

Master Thesis

Universität Stuttgart
Institut für Navigation



Temperature dependency of a low-cost IMU

Yongxu Duan
(Matr.-Nr.3299533)

Supervisor: Msc Clemens Sonnleitner
Examiner: Prof. Dr. techn. Thomas Hobiger

February 2020

Declaration

Hereby, I ***Yongxu Duan*** confirm that the submitted Master thesis on the topic

Temperature dependency of a low-cost IMU

1. has been composed by myself fully independently,
2. that I have used no sources other than those indicated, and that I have appropriately declared all citations,
3. that the submitted Master thesis was neither completely nor in substantial parts subject of another examination,
4. that the work was neither completely nor in parts published before, and
5. that the electronic copy is consistent with the other copies.

place, date, signature

ACKNOWLEDGEMENTS

I would like to express my gratitude to those who helped me during writing this thesis and also those who helped me throughout my whole master study.

My deepest gratitude goes first to Prof. Dr. techn. Thomas Hobiger and Msc Clemens Sonnleitner. Thanks for providing me an opportunity to do master thesis at INS and their technical support, suggestion, encouragement and guidance. I really learned a lot from both of them, which helps me a lot in my research.

Secondly, I would like to express my gratitude to all the other professors and teachers for providing high quality courses. Thanks for their efforts to GEOENGINE program, I learned a lot and understand this major better.

Last but not least, my great thanks would go to my parents for their both mental and economical support. My parents' love is and will be always my greatest motivation. I will also say thanks to my lovely friends and classmates.

ABSTRACT

Nowadays, the Inertial Navigation System (INS) has great influence on many different positioning applications. With the development of Micro-Electro-Mechanical-Systems (MEMS) technology, low-cost inertial sensors can be now manufactured in small size and light weight. However, MEMS inertial sensors are still not performing as well as traditional sensors. Precisely characterizing the errors is therefore important.

Temperature is one important factor that influence the performance of MEMS IMU. In this case, this project examines the effect of changing the temperature by using a strapdown MEMS-based inertial sensor by analyzing the static observation data. In order to characterize the performance of the sensor, an analytical tool called Allan Variance is used to extract the noise properties of sensor at different temperature.

Keywords: MEMS, Inertial Measurement Unit, temperature

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Background	1
1.1 Objectives	2
1.2 Thesis Organization.....	3
2. INERTIAL NAVIGATION	4
2.1 Inertial Systems.....	4
2.1.1 Stable platform system	4
2.1.2 Strapdown system.....	5
2.2 Gyroscopes Principles of Operation.....	5
2.2.1 Mechanical gyroscope	5
2.2.2 Optical gyroscope	7
2.2.3 MEMS gyroscope	9
2.3 Accelerometer Principle of Operation.....	11
2.3.1 Mechanical accelerometer	11
2.3.2 Optical accelerometer	13
2.3.3 MEMS accelerometer	13
2.4 MEMS Sensors Noise Characteristics	15
2.4.1 Constant bias	15
2.4.2 Angle/Velocity random walk.....	15
2.4.3 Bias instability	15
2.4.4 Temperature effects	16
2.4.5 Calibration errors.....	16
3. ALLAN VARIANCE	17
3.1 Allan Variance Principles	17
3.2 Overlapping Allan Variance	17
3.3 Noise Identification.....	19
3.3.1 Angle/Velocity random walk.....	20
3.3.2 Bias instability	20
4. HARDWARE IMPLEMENTATION	22
4.1 Hardware Overview	22
4.1.1 MPU-9250 MEMS IMU.....	22
4.1.2 Single-board computer	24
4.2 Software Design	25
4.3 Experimental Setup	26
5. RESULTS.....	28
5.1 Raw Data	28
5.2 Allan Deviation Plots	29
5.3 Noise Analysis	34
5.3.1 Accelerometer noise analysis	35
5.3.2 Gyroscope noise analysis	37
5.4 Cooling Down Process analysis	39
5.5 Temperature Compensation Test.....	47

6. CONCLUSION.....	53
BIBLIOGRAPHY	54
APPENDIX A	56
APPENDIX B.....	57

LIST OF FIGURES

<i>Figure 1.1: Growing Market of MEMS Inertial Sensor</i>	2
<i>Figure 2.1: A stable platform IMU.....</i>	4
<i>Figure 2.2: Stable platform inertial navigation algorithm</i>	5
<i>Figure 2.3: Strapdown inertial navigation algorithm</i>	5
<i>Figure 2.4: Conventional two-axis gyroscope</i>	6
<i>Figure 2.5: Different types of mechanical gyroscopes.....</i>	6
<i>Figure 2.6: The Sagnac effect.....</i>	8
<i>Figure 2.7: Schematic diagram of a ring laser gyroscope.....</i>	8
<i>Figure 2.8: Generation of Coriolis force</i>	9
<i>Figure 2.9: Early prototype of MEMS gyro</i>	10
<i>Figure 2.10: BAE System's silicon vibrating structure gyroscope</i>	10
<i>Figure 2.11: Tuning fork MEMS gyroscope.....</i>	11
<i>Figure 2.12: A spring accelerometer</i>	12
<i>Figure 2.13: Pendulum accelerometer.....</i>	12
<i>Figure 2.14: Pendulous fiber optic accelerometer.....</i>	13
<i>Figure 2.15: In-plane pendulous accelerometer</i>	14
<i>Figure 2.16: Z-axis accelerometer</i>	14
<i>Figure 3.1: Comparison of non-overlapