuracy of the devices. The gathered data from the long-time test can be seen in Figure 13.

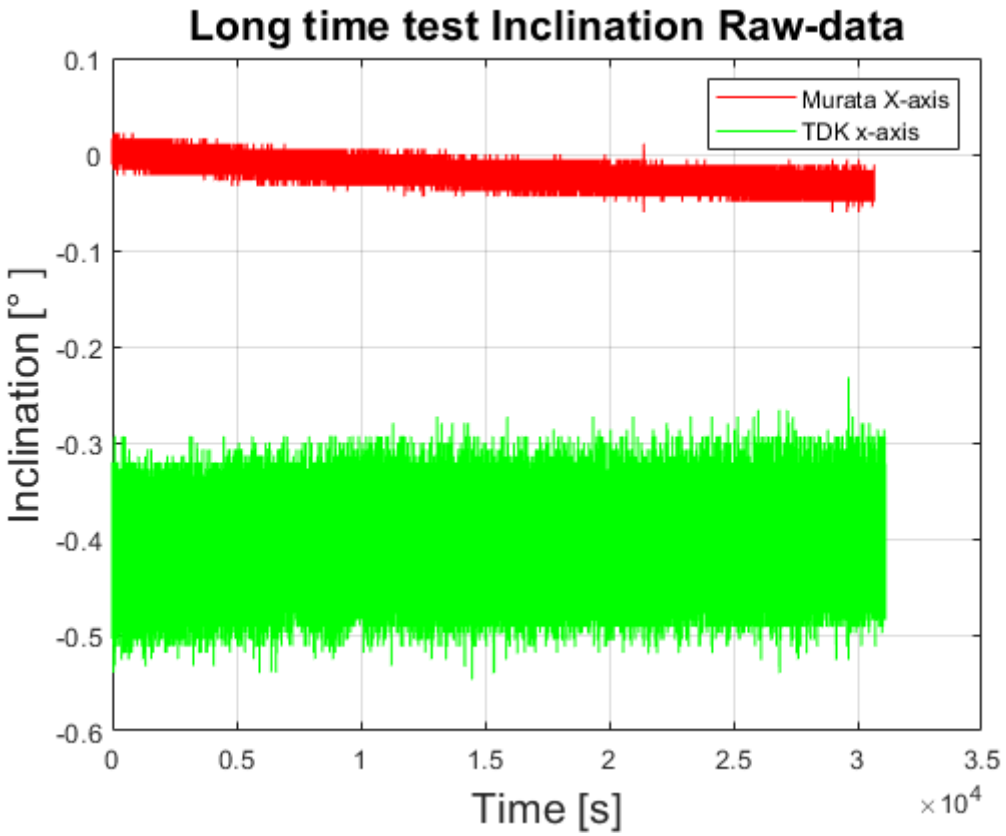


Figure 13: Data sampled during an eight-hour hour static test. It is a noticeable difference between the two sensors, the Murata SCL3300 inclinometer has less noise and more precise accuracy. While the TDK IIM42652 has much higher noise levels.

It is well noticeable that the signal from the TDK has more noise in its signal. The normal standard variance of the TDK is much wider than the Murata, see Table 1.

Table 1: The table below Table on the 3sigma standard deviation and the samples mean values.

	Mean	Variance	3σ
TDK IIM42652	-0.4045	9.9557e-04	0.0947
Murata SCL3300	-0.0197	1.1874e-04	0.0327
TDK/MURATA		8.38ggr	

The above table indicates that the TDK has 8.38 times wider variance between its samples.

Calculating the Allan deviation should then point in the same direction, see (Appendix 10Appendix) for Matlab code example. The AVAR plots is depicted in Figure 14 and Figure 15.

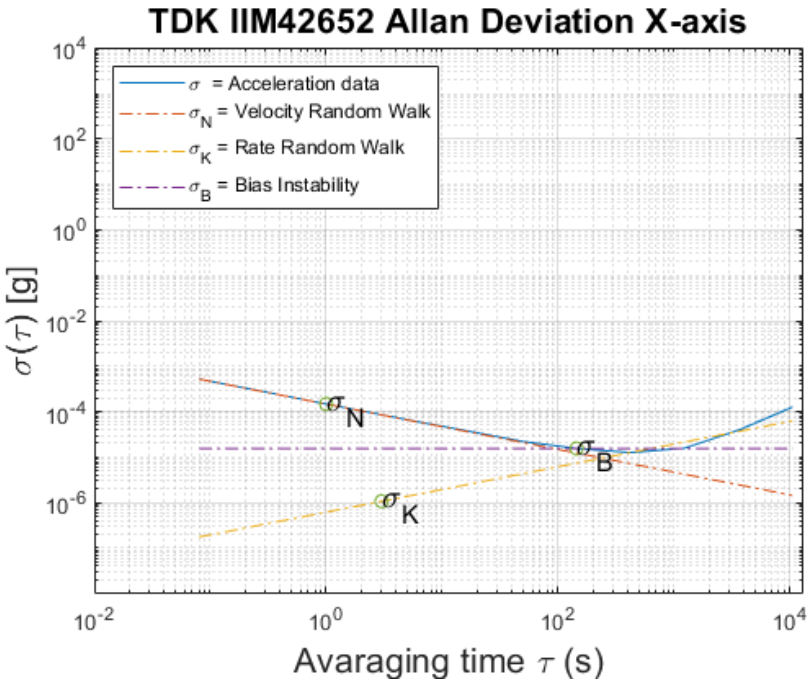


Figure 14: Allan deviation plot of the TDK IIM42652

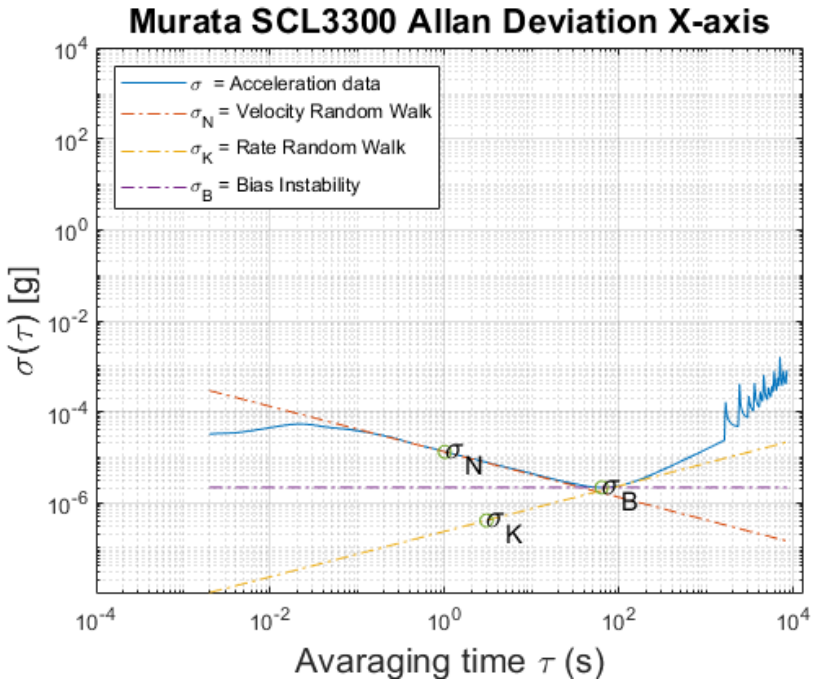


Figure 15: Allan deviation plot of the Murata SCL3300

From the Allan deviation plot above, can it visually be distinguished which MEMS that has the lowest noise parameters. Hence, given the sensors settings and measured data has the Murata SCL3300 lower noise levels then the TDK IIM42652. The parameters stored as variables and can be seen in Table 2.

Table 2: Allan deviation data from which was found in Figure 14 Figure 15

	Velocity Random Walk	Bias instability	Acceleration Random Walk
Hz = 1/time unit	$\frac{m}{s^2}\frac{1}{\sqrt{Hz}}$	$\frac{m}{s^2} = \mathrm{g}$	$\frac{m}{s^3}\frac{1}{\sqrt{Hz}}$
Murata	$0.1298 \times 10^{-4} = 12.98\ ug/\sqrt{s}$	$0.0214 \times 10^{-4} = 2.14\ ug$	$0.004 \times 10^{-4} = 0.4\ ug \times \sqrt{s})$
TDK	$1.470 \times 10^{-4} = 129.8\ ug/\sqrt{s}$	$0.155 \times 10^{-4} = 15\ ug/\sqrt{s}$	$0.011 \times 10^{-4} = 1.1\ ug \times \sqrt{s})$

The test results from the Allan deviation suggests that the Murata has lower noise values over the hole time span, (the TDK has only better performance when the sensors has been on for more than 23min without recalibration). The Murata has also 10 times lower noise levels when compared to the TDK for estimated noise levels at short time interval.

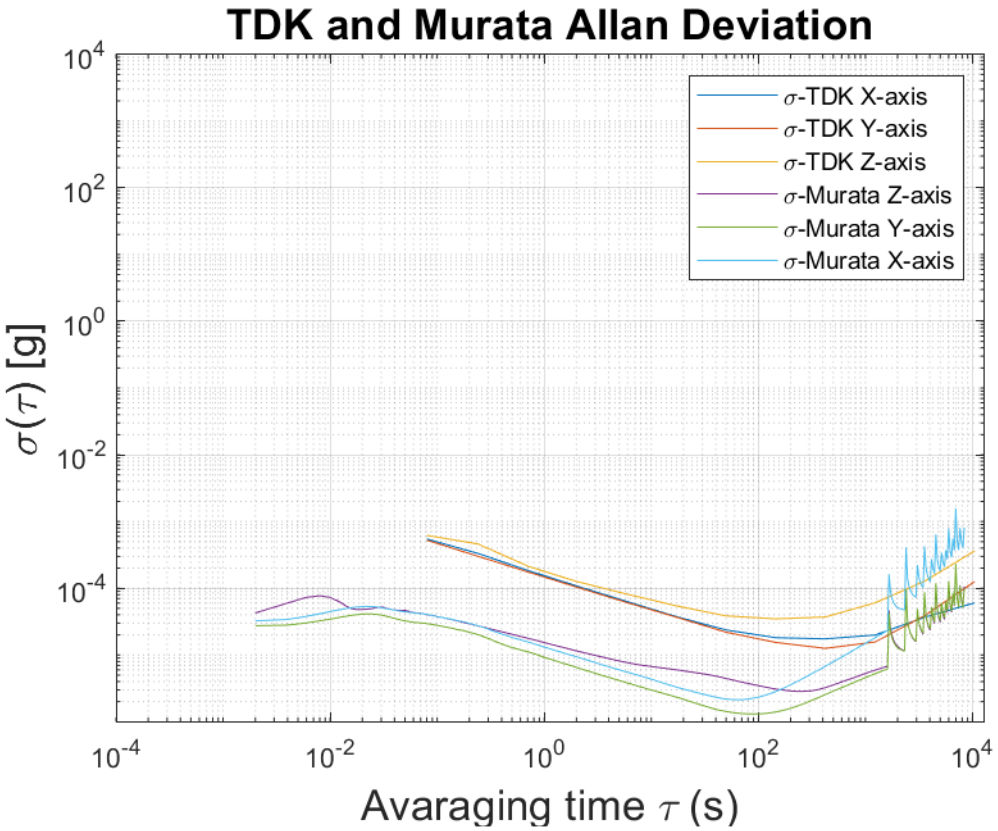


Figure 16: Allan deviation plots between Murata SCL3300 and TDK IIM42652, where lower values are preferred and indicates that the sensor are of good quality and have low instinct noise which is preferred.

4.2.1 Thermal Test

In this chapter, one thermal test is presented. The thermal test was conducted to see how much drift of the bias that was affected by rapid change in temperature. Rapid change in temperature is common in countries which have high plateaus climates, as well in cities where reflection

and shadowing of sunlight are common. These types of tests are important because the static test will not show how the bias is affected by the heat.
This test was conducted to see how fast the sensors drifted away when they were exposed to significant amount of heat in a short time. The test was performed using a kitchen oven given circumstances caused by Covid-19 restrictions. A dedicated heater made for testing of electronic devices would be preferred to use instead of a kitchen oven since it has much more stable temperature regulation and does not induce the same vibrations. The tests were conducted from ambient temperature to a maximum around 80°C. It is necessary to check the max allowed temperature on the components, otherwise its ease to break something.

In this test the Murata SCL3300 was used with its acceleration mode and had therefore a 40Hz LP filter and half of the accuracy compared with its inclination mode. Figure 17 shows how the thermal tests were conducted (see Appendix 7) for more pictures. A plate was used to protect the sensors from direct radiated heat, otherwise they can be grille by the oven temperature hysteresis.

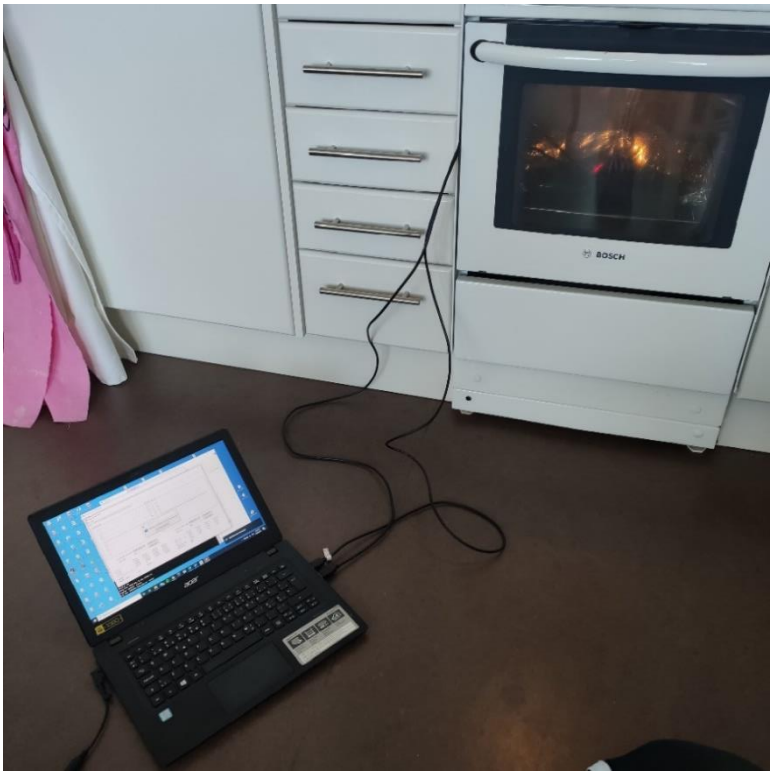


Figure 17: Thermal testing by using kitchen equipment.

Figure 18 depicts an example of generated heat data from the tests. From the figure it can be noticed that both the sensors have drift caused by temperature. In Figure 19, two tangents have been approximated between the data's endpoints. It is visually noticeable that the TDK has a bigger drift (the lower data in the plot) when comparing it with the Murata (the higher data in the plot). Temperature drift and the results from the long-time test point toward Murata as the better alternative if the two sensors are compared.

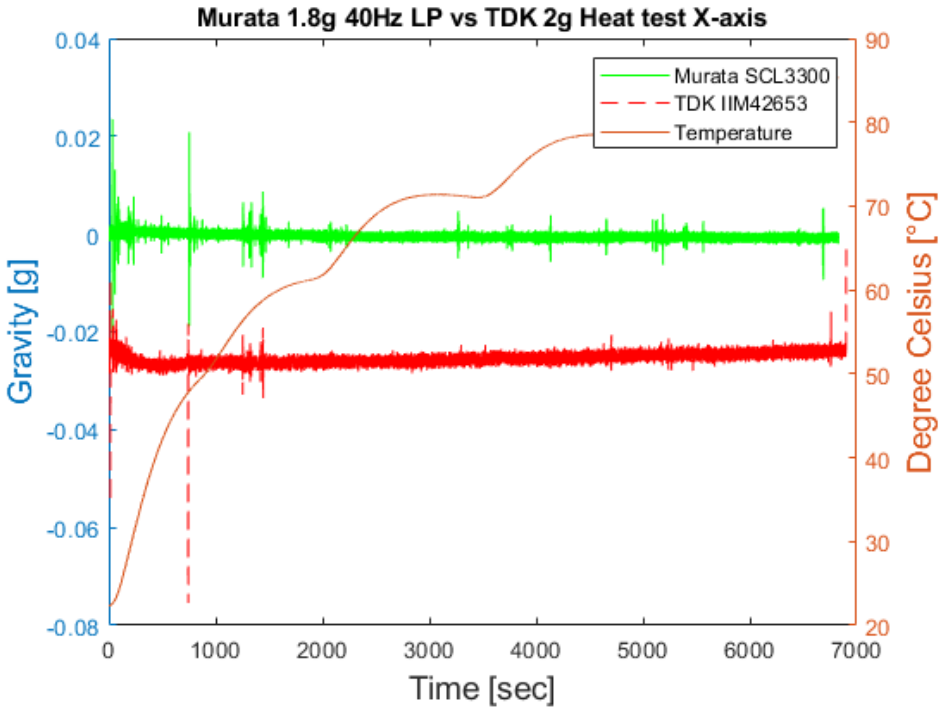


Figure 18: Heat test of the sensors. This test compared the Murata's accelerometer setting mode 1, and it will therefore drop half of its resolution compared to inclination setting.

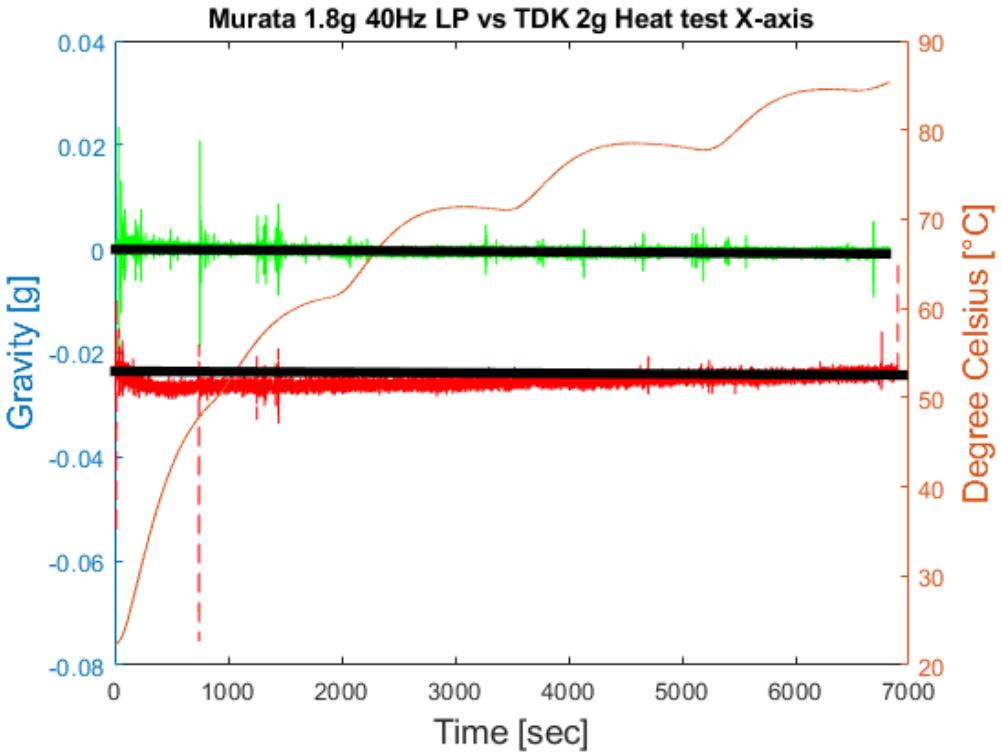


Figure 19: The above figure illustrates how the heat affect the sensors. The black lines are two approximate straight line between the endpoints of the sampled data mean values. It shows that the TDK has the biggest drift.

4.2.2 Inclination Testing with Tilt Table

One of the key points in this project was to measure the inclination with as high accuracy as possible. Based on the facts gained in section 6.2 Static Tests, was it noticed that the Murata SCL3300 has the lowest noise levels, therefore will only one test of booth sensors be presented together. And a separate test will be presented for the Murata SCL3300, which shows how small inclination it can measure. The SCL3300 has an option to use the inbuilt processing power to calculate the inclination angle. Meanwhile, the inclination on the IIM42652 was needed to be calculated. This can be performed using its onboard accelerometer sensors and using equation 26.

26

$$\theta = tan2d\left(\frac{g_x}{\sqrt{g_z^2 + g_y^2}}\right)$$

The equation tan2 can be found in almost all popular programing language libraries, and tan2d is just a Matlab version, which returns the answer in d = degrees. It is also the same formula used internally by the SCL3300.

The tilt table was used in the inclination tests and the first test was a series of incremental steps where each step was an incremental of 500um, which corresponds to an inclination of 0.58139°. Six steps were taken in each direction, with a start and ending at a horizontal plane of zero-degree inclination, see Figure 20.

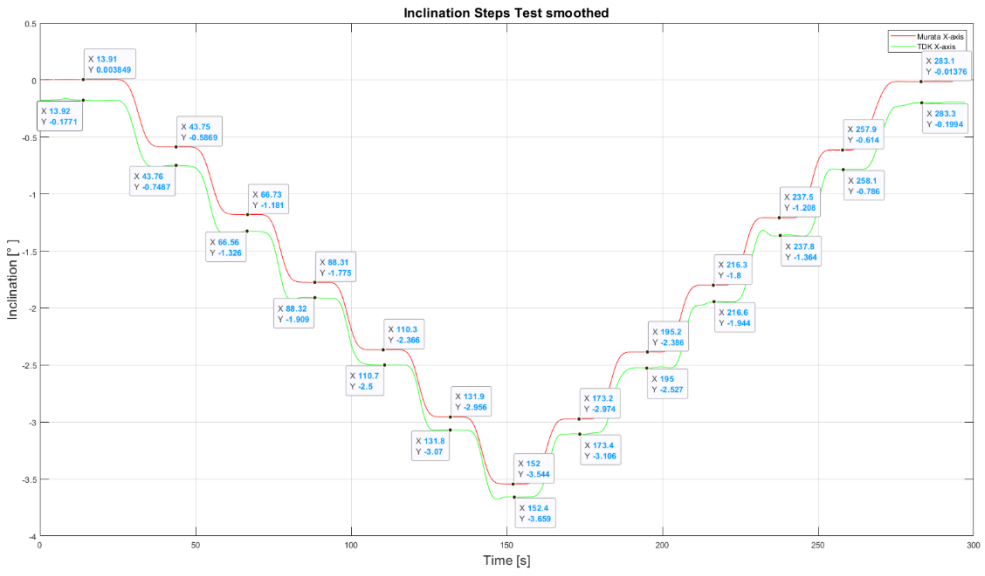


Figure 20: Processed data from an inclination test. The green (lowest line is the TDK and the red (highest line) is the Murata (for bigger picture see Appendix 5).

In Table 3 shows a comparison of the data which is not generated from the smoothed data visualized in Figure 20. The data has been taken from its raw data and processed (raw data see Appendix 5). The data given in the plot is only an illustration of what it looks like when the data has been filtered.

Table 3: The measured angles of the two sensors compared with the theoretical set value.

Tilt table°	Murata SCL3300°	Error °	TDK IIM42652°	Error °
0	0.003849	0.003849	0.1771	0.1771
0.5813	0.5869	0.0056	0.7487	0.1674
1.1627	1.181	0.0183	1.326	0.1633
1.7441	1.775	0.0309	1.909	0.1649
2.3255	2.366	0.0405	2.5	0.1745
2.9069	2.956	0.0491	3.07	0.1631
3.4883	3.544	0.0557	3.659	0.1707
2.9069	2.974	0.0671	3.106	0.1991
2.3255	2.386	0.0605	2.527	0.2015
1.7441	1.8	0.0559	1.944	0.1999
1.1627	1.208	0.0453	1.364	0.2013
0.5813	0.614	0.0327	0.786	0.2047
0	0.0137	0.0137	0.1994	0.1994

To be able to do this sort of test correctly (i.e., in order to get accurate results) a motorized tilt table is a must. The human error is one of the biggest factors for errors in the inclination tests. Therefore, these reading results should be taken with care.

The second inclination test was conducted only to verify how accurate the Murata SCL3300 was. This test was also performed with six steps of incremental steps. Each step corresponds to 0.05813° increment and an illustration of it can be seen in Figure 21. The non-filtered data can be seen in Appendix 4.

Data from the test can be seen in Table 4, which shows that the SCL3300 has a very accurate inclination capability. This is without using any digital filtering of the data more than what is included in the sensor itself. The illustrated data in Figure 21 were not used to compute the data presented in the Table 4. The data were extracted from the plots given in Appendix 4, where only the stable regions on the steps were used for calculating the inclination, mean value, and deviation.

Table 4: The table consists of data from the smaller incremental test done on the Murata SCL3300

Tilt Table Angle	SCL3300	Compensate the offset error
Resolution 0,00277778	Mean value ± 3σ	
0°	-0.0483 ± 0.0164°	0 ± 0.0164°
0.05813°	-0.1051 ± 0.0152°	-0.0568 ± 0.0152°
0.11627°	-0.1657 ± 0.0179°	-0.1174 ± 0.0179°
0.17441°	-0.2238 ± 0.0166°	-0.1756 ± 0.0166°
0.23255°	-0.2856 ± 0.0169°	-0.2373 ± 0.0169°
0.29069°	-0.3467 ± 0.0150°	-0.2984 ± 0.0150°
0.34883°	-0.4108 ± 0.0148°	-0.3625 ± 0.0148°

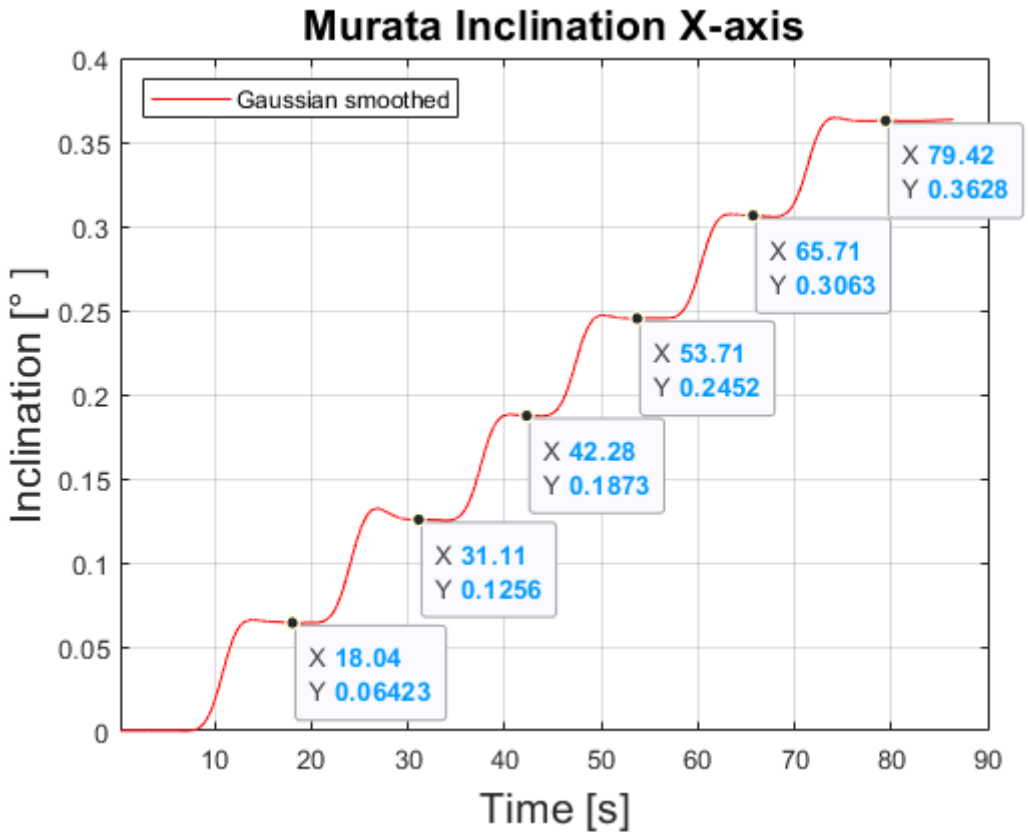


Figure 21: The precision test on the Murata SCL3300; the data has been filtered to take away noise which was caused during adjustments of the tilt tables inclination.

4.3 UNCERTAINTIES

This section highlights the used methods' uncertainties.

A known reference: To know if the measurement is right or not a known reference point is needed (e.g., a high precision tilt table). This make it possible to know what the measurement should be when reading the sampled data from the sensors.

A manual reference: Will induce human errors when adjusting the table inclination. This can be canceled out if a motorized drive stage is used to control the device.

Human errors: Caused by programming or by evaluation software has an impact on the results. The first thing to do is to check if all functionalities in the evaluation software are correct before doing any testing. During the testing of the MEMS was 500 samples per seconds used. After many hours of testing and troubleshooting Matlab cod, was a bug discovered in the "SmartMotion Platform 1.9.6" software. The option to use 500 samples per second gave instead 12.5 samples per seconds. Which is another inbuilt option which can be used if the configuration bits are initialized wrong. Therefore, it can be important to test other settings, so they are not incorrect, such as:

- Sample time, filter settings, time between samples.
- Calibration

- Placement of sensor

Environmental effects: The impact of environmental effects on the measurements is something that needs to be considered. It can be everything from temperature, external vibrations, or just the sound from the PC power supply. An example of this can be seen in Figure 22, which illustrates the spectra power density of the measurement data signal. It can be seen that a 50Hz AC signal has eaten its way into the MEMS device. The below data comes from the laboratory room in MC2.

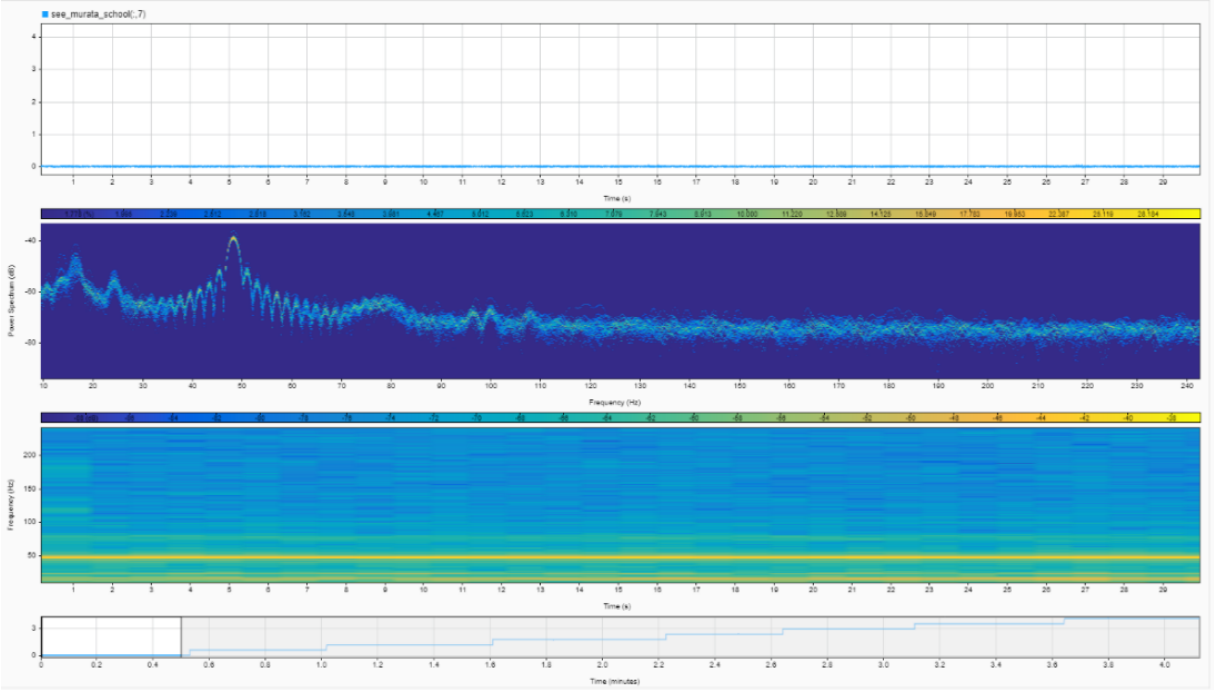


Figure 22: Frequency spectra of measurement data. A 50Hz signal have caused some disturbances in the data, it has almost no effects on the measurements. This can be solved by having extra filtering of the data.

5 SUMMARY

Given the todays circumstances have two MEMS been evaluated if they have the needed performance to detect misalignment on backhaul antennas. In this report, it has been verified that simple trigonometry can be used to generate the same inclination angle as onboard hardware. It has also been verified that onboard signal processing and compute of inclination, works as good as calculate it by trigonometry. If a host device has limiting computational power could a MEMS with internal computing capabilities be used to offloading the host. It has also been showed that it is necessary to have a good reference point to evaluating the inclination. In the report also has three different noise characteristics been presented, which was one of the evaluating methods used to evaluate the MEMS. The report describes and points out how to set up a model of the MEMS in sensor fusion. It is noticed that the data sheet parameters were not enough to fully utilize the simulation software. It has been confirmed that sensitivity of the MEMS can get worse if a bigger measurement range and higher cutoff frequency is used.

The conclusion which can be drawn from the projects data, states the Murata SCL3300 is the best candidate for inclination detection, based on less noise, higher accuracy, and more robust design.

For future work would it be interesting to use a motorized actuator for steering the tilt table, which make it possible to do dynamic testing of MEMS and simulating mast movements. Future work may consider investigating into merging antenna data with a set of sensors to get the best possible estimation. This work has gathered the needed sources to be able to do a correct evaluation of the sensors in a static system. It has also pointed out one well-established standard used for qualifying noises in precision instruments and to find random drifts. The sources; IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology—Random Instabilities [1] and IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Laser Gyros [2] are two well described standards, which the last named includes information about dynamic and static modeling, and a “A-Z guide” on a how a complete testing procedure of (laser gyros) on a professional level. The source is not the same as a MEMS accelerometer, but it will give the insight needed on what are needed to fully quantify the sensors performance and if a modeling of them is of interests.

REFERENCES

[1] IEEE-SA Standards Board, “IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology--Random Instabilities,” *IEEE Std*, vol. 1139, 2008.

[2] “IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Laser Gyros,” *IEEE Std 647-2006 (Revision of IEEE Std 647-1995)*, pp. 1–96, Sep. 2006, doi: 10.1109/IEEESTD.2006.246241.

[3] G. Langfelder, "Microelectromechanical systems integrating motion and displacement sensors," in *Smart Sensors and MEMS Edition*, Second Edition ed., Milan, Italy, Elsevier Ltd, 2018, pp. 395-428.

[4] "cumatix," [Online]. Available: https://cumatix.se/imu/imu-teknik/. [Accessed 24 04 2021].

[5] V. G. G. Elisson, "Low cost relative GNSS positioning with IMU integration," Chalmers University of Technology / Department of Signals and Systems, 2014.

[6] S. C. a. B. S. Dr. Chris Goodall, "https://www.analog.com/," 27 04 2021. [Online]. Available: https://www.analog.com/en/technical-articles/the-battle-between-mems-and-fogs-for-precision-guidance.html. [Accessed 27 04 2021].

[7] E. Udd, "SPIE," SPIE, 03 November 2016. [Online]. Available: https://spie.org/news/history-of-the-fiber-optic-gyro-in-spie-professional?SSO=1#:~:text=In%201976%2C%20Victor%20Vali%20and,year%2C%20McDonnell%20Douglas%20Astronautics%20Co.. [Accessed 29 04 2021].

[8] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, United Kingdom, 2007.

[9] J. Watson, "MEMS GYROSCOPE PROVIDES PRECISION INERTIAL SENSING IN HARSH, HIGH TEMPERATURE ENVIRONMENTS," Analog Devices, Inc, Norwood, MA 02062-9106, 2016. [Accessed 27 04 2021]

[10] L. Bengtsson, Elektriska Mätsystem och mätmetoder, 1:2 ed., Lund: Lars Bengtsson and Studentlitteratur, 2012, pp. 32-40.

[11] M. Andrejašic, "MEMS ACCELEROMETERS," University of Ljubljana Faculty for mathematics and physics Department of physics, Ljubljana, 2008.

[12] http://www-personal.umich.edu/~bkerkez/courses/cee575/Handouts/5_adxl_theory.pdf. [Online] [Accessed 16 05 2021]

[13] J.Voldman, "course materials for 6.777J / 2.372J Design and Fabrication of Microelectromechanical Devices," in *MIT*, Massachusetts, 2007.

[14] Murata, "Murata," https://www.murata.com/, [Online]. Available: https://www.murata.com/en-global/products/sensor/library/mems-basic/evb. [Accessed 06 05 2021].

[15] W. J. Riley, “NIST Special Publication 1065 Handbook of Freequency Stability Analysis,” *International Journal of Rock Mechanics and Mining Science*, vol. 31, no. 1, 2008.

[16] https://tf.nist.gov/phase/Properties/nine.htm.[Accessed 15 05 2021]

[17] D. M. Pozar, *Microwave Engineering, 4th Edition*. 2012

[18] L.Barreda " Characterization of Errors and Noises in MEMS Inertial Sensors Using Allan Variance Method", Universitat Politecnica de catalunya Barecelonatech, Departament de Teoria del Senyal i Comunicacions, Barcelona, 2016 [Accessed 16 05 2021]

[19] "Mathworks," [Online]. Available: https://se.mathworks.com/help/nav/ug/inertial-sensor-noise-analysis-using-allan-variance.html. [Accessed 01 05 2021].

[20] Freescale Semiconductor, Inc, "NXP," 3 02 2015. [Online]. Available: https://www.nxp.com/docs/en/application-note/AN5087.pdf. [Accessed 25 05 2021].

[21] "Murata.com," Murata Electronics Oy, [Online]. Available: https://www.murata.com/-/media/webrenewal/products/sensor/pdf/datasheet/datasheet_scl3300-d01.ashx?la=en&cvid=20210316063715000000. [Accessed 30 04 2021].

[22] TDK InvenSense, "https://invensense.tdk.com/," 07 01 2021. [Online]. Available: http://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2021/01/DS-000440-IIM-42652-v1.0.pdf. [Accessed 17 05 2021].

[23] D. H. (. H. Titterton, Strapdown inertial navigation technology, 2nd edition, London; Reston: IET © 2004, 2004.

[24] Louis E. Frenzel Jr., *Principles of Electronic Communication Systems*, vol. 28, no. 10. McGraw-Hill Education, 2015

[25] J. E. S. C. T. E. Y. L. H. Z. Jonas Hansryd, "MICROWAVE backhaul evolution -Reaching Beyond 100Ghz," © Ericsson AB 2017 Ericsson, Gothenburg, 2017.

[26] https://se.mathworks.com/help/nav/ug/inertial-sensor-noise-analysis-using-allan-variance.html.

APPENDIX 1

Below is the most important part of Murata SCL3300 data sheet

Parameter	Condition	Min	Nom	Max	Unit
Measurement range	Mode 1 Mode 2 Mode 3, Mode 4[A]		1.2 2.4 -		g
	Mode 1 Mode 2 Mode 3, Mode 4[A]		±90 ±90 (±10)		°
Offset error [B]	All modes, X, Z channels	-20 -1.15		20 1.15	mg °
	All modes, Y channel	-25 -1.45		20 1.15	mg °
Offset temperature dependency [C]	-40°C ... +125°C, X, Y channels	-10 -0.57		10 0.57	mg °
	-40°C ... +125°C, Z channel	-15 -0.86		15 0.86	mg °
Offset lifetime drift [D]	All modes, X, Z channels	-8 -0.46	±4 ±0.23	8 0.46	mg °
	All modes, Y channels	-12 -0.69	±6 ±0.34	12 0.69	mg °
Sensitivity (acceleration output)	Mode 1 Mode 2 Mode 3, Mode 4		6000 3000 12000		LSB/g
	Mode 1 Mode 2 Mode 3, Mode 4 valid only between 0...1° [E]		105 52 209		LSB/°
Sensitivity (inclination output)	All modes		182		LSB/°
Sensitivity error [B]	-40°C ... +125°C Mode 1	-0.7		0.7	%
Sensitivity temperature dependency [C]	-40°C ... +125°C Mode 1	-0.3		0.3	%
Linearity error [F]	-1g ... +1g range	-4		4	mg
Integrated noise (RMS, accelerometer) [G]	Mode 3, X, Y, Z channels Mode 4, X, Z channels Mode 4, Y channel		0.13 0.08 0.06		mg_{RMS}
Noise density [G]	Mode 3, X, Y, Z channels Mode 4, X, Z channels Mode 4, Y channel		32 20 15		µg/√Hz
	Mode 3, X, Y, Z channels Mode 4, X, Z channels Mode 4, Y channel		0.0018 0.0012 0.0009		°/√Hz
Cross axis sensitivity [H]	per axis	-1.5	±0.2	1.5	%
Amplitude response, -3dB frequency	Mode 1		40		Hz
	Mode 2		70		Hz
	Mode 3, Mode 4		10		Hz

Below is the most important part of TDK IIM42562 data sheet

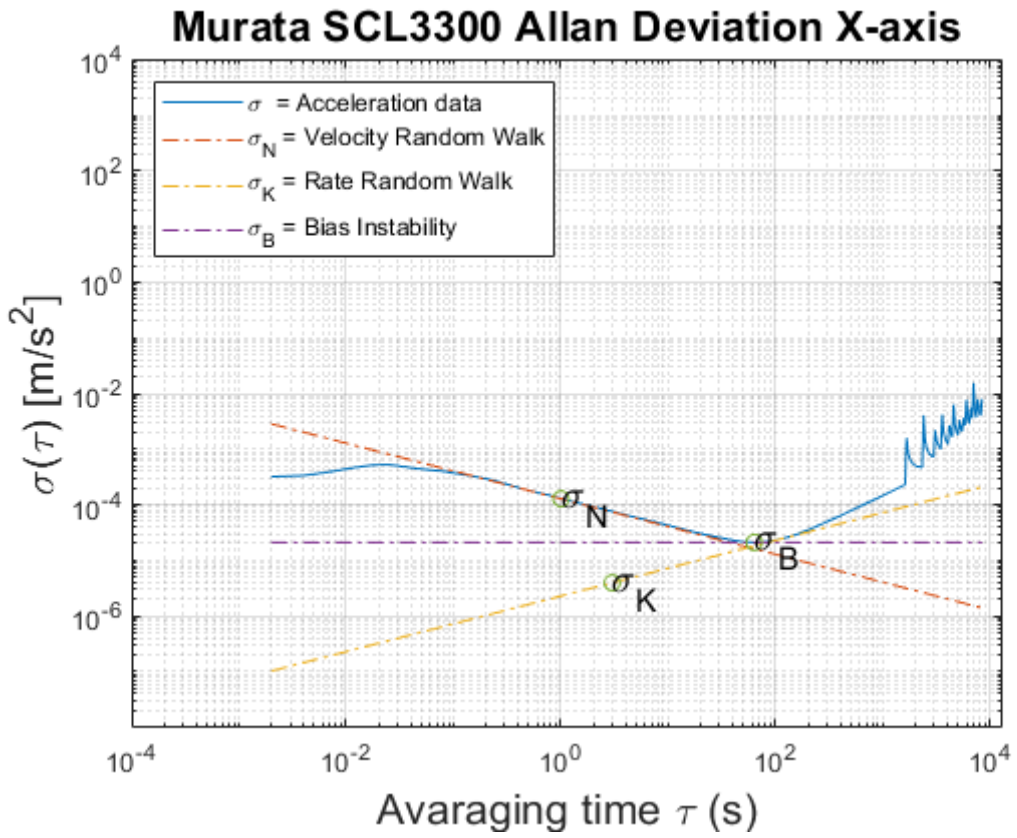
Typical Operating Circuit of section 4.2, VDD = 1.8V, VDDIO = 1.8V, T_A=25°C, unless otherwise noted.

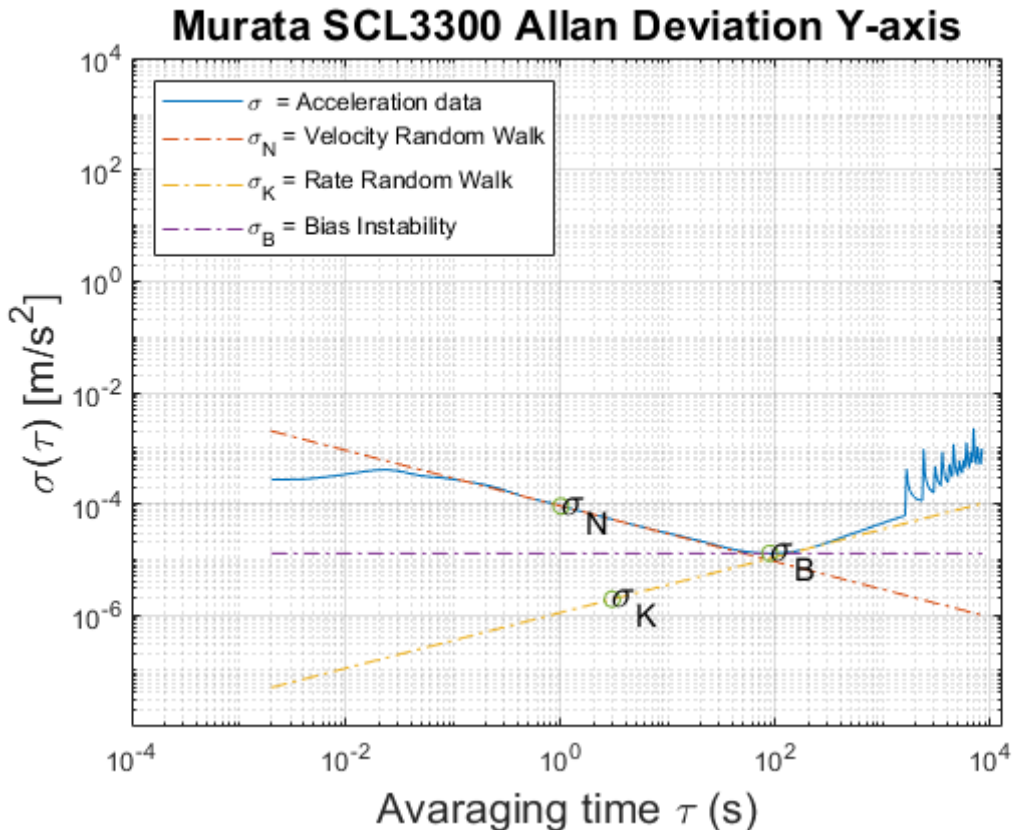
PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	ACCEL_FS_SEL =0		±16		*g*	2
	ACCEL_FS_SEL =1		±8		*g*	2
	ACCEL_FS_SEL =2		±4		*g*	2
	ACCEL_FS_SEL =3		±2		*g*	2
ADC Word Length	Output in two's complement format		16		bits	2, 6
Sensitivity Scale Factor	ACCEL_FS_SEL =0		2,048		LSB/*g*	2
	ACCEL_FS_SEL =1		4,096		LSB/*g*	2
	ACCEL_FS_SEL =2		8,192		LSB/*g*	2
	ACCEL_FS_SEL =3		16,384		LSB/*g*	2
Sensitivity Scale Factor Initial Tolerance	Component-level		±0.5		%	1, 5
Sensitivity Change vs. Temperature	-40°C to +105°C		±0.005		%/°C	3, 5
Nonlinearity	Best Fit Straight Line, ±2g		±0.1		%	3, 5
Cross-Axis Sensitivity			±1		%	3, 5
ZERO-G OUTPUT						
Initial Tolerance	Board-level, all axes		±20		mg	3, 5
Zero-G Level Change vs. Temperature	-40°C to +105°C		±0.15		mg/°C	3, 5
OTHER PARAMETERS						
Power Spectral Density	@ 10 Hz		70		µg/√Hz	1, 5
RMS Noise	Bandwidth = 100 Hz		0.70		mg-rms	4, 5
Low-Pass Filter Response	ODR < 1 kHz	5		500	Hz	2
	ODR ≥ 1 kHz	42		3979	Hz	2
Accelerometer Startup Time	From sleep mode to valid data		10		ms	3, 5
Output Data Rate		12.5		32000	Hz	2

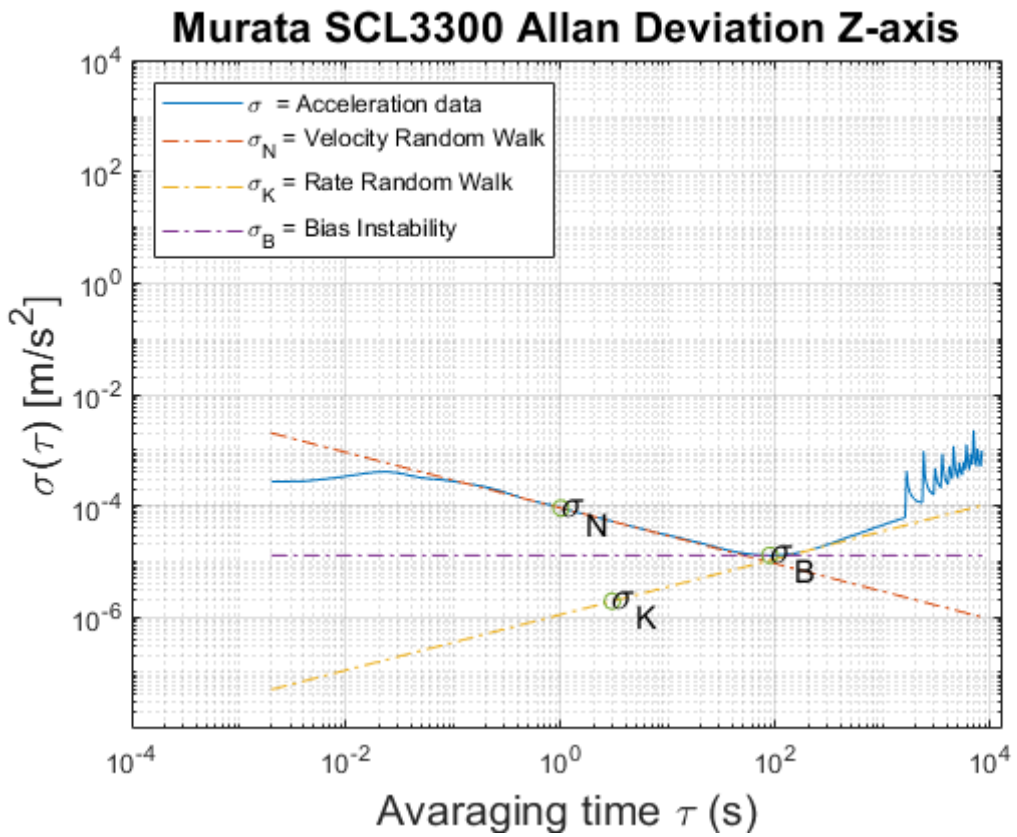
Table 2. Accelerometer Specifications

APPENDIX 2

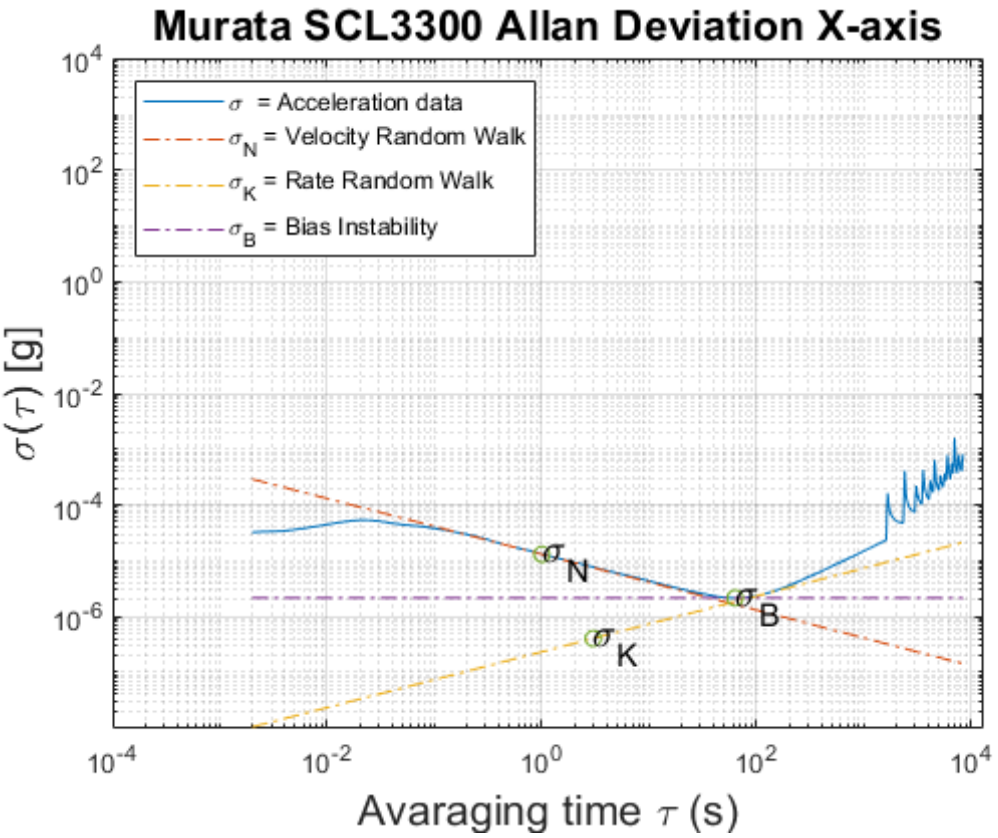
Allan deviation plots of Murata Inclinometer SCL3300







Following values is in g:



tauParams = 1×3

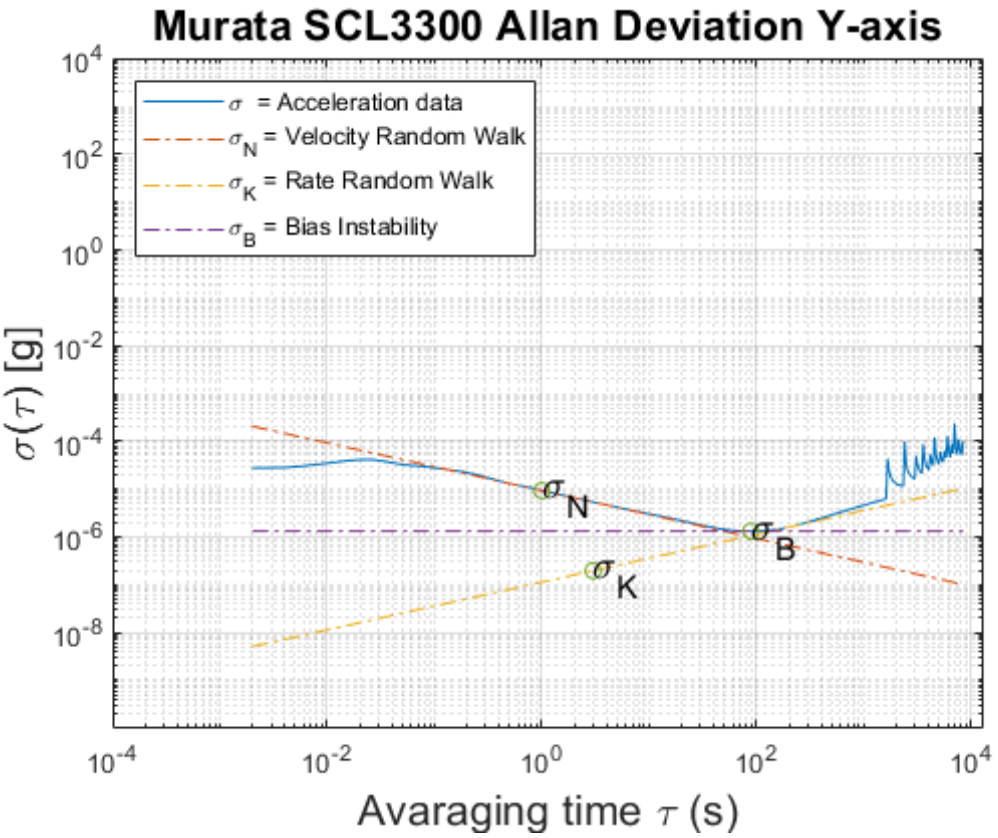
1.0000 3.0000 63.1240

params = 1×3

10^{-4} ×

0.1298 0.0040 0.0214

B = 3.2223e-06



tauParams = 1×3

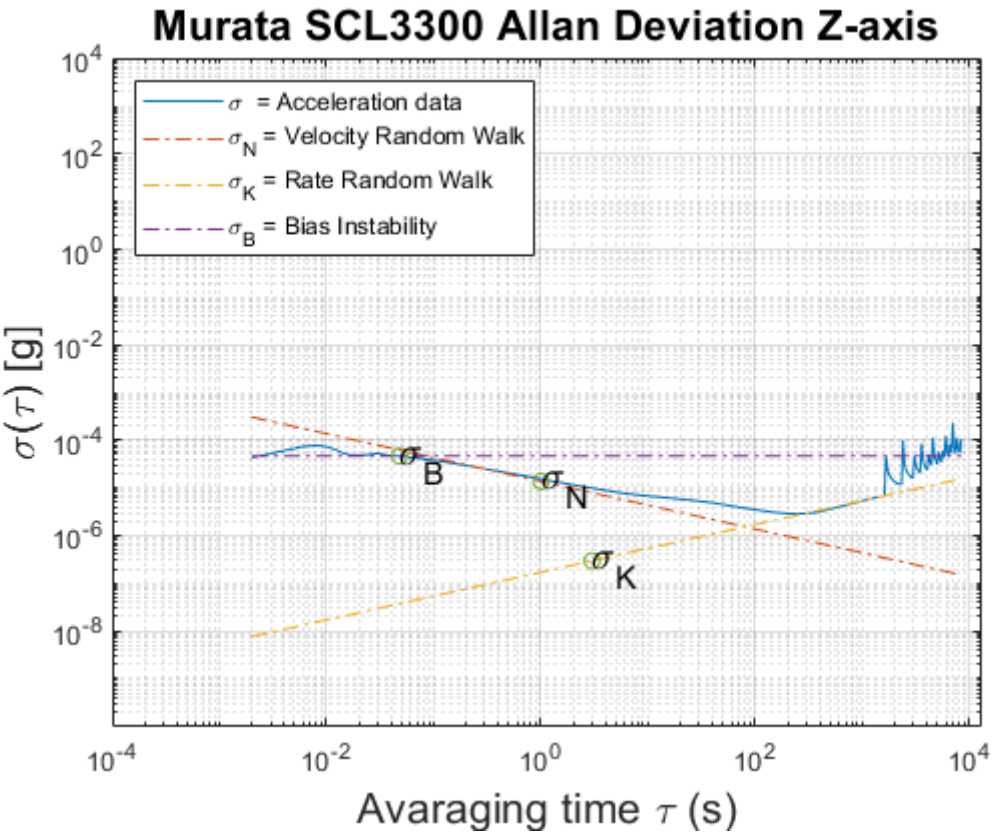
1.0000 3.0000 88.3460

params = 1×3

10^{-5} ×

0.9219 0.0195 0.1306

B = 1.9662e-06



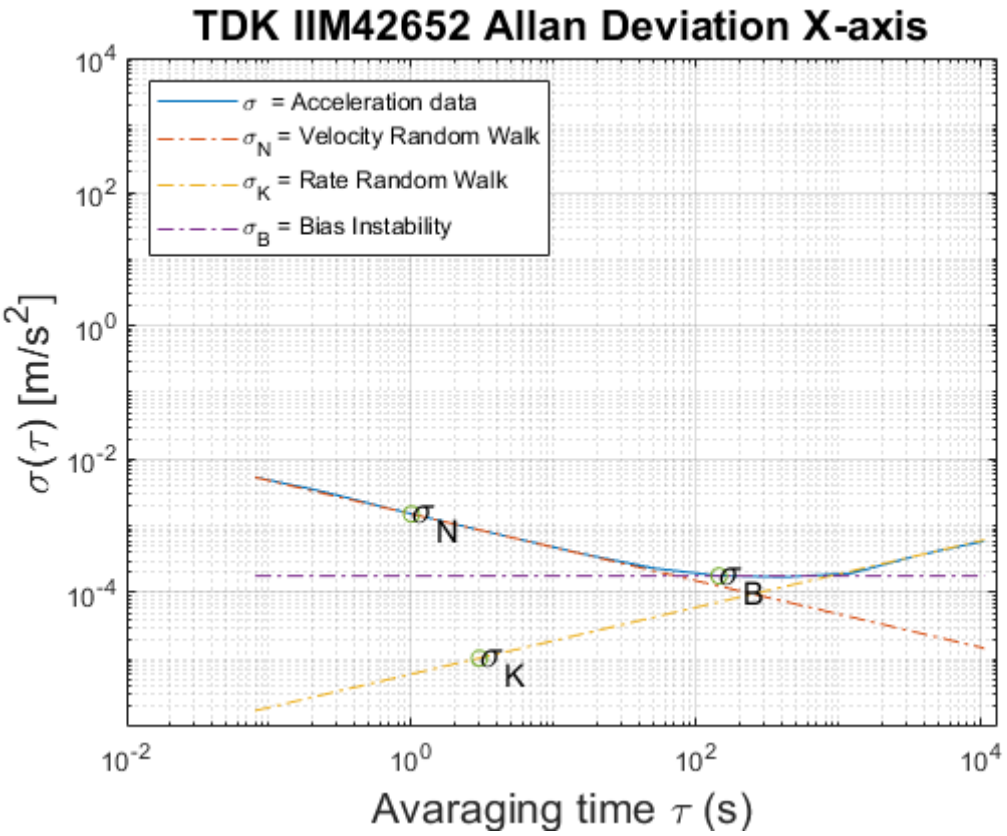
tauParams = 1×3
1.0000 3.0000 0.0480
params = 1×3
10^{-4} ×
0.1361 0.0029 0.4656
1. B = 7.0091e-05

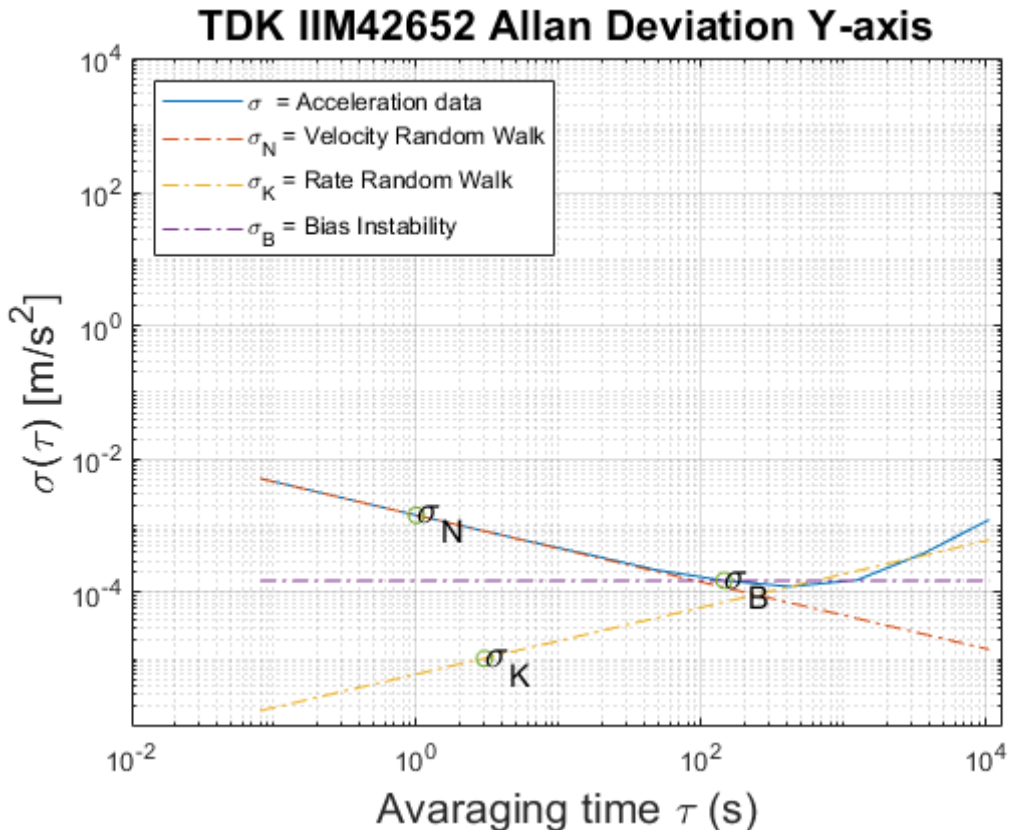
Observe the Bias has been stuck on a local minimum…

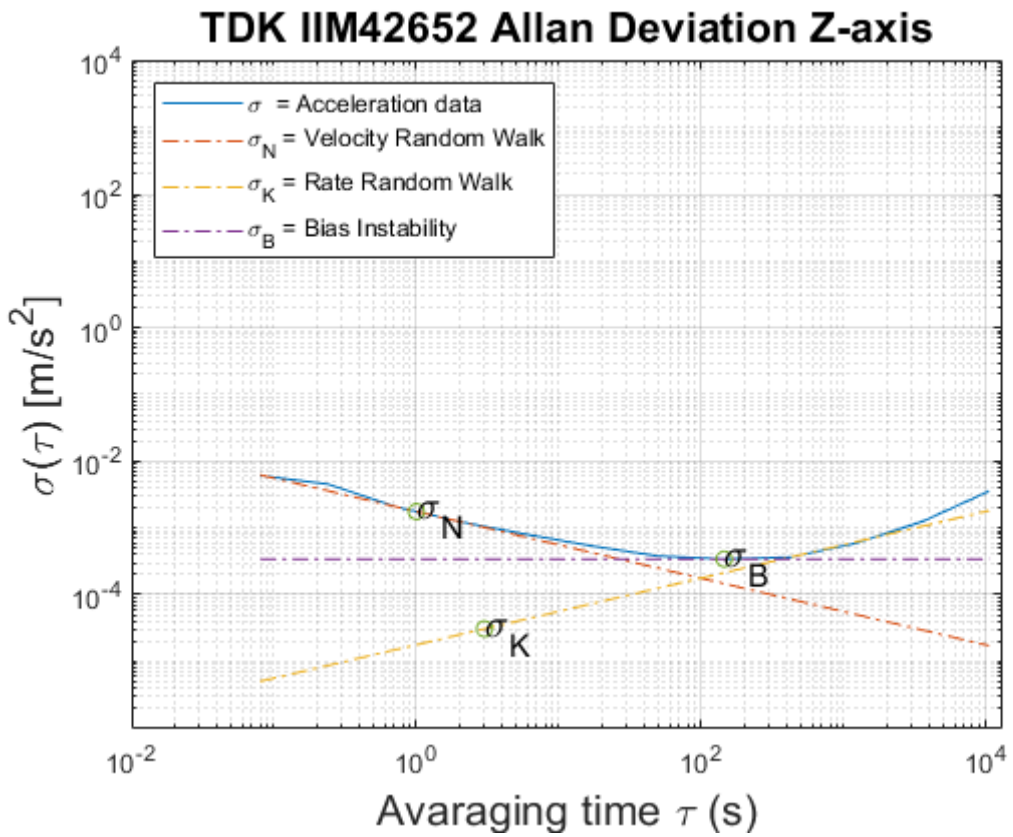
APPENDIX 3

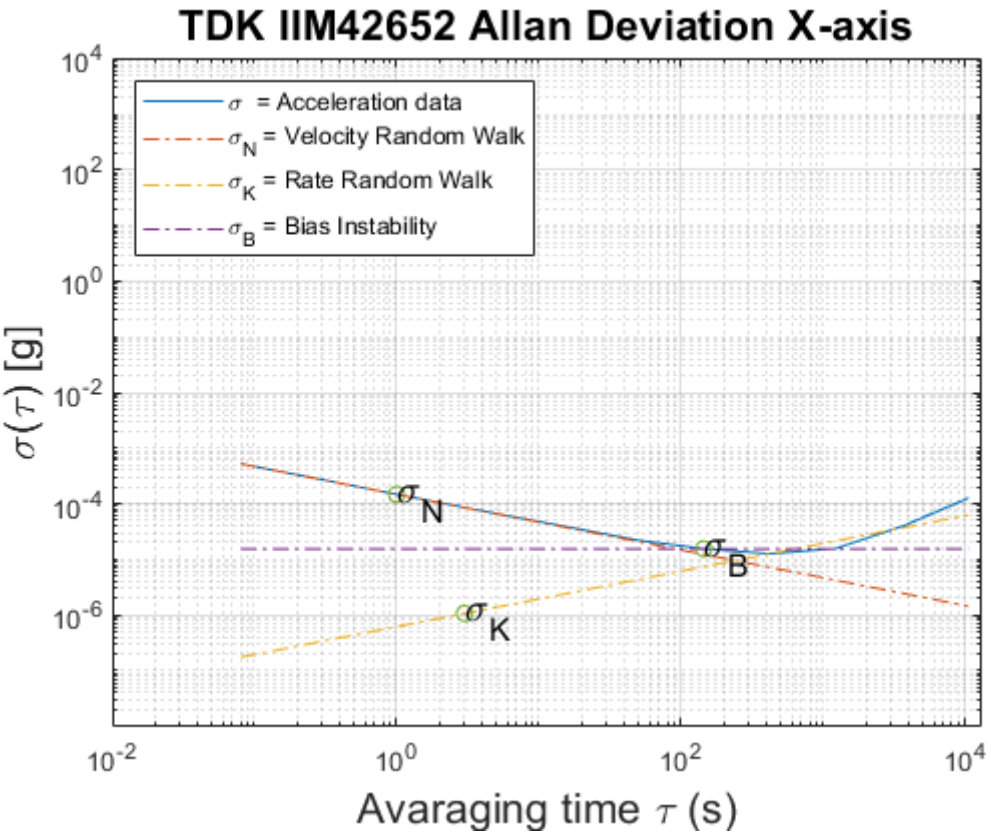
Allan deviation plots of TDK IIM42652 IMU

2. t0 = 0.0800 tdk sample speed

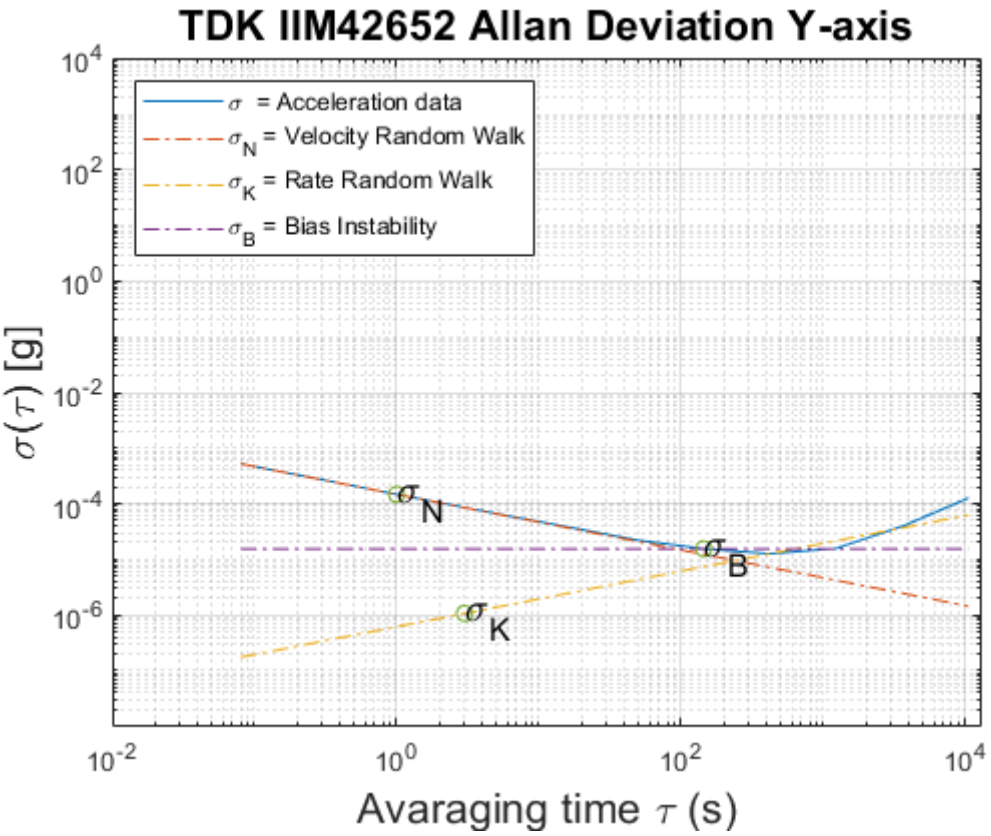



TDK IIM42652 Allan Deviation Y-axis
σ = Acceleration data
σN = Velocity Random Walk
σK = Rate Random Walk
σB = Bias Instability
σ(τ) [m/s2]
Avaraging time τ (s)

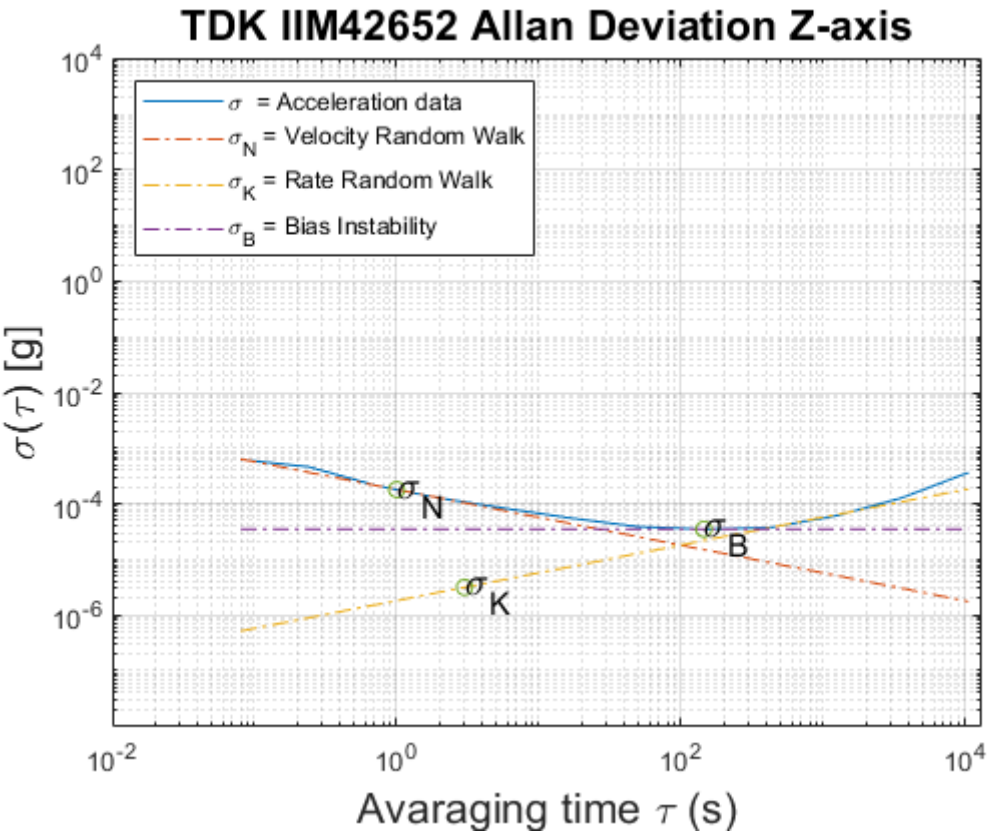

TDK IIM42652 Allan Deviation Z-axis
σ = Acceleration data
σN = Velocity Random Walk
σK = Rate Random Walk
σB = Bias Instability
σ(τ) [m/s2]
Avaraging time τ (s)



3. tauParams = 1×3
4. 1.0000 3.0000 144.4800
5. params = 1×3
6. 10^{-3} ×
7. 0.1470 0.0011 0.0155
8. B = 2.3272e-05
9. t0 = 0.0800

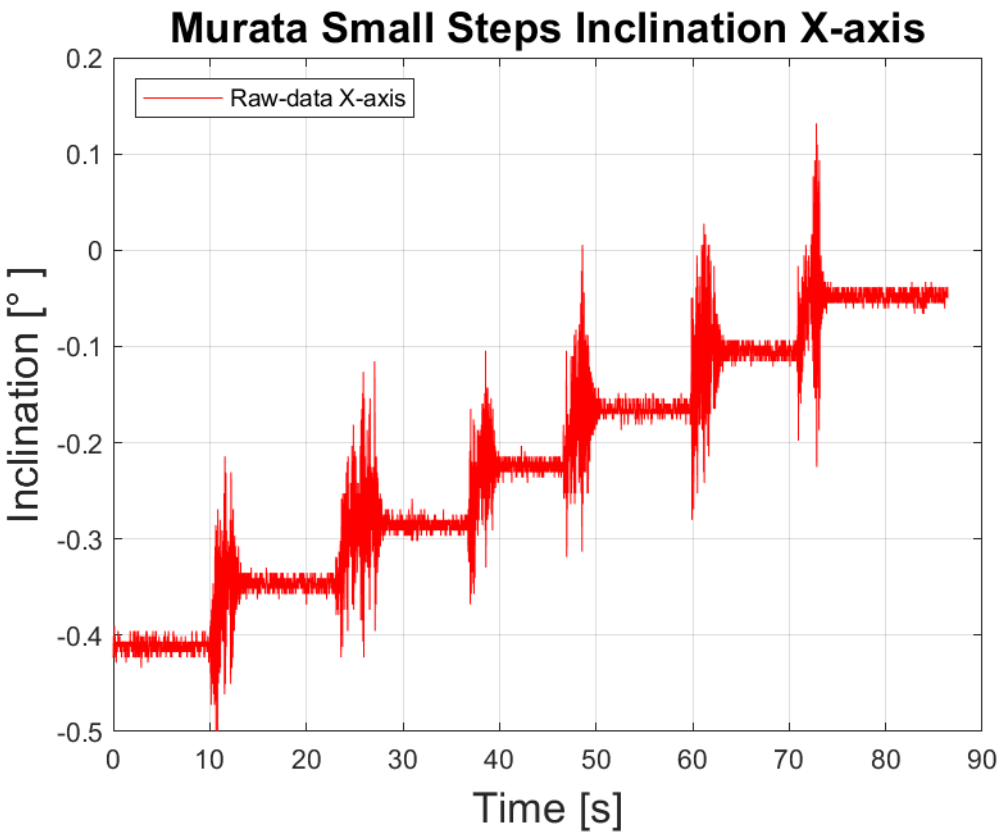


10. tauParams = 1×3
11. 1.0000 3.0000 144.4800
12. params = 1×3
13. 10^{-3} ×
14. 0.1470 0.0011 0.0155
15. B = 2.3272e-05
16. t0 = 0.0800

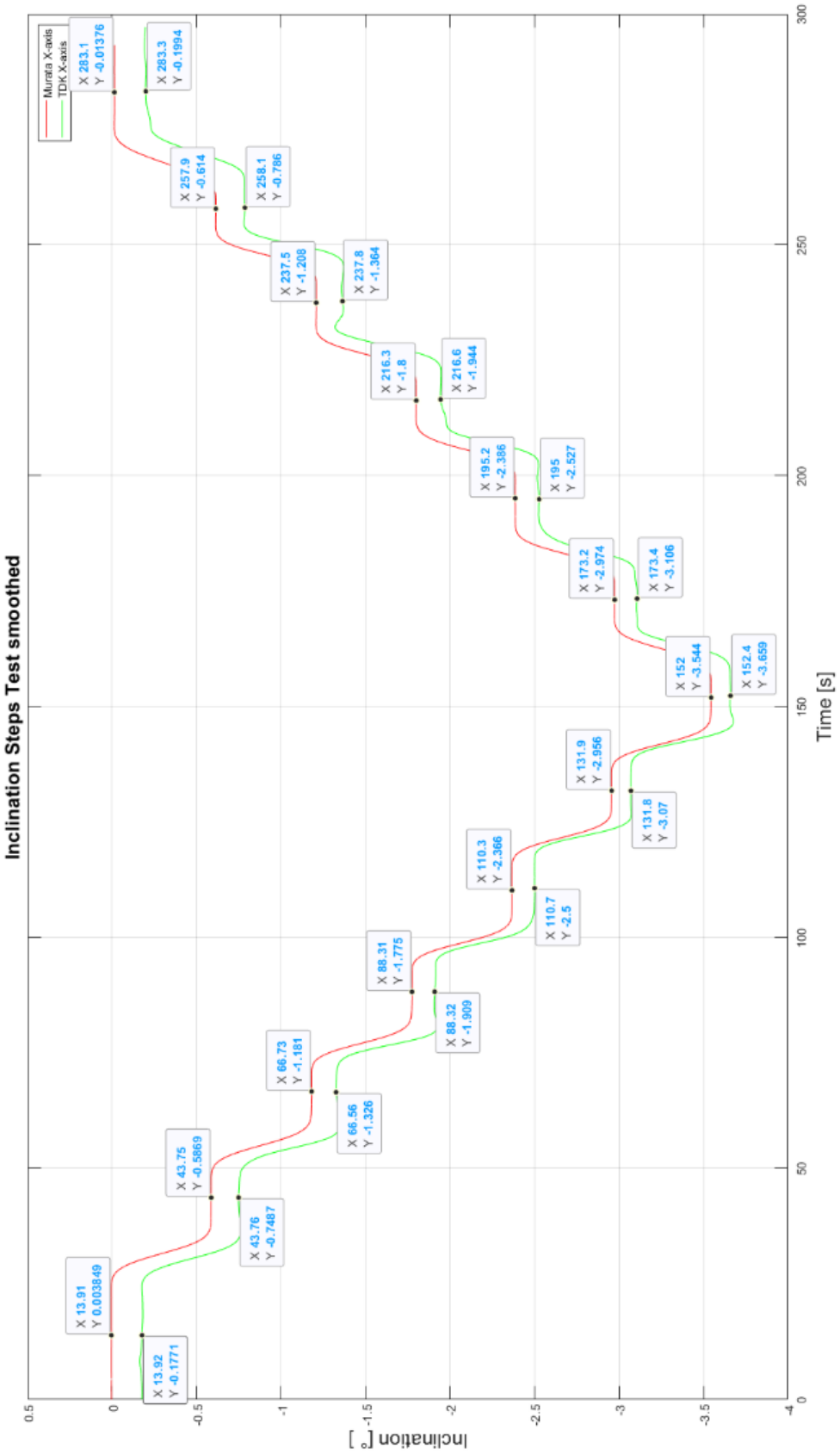


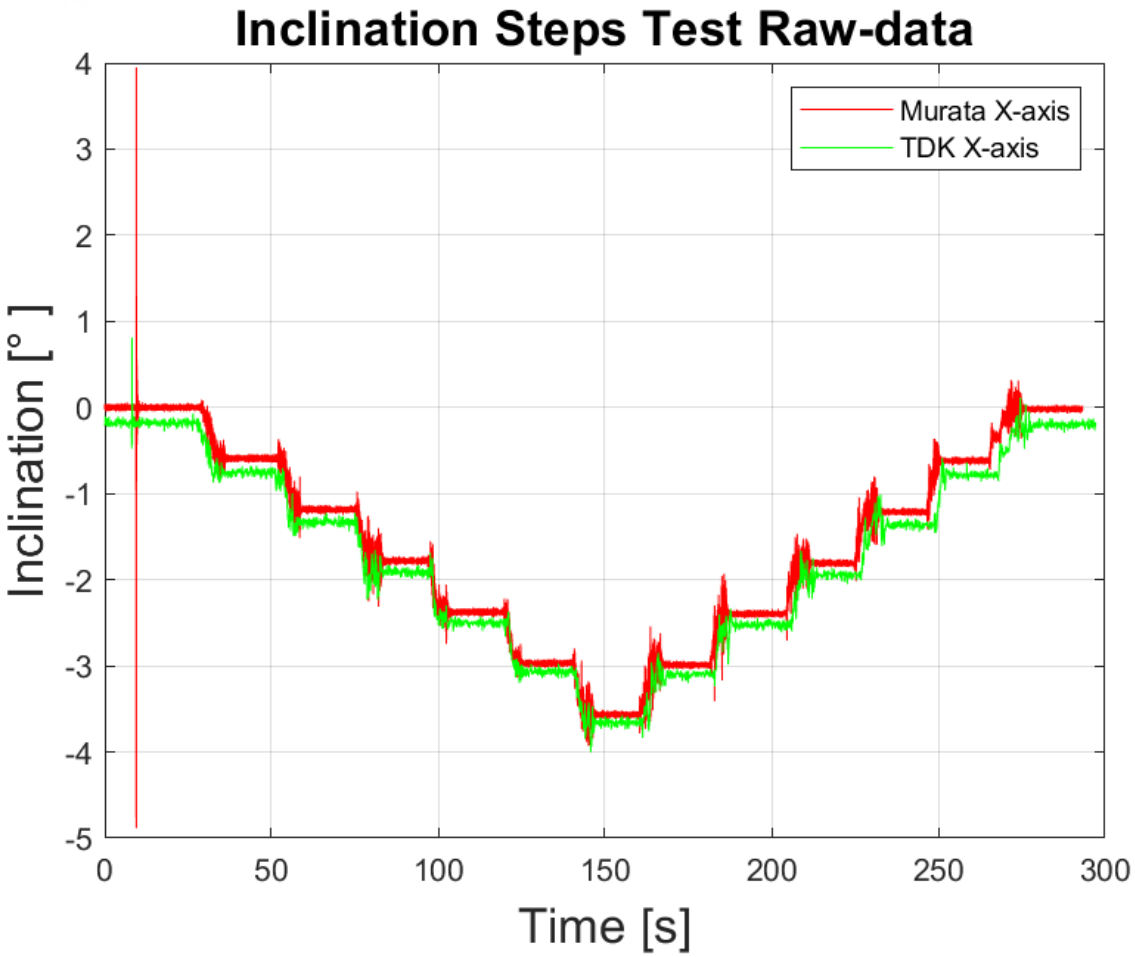
17. tauParams = 1×3
18. 1.0000 3.0000 144.4800
19. params = 1×3
20. 10^{-3} ×
21. 0.1792 0.0031 0.0348
22. B = 5.2414e-05

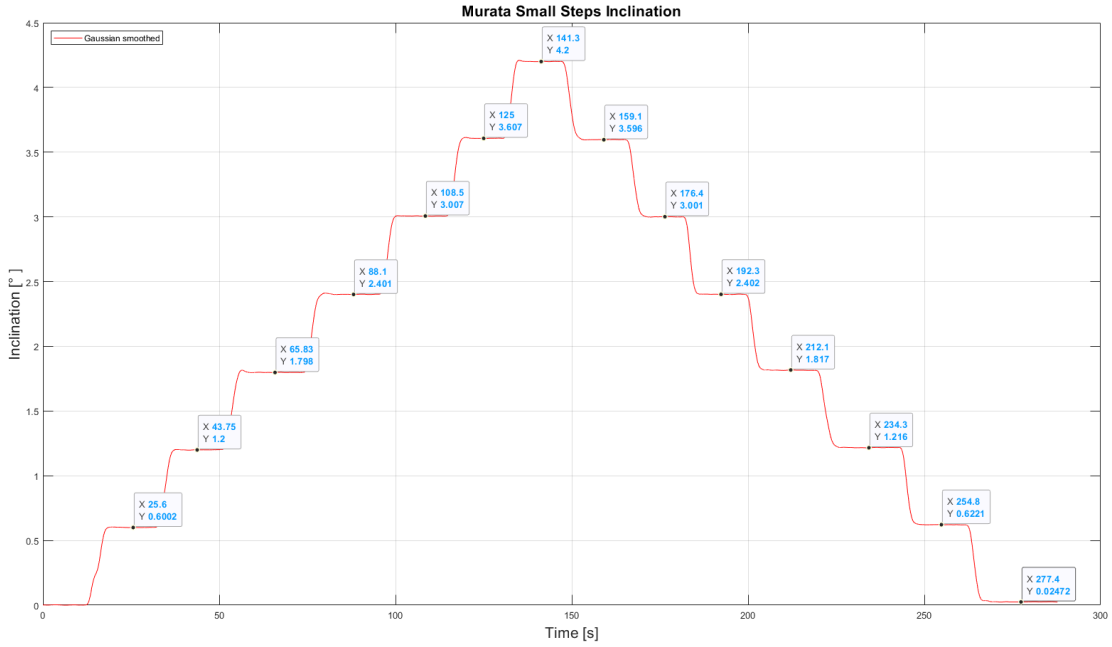
APPENDIX 4

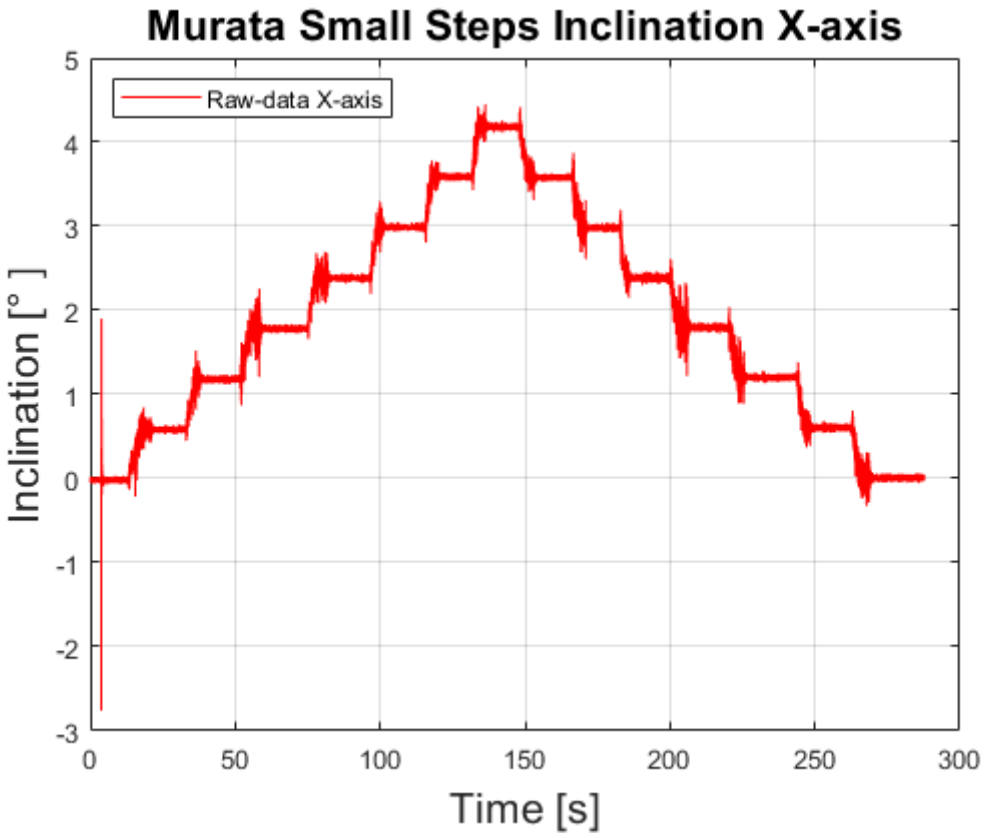

Murata Small Steps Inclination X-axis
Raw-data X-axis
Inclination [°]
Time [s]
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
10
20
30
40
50
60
70
80
90

APPENDIX 5


Inclination Steps Test smoothed
Murata X-axis
TDK X-axis
X 283.1
Y -0.01376
X 283.3
Y -0.1994
X 257.9
Y -0.614
X 258.1
Y -0.786
X 237.5
Y -1.208
X 237.8
Y -1.364
X 216.3
Y -1.8
X 216.6
Y -1.944
X 195.2
Y -2.386
X 195
Y -2.527
X 173.2
Y -2.974
X 173.4
Y -3.106
X 152
Y -3.544
X 152.4
Y -3.659
X 131.9
Y -2.956
X 131.8
Y -3.07
X 110.3
Y -2.366
X 110.7
Y -2.5
X 88.31
Y -1.775
X 88.32
Y -1.909
X 66.73
Y -1.181
X 66.56
Y -1.326
X 43.75
Y -0.5869
X 43.76
Y -0.7487
X 13.91
Y 0.003849
X 13.92
Y -0.1771
Inclination [°]
Time [s]


Inclination Steps Test Raw-data
Murata X-axis
TDK X-axis
Inclination [°]
Time [s]
4
3
2
1
0
-1
-2
-3
-4
-5
0
50
100
150
200
250
300


Murata Small Steps Inclination
Gaussian smoothed
Inclination [°]
Time [s]
X 25.6
Y 0.6002
X 43.75
Y 1.2
X 65.83
Y 1.798
X 88.1
Y 2.401
X 108.5
Y 3.007
X 125
Y 3.607
X 141.3
Y 4.2
X 159.1
Y 3.596
X 176.4
Y 3.001
X 192.3
Y 2.402
X 212.1
Y 1.817
X 234.3
Y 1.216
X 254.8
Y 0.6221
X 277.4
Y 0.02472



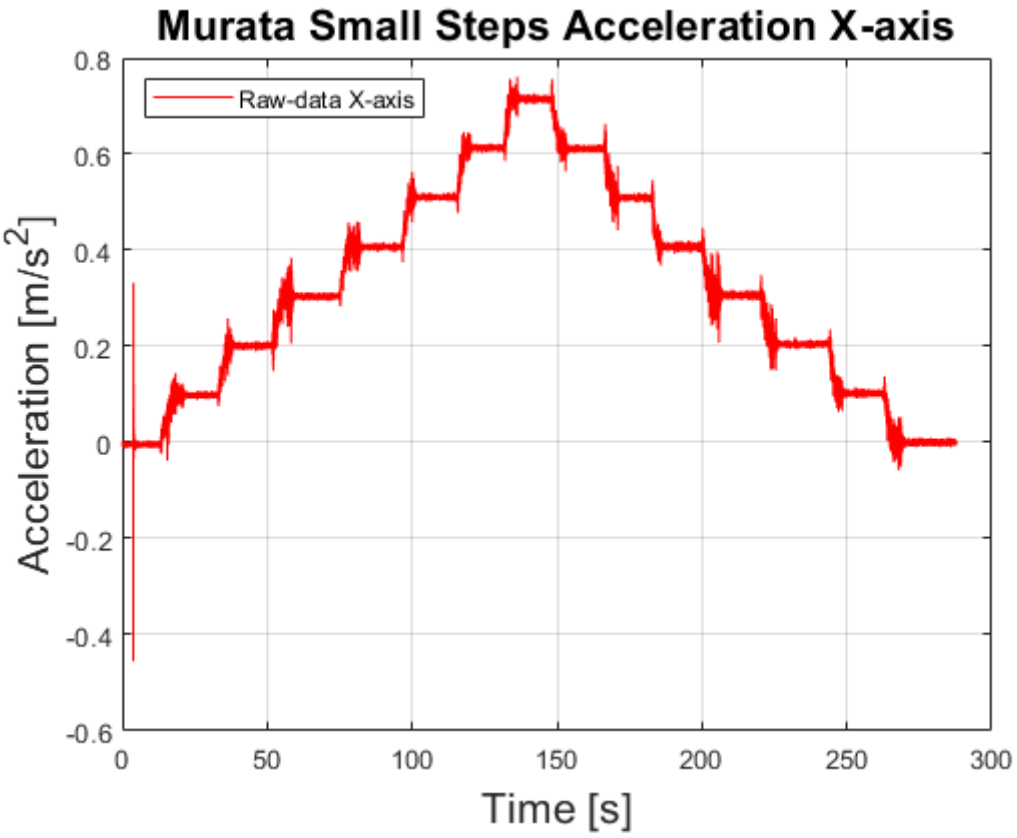
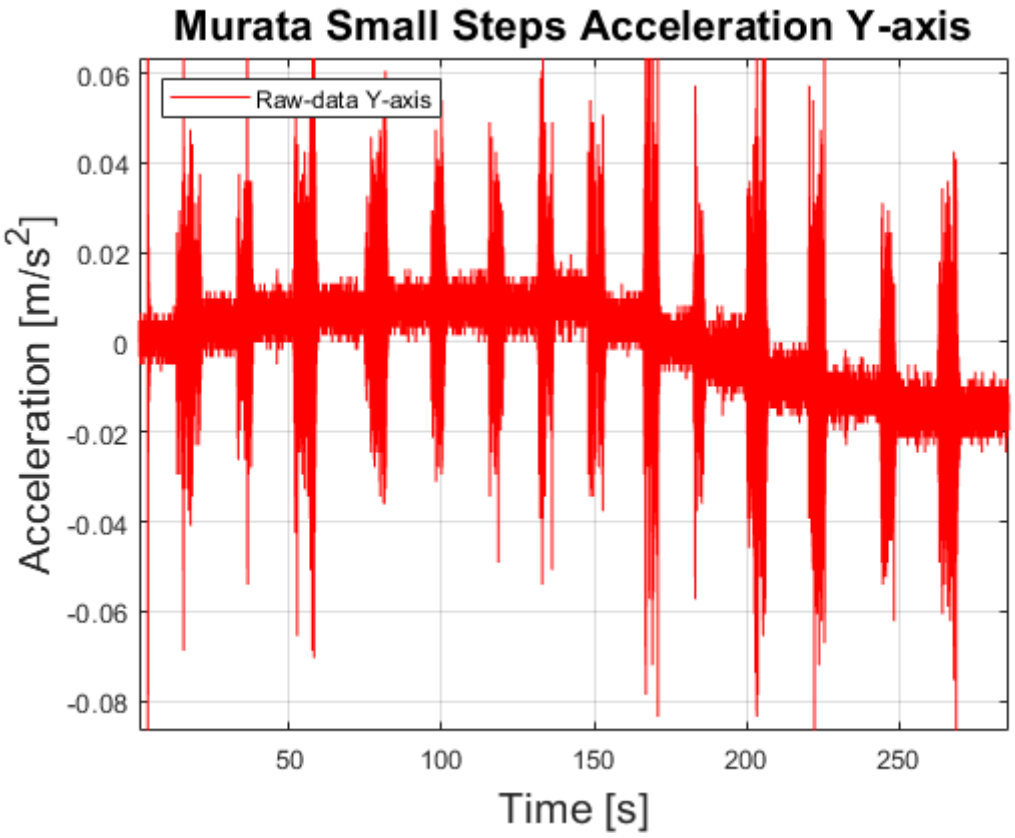
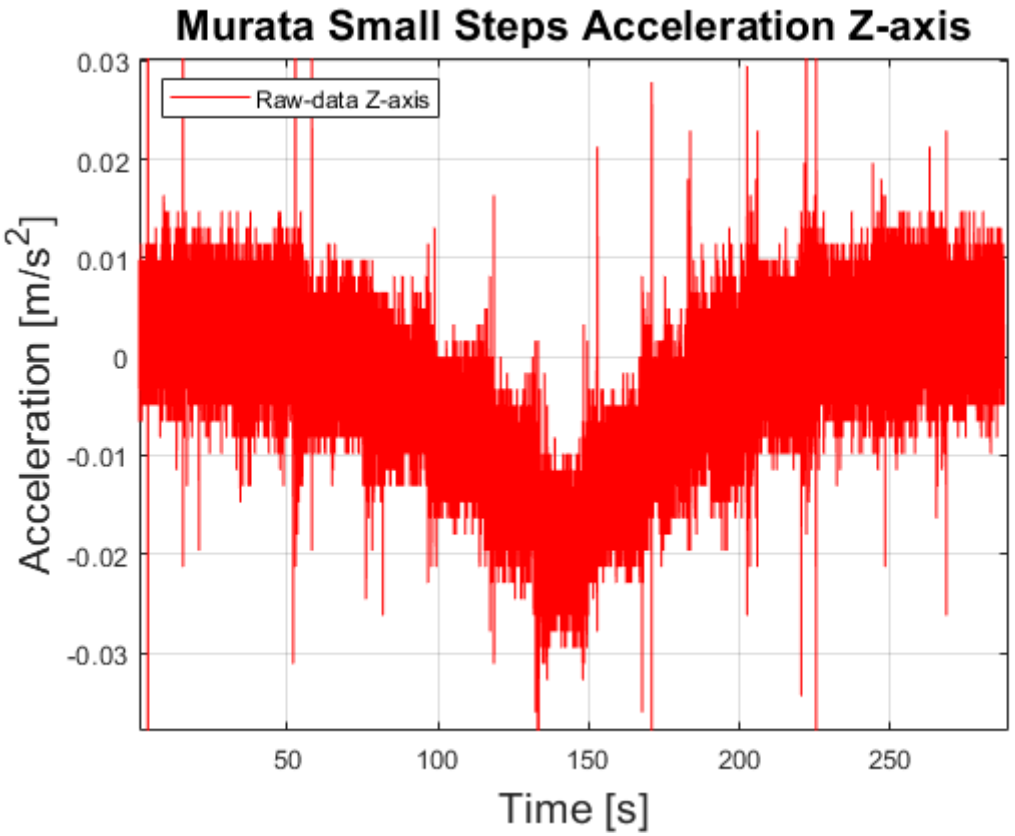
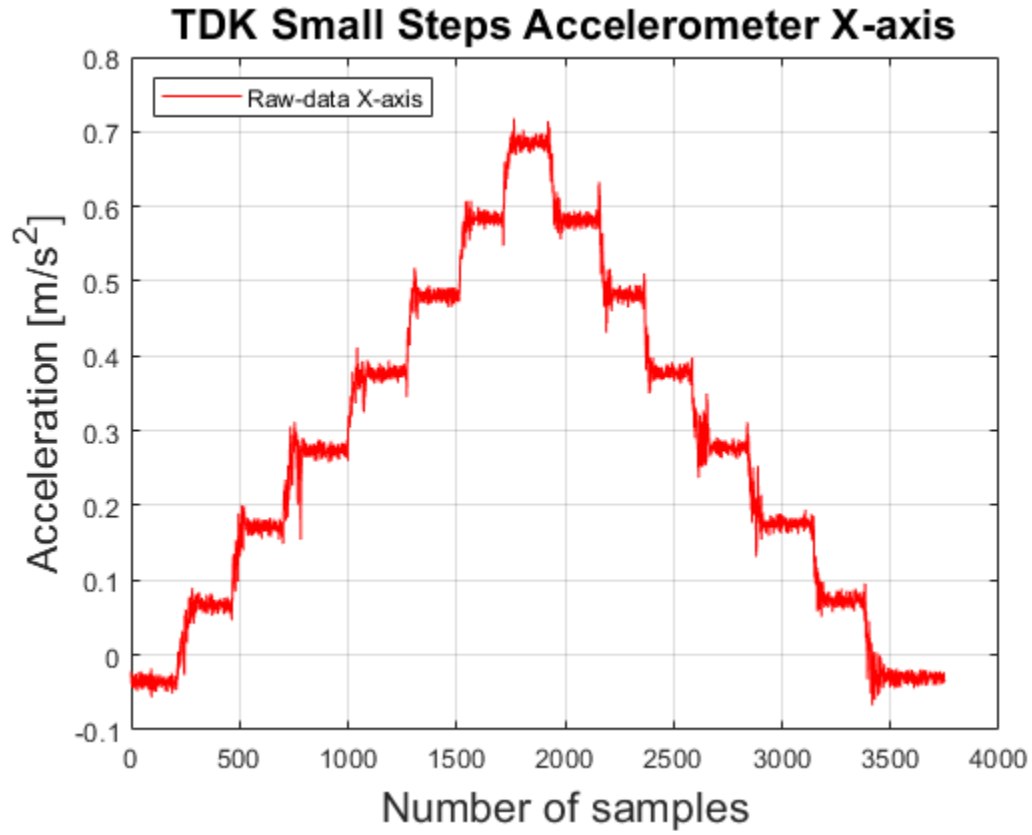


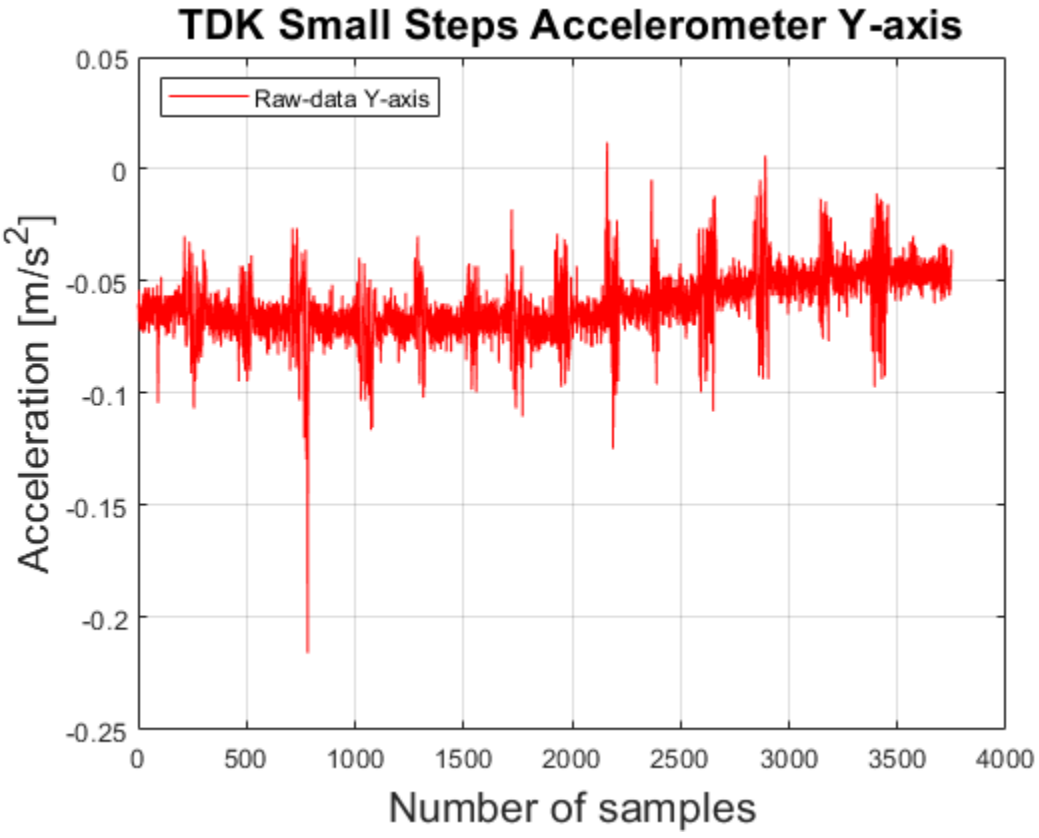
Figure 23:This is calculated by using the other angles

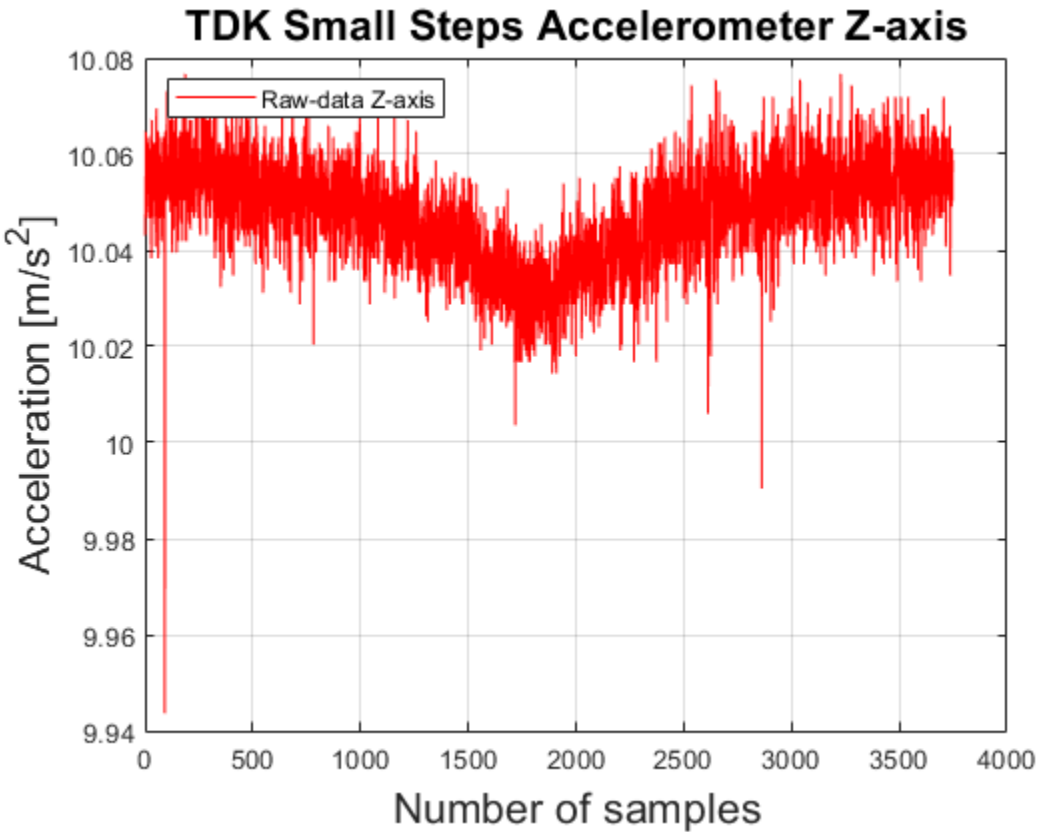

Murata Small Steps Acceleration Y-axis
Raw-data Y-axis
Acceleration [m/s^2]
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
50
100
150
200
250
Time [s]


Murata Small Steps Acceleration Z-axis
Raw-data Z-axis
Acceleration [m/s^2]
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
50
100
150
200
250
Time [s]

APPENDIX 6


TDK Small Steps Accelerometer X-axis
Raw-data X-axis
Acceleration [m/s^2]
Number of samples
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
500
1000
1500
2000
2500
3000
3500
4000

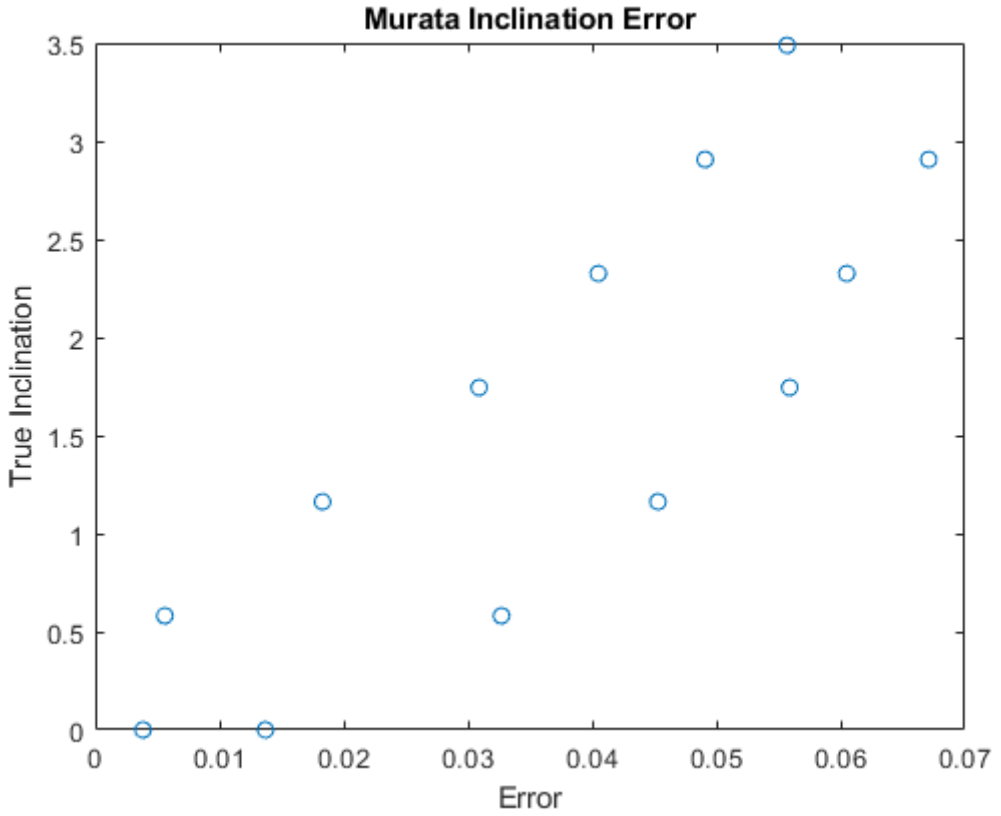

TDK Small Steps Accelerometer Y-axis
Raw-data Y-axis
Acceleration [m/s^2]
Number of samples
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
500
1000
1500
2000
2500
3000
3500
4000

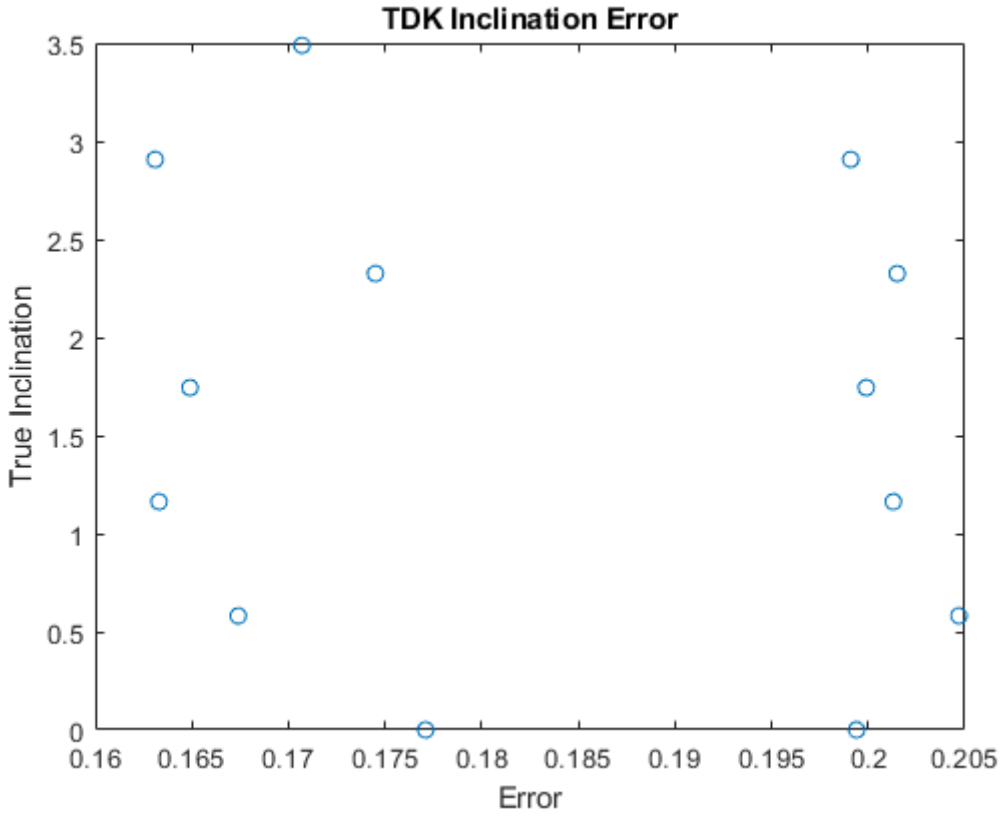


Inclination data
[283.280000000000,-0.199376272677576]
[283.066000000000,-0.0137637867288582]
[258.080000000000,-0.785978497600122]
[257.862000000000,-0.613953533809969]
[237.840000000000,-1.36372555742020]
[237.500000000000,-1.20792188777993]
[216.560000000000,-1.94400315823971]
[216.332000000000,-1.79978149730462]
[194.960000000000,-2.52675337009779]
[195.160000000000,-2.38551118001140]
[152.400000000000,-3.65882437721625]
[152.016000000000,-3.54425241405594]
[173.440000000000,-3.10646794529606]
[173.186000000000,-2.97371358340857]
[131.840000000000,-3.06979971898281]
[131.854000000000,-2.95573884015897]
[110.720000000000,-2.50035288320266]
[110.282000000000,-2.36624041734731]
[88.3200000000000,-1.90924477851761]
[88.3080000000000,-1.77500140293583]
[66.5600000000000,-1.32565307645398]
[66.7320000000000,-1.18100599551796]

[43.7600000000000,-0.748682277575355]
[43.7500000000000,-0.586945655700750]
[13.9200000000000,-0.177123829453360]
[13.9120000000000,0.00384886598028883]

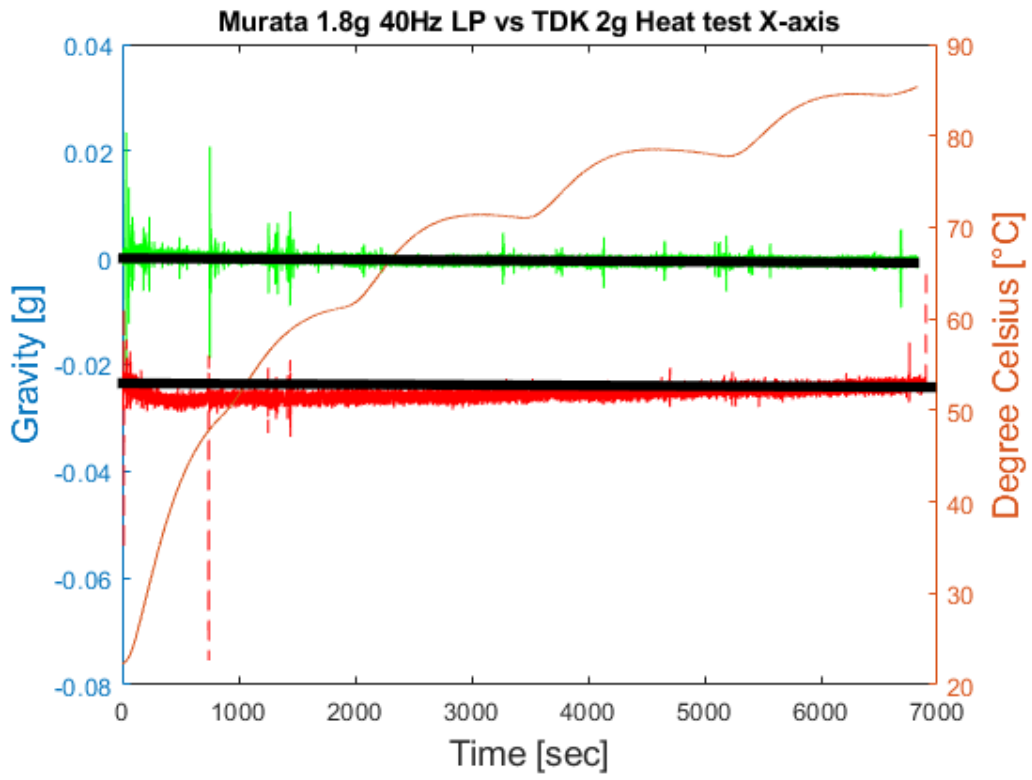
23. mean_error_murata = 0.0369
24. l2_murata = 0.0638
25. var_error_murata = 4.5214e-04
26. mean_error_tdk = 0.1836
27. var_error_tdk = 2.9754e-04
28. l1_tdk = 0.0517

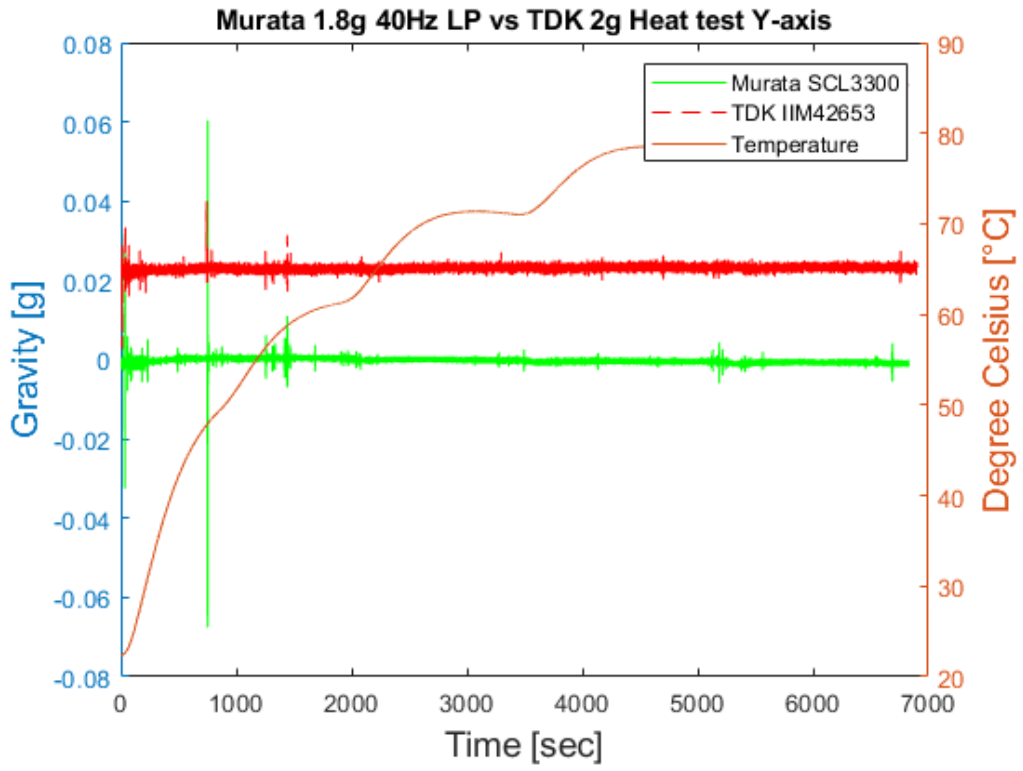

Murata Inclination Error
True Inclination
Error
0
0.5
1
1.5
2
2.5
3
3.5
0.01
0.02
0.03
0.04
0.05
0.06
0.07

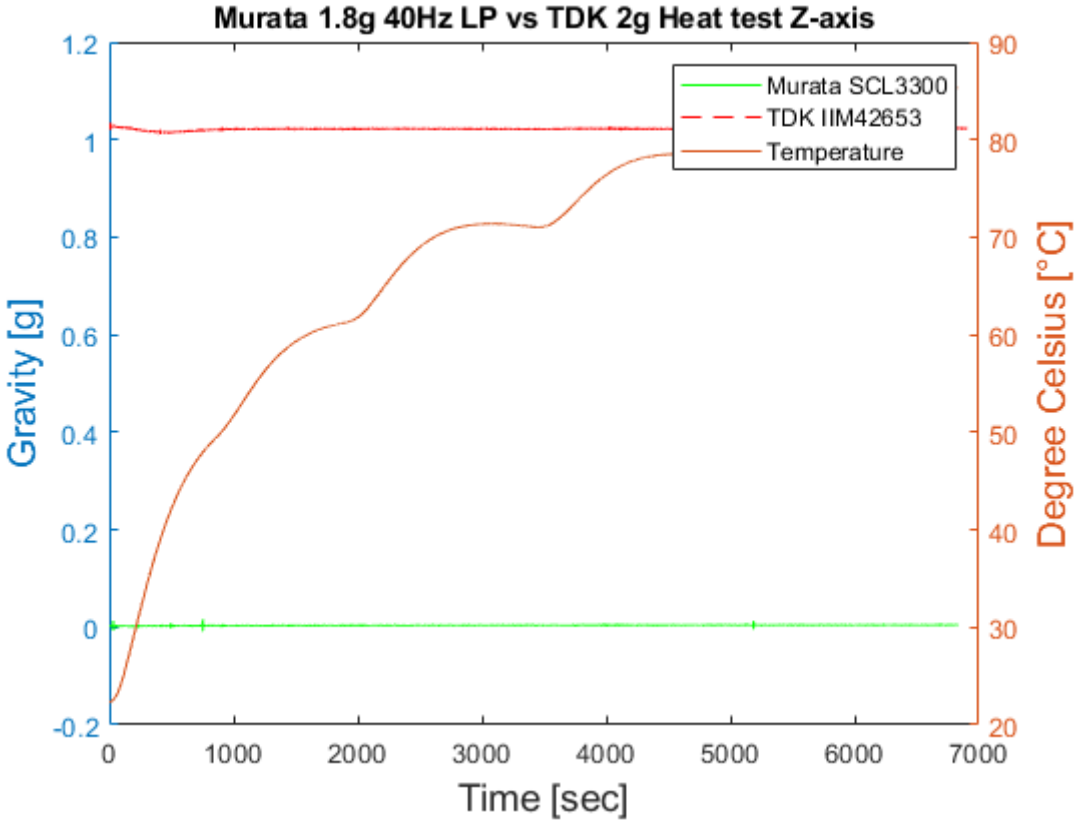

TDK Inclination Error
True Inclination
Error
0
0.5
1
1.5
2
2.5
3
3.5
0.16
0.165
0.17
0.175
0.18
0.185
0.19
0.195
0.2
0.205

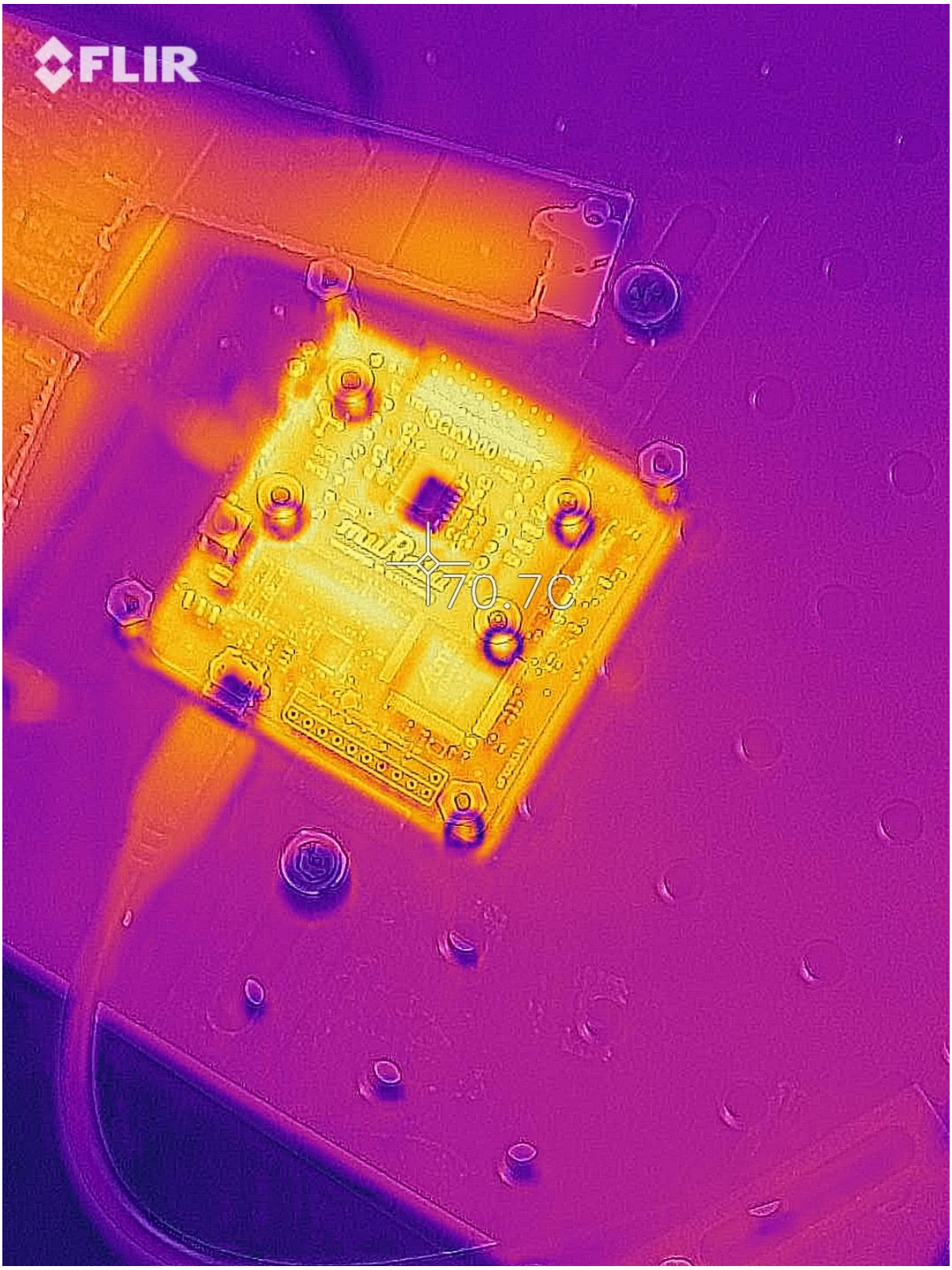
APPENDIX 7

This section includes plots, photos, and calculated data from heat test
Example of X-value data from both sensors
TDK
varis2 = 1×3
10^{-5} ×
    0.1145   0.0434   0.3315 Variance
me2 = 1×3
   -0.0254   0.0231   1.0220 Mean
l2 = 1×3
    0.0032   0.0020   0.0055 3sigma
Murata SCL33000
varis = 1×3
10^{-6} ×
    0.5422   0.2845   0.2770 Variance
me = 1×3
    0.0042   -0.0003   -0.0004 Mean
l = 1×3
    0.0022   0.0016   0.0016 3sigma




Murata 1.8g 40Hz LP vs TDK 2g Heat test Y-axis
Murata SCL3300
TDK IIM42653
Temperature
Gravity [g]
Degree Celsius [°C]
Time [sec]
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
90
80
70
60
50
40
30
20
0
1000
2000
3000
4000
5000
6000
7000


Murata 1.8g 40Hz LP vs TDK 2g Heat test Z-axis
Murata SCL3300
TDK IIM42653
Temperature
Gravity [g]
Degree Celsius [°C]
Time [sec]
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
90
80
70
60
50
40
30
20
0
1000
2000
3000
4000
5000
6000
7000


FLIR
70.7C

APPENDIX 8

Program used for collecting the data that the IMUs have stored on the user's PC

It can handle multiple files when longer sample series are done. The Murata MEMS Demo software will make new files when the files groves to big. To manually downloading 20 files will take forever and therefore following program was developed.

```
%%%% LONG TIME TEST JOELS APARTMENT %%%%%%%%

% STATIC TESTING %

%Prompt user to select the data, of type txt
%Path is the path of the files, files is the file names
clc;
clear all;

%Inint parameters
data_in = table;

% Is it static test? Same angle but many files
% 1 if static
% 2 if different measurements and angles
% 3 default -> prompt user
static_value = 3;


%%%%%%%%%%% Start of program %%%%%%%%%%%%%%%

%Prompt user to select the file to read from
%Returning noumber of files file/files names, path and
[num_of_files,string_of_files,pth,files] = number_of_files_function();

%Mean value of the angles
%Extract the data

[tdk,data,static_value]                                                    =
extraction_and_comput_mean_value(data_in,num_of_files,pth,files,static_value);

%[mean_value,data,static_value] Not used anymore

if(tdk == 1)
    data_tdk = data;
else

   data_murata = data;
    %Murata has proper timestamp from start
```

```
end
```

APPENDIX 9

Need to extract the data first before using the example below.

```
%Solving the offset value;
m6 = mean(angle_X_ROI_6);
m0 = mean(angle_X_ROI) -m6
m1 = mean(angle_X_ROI_1) -m6
m2 = mean(angle_X_ROI_2) -m6
m3 = mean(angle_X_ROI_3) -m6
m4 = mean(angle_X_ROI_4) -m6
m5 = mean(angle_X_ROI_5) -m6
m6 = m6 -m6
```

```
varis0 = var(angle_X_ROI)          %Variance
m0 =  mean(angle_X_ROI)          %Mean value
l0 = 3.*sqrt(varis0)              %Standard deviation
```

```
varis1 = var(angle_X_ROI_1)          %Variance
m1 =  mean(angle_X_ROI_1)          %Mean value
l1 = 3.*sqrt(varis1)              %Standard deviation
```

```
varis2 = var(angle_X_ROI_2)          %Variance
m2 =  mean(angle_X_ROI_2)          %Mean value
l2 = 3.*sqrt(varis2)              %Standard deviation
```

```
varis3 = var(angle_X_ROI_3)          %Variance
m3 =  mean(angle_X_ROI_3)          %Mean value
l3 = 3.*sqrt(varis3)              %Standard deviation
```

```
varis4 = var(angle_X_ROI_4)          %Variance
m4 =  mean(angle_X_ROI_4)          %Mean value
l4 = 3.*sqrt(varis4)              %Standard deviation
```

```
varis5 = var(angle_X_ROI_5)          %Variance
m5 =  mean(angle_X_ROI_5)          %Mean value
l5 = 3.*sqrt(varis5)               %Standard deviation
```

```
varis6 = var(angle_X_ROI_6)          %Variance
m6 =  mean(angle_X_ROI_6)          %Mean value
l6 = 3.*sqrt(varis6)               %Standard deviation
```

APPENDIX 10

```
%Example on Allan deviation



Mura = table2array(data_murata);
Mura = Mura(:,1:11);
Mura(:,3:5) = Mura(:,3:5);%.*9.819;


t0 = Mura(2,1);
Fs = 1/t0;
omega = Mura(:,5);  %X-axis


%Calculate the ensamen average of discreet data
theta = cumsum(omega, 1)*t0;
maxNumM = Fs;
L = size(theta, 1);
maxM = 2.^floor(log2(L/2));
                %The averaging intervals
m = logspace(log10(1), log10(maxM), maxNumM).'; %transpose
m = ceil(m);        % m must be an integer
m = unique(m);      % Remove duplicates of same value


h = L./m(:,:).';    %start on error calculation
Kerror = (1./(sqrt( 2.*(h - 1)))).'; %Error calculation
%It will be extreamly small with the data used in this project



%Matlab function computes the Allan variance
[avar, tau] = allanvar(omega, m, Fs);


%Square root of the variance gives the deviation
adev = sqrt(avar);



%Computing the Bias instability
slope = 0;
logtau = log10(tau);
logadev = log10(adev);
dlogadev = diff(logadev) ./ diff(logtau);
```

```
[~, i] = min(abs(dlogadev - slope));


%Find the y-intercept of the line
b = logadev(i) - slope*logtau(i);


%Determine the bias instability coefficient from the line
scfB = sqrt(2*log(2)/pi); %Approximation
logB = b - log10(scfB);
B = 10^logB;
tauB = tau(i);
lineB = B * scfB * ones(size(tau));


%Random walk slope causing noise for short integration times
slope = -0.5;
logtau = log10(tau);
logadev = log10(adev);
dlogadev = diff(logadev) ./ diff(logtau);
[~, i] = min(abs(dlogadev - slope));




%Find the interception of the line
b = logadev(i) - slope*logtau(i);
logN = slope*log(1) + b;
N = 10^logN;
tauN = 1;
lineN = N ./ sqrt(tau);


%Find rate/Acceleration random walk
%Will cause more uncertainties in the data if longer integration
%of data is done
slope = 0.5;
logtau = log10(tau);
logadev = log10(adev);
dlogadev = diff(logadev) ./ diff(logtau);
[~, i] = min(abs(dlogadev - slope));


%Find the interception of the y-line
b = logadev(i) - slope*logtau(i);
logK = slope*log10(3) + b;
K = 10^logK;
tauK = 3;
lineK = K .* sqrt(tau/3);
```

```
tauParams = [tauN, tauK, tauB];
params = [N, K, scfB*B];



figure
loglog(tau, adev, tau, [lineN, lineK, lineB],'-.', ...
   tauParams, params,'o')
%tau,(adev + adev.*Kerror),tau,(adev - adev.*Kerror)    %Messy with error
%function, will be better if the data is calculated to hr
title('Murata SCL3300 Allan Deviation X-axis','FontSize',16)
ylabel('\sigma(\tau) [g]','FontSize',16)
xlabel('Avaraging time \tau (s)','FontSize',16)
text(tauParams, params, {'\sigma_N', '\sigma_K', '\sigma_B'},'FontSize',15)
legend({['\sigma  = Acceleration data'],['\sigma_N = Velocity Random Walk'],['\sigma_K =
Rate Random Walk'],...
   ['\sigma_B = Bias Instability']},...
   'Location','northwest')
grid on
axis equal



tauParams = [tauN, tauK, tauB]  %Time where the noise terms was discovered
params = [N, K, scfB*B]         %Value of the noise terms
B
```

APPENDIX 11

A simplified version of plotting the data

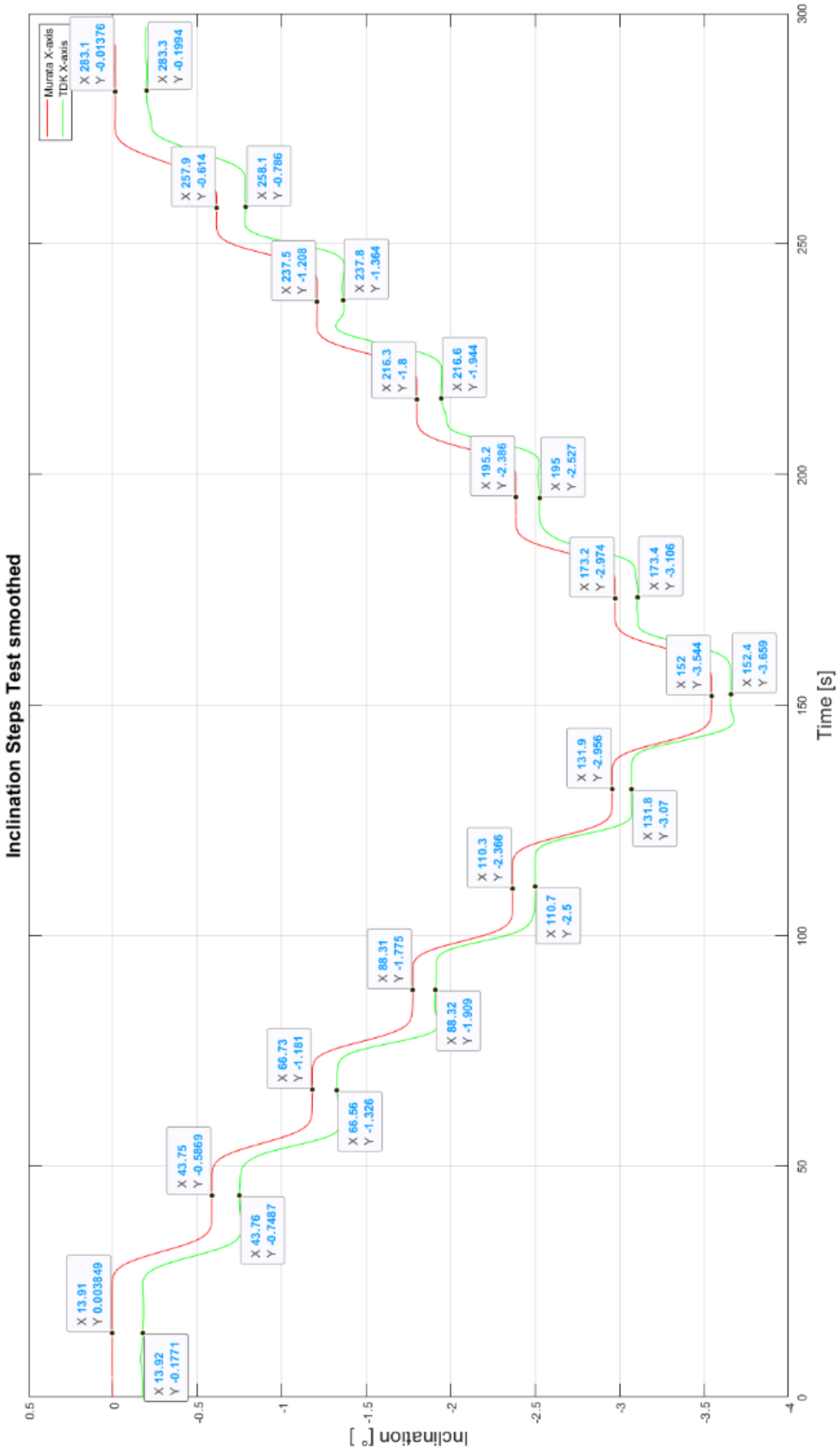
```
Mura = table2array(data_murata);
Mura = Mura(:,1:11);
Mura(:,3:5) = Mura(:,3:5).*9.819;


t0 = Mura(2,1);
Fs = 1/t0;
time = Mura(:,1);
accelData = Mura(:,3:8);


%Inclination
angle_Z = accelData(:,4);
angle_Y = accelData(:,5);
angle_X = accelData(:,6);


%Acceleration
acc_Z = accelData(:,1);
acc_Y = accelData(:,2);
acc_X = accelData(:,3);




%smoothedData2 = smoothdata(acc_X,'gaussian','SmoothingFactor',0.035);
%small = min(smoothedData2);
%figure
%smoothedData2 = smoothedData2 + abs(small);
smoothedData2 = smoothdata(angle_X,'gaussian','SmoothingFactor',0.008);
small = min(smoothedData2);


smoothedData2 = smoothedData2 + abs(small);




figure
plot(Mura(:,1),smoothedData2,'Color','r','DisplayName','Input data X')


%title('Murata Small Steps Acceleration','FontSize',16)
%ylabel('Acceleration [m/s^2]','FontSize',16)
%xlabel('Number of samples [N]','FontSize',16)
title('Murata Small Steps Inclination','FontSize',16)
ylabel('Inclination [° ]','FontSize',16)
```

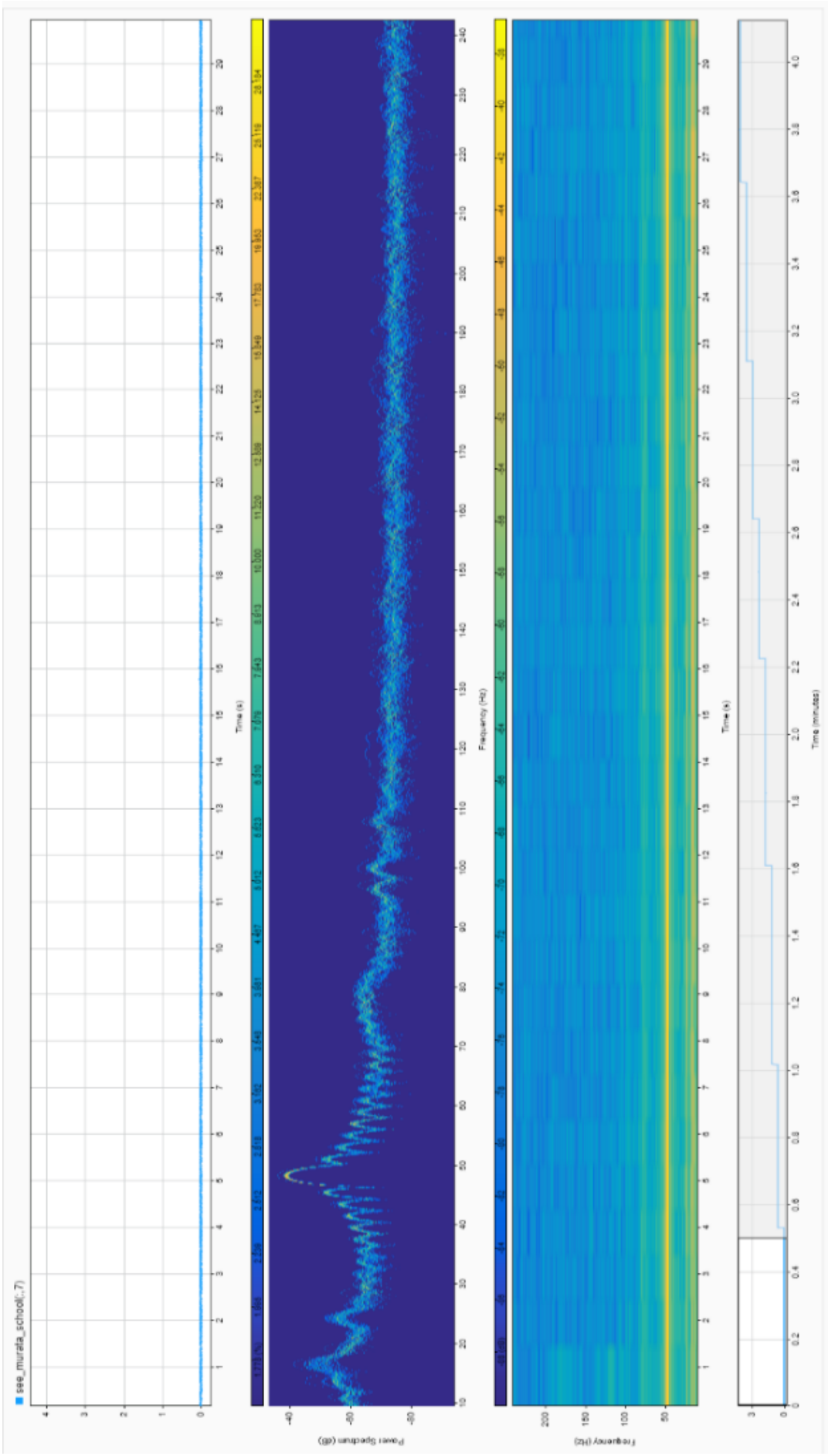
```
xlabel('Time [s]','FontSize',16)
legend('Gaussian smoothed','Location','northwest')
grid on


figure
plot(Mura(:,1),angle_X,'Color','r')
%title('Murata Small Steps Acceleration','FontSize',16)
%ylabel('Acceleration [m/s^2]','FontSize',16)
%xlabel('Number of samples [N]','FontSize',16)
title('Murata Small Steps Inclination X-axis','FontSize',16)
ylabel('Inclination [° ]','FontSize',16)
xlabel('Time [s]','FontSize',16)
legend('Raw-data X-axis','Location','northwest')
grid on


figure
plot(Mura(:,1),acc_X,'Color','r')
%title('Murata Small Steps Acceleration','FontSize',16)
%ylabel('Acceleration [m/s^2]','FontSize',16)
%xlabel('Number of samples [N]','FontSize',16)
title('Murata Small Steps Acceleration X-axis','FontSize',16)
ylabel('Acceleration [m/s^2]','FontSize',16)
xlabel('Time [s]','FontSize',16)
legend('Raw-data X-axis','Location','northwest')
grid on


figure
plot(Mura(:,1),acc_Y,'Color','r')
%title('Murata Small Steps Acceleration','FontSize',16)
%ylabel('Acceleration [m/s^2]','FontSize',16)
%xlabel('Number of samples [N]','FontSize',16)
title('Murata Small Steps Acceleration Y-axis','FontSize',16)
ylabel('Acceleration [m/s^2]','FontSize',16)
xlabel('Time [s]','FontSize',16)
legend('Raw-data Y-axis','Location','northwest')
grid on


figure
plot(Mura(:,1),acc_Z,'Color','r')
%title('Murata Small Steps Acceleration','FontSize',16)
%ylabel('Acceleration [m/s^2]','FontSize',16)
%xlabel('Number of samples [N]','FontSize',16)
title('Murata Small Steps Acceleration Z-axis','FontSize',16)
ylabel('Acceleration [m/s^2]','FontSize',16)
xlabel('Time [s]','FontSize',16)
legend('Raw-data Z-axis','Location','northwest')
```

```
grid on
```

Long time sample data

me_tdk = -0.4045
var_tdk = 9.9557e-04
sigma3_tdk = 0.0947
me_murata = -0.0197
var_mur = 1.1874e-04
sigma3_murata = 0.0327


Inclination Steps Test smoothed
Murata X-axis
TDK X-axis
X 283.1
Y -0.01376
X 283.3
Y -0.1994
X 257.9
Y -0.614
X 258.1
Y -0.786
X 237.5
Y -1.208
X 237.8
Y -1.364
X 216.3
Y -1.8
X 216.6
Y -1.944
X 195.2
Y -2.386
X 195
Y -2.527
X 173.2
Y -2.974
X 173.4
Y -3.106
X 152
Y -3.544
X 152.4
Y -3.659
X 131.9
Y -2.956
X 131.8
Y -3.07
X 110.3
Y -2.366
X 110.7
Y -2.5
X 88.31
Y -1.775
X 88.32
Y -1.909
X 66.73
Y -1.181
X 66.56
Y -1.326
X 43.75
Y -0.5869
X 43.76
Y -0.7487
X 13.91
Y 0.003849
X 13.92
Y -0.1771
Inclination [°]
Time [s]
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
0
50
100
150
200
250
300


see_murata_school(:,7)
Time (s)
Frequency (Hz)
Power Spectrum (dB)
Frequency (Hz)
Time (s)
Time (minutes)

CHALMERS
UNIVERSITY OF TECHNOLOGY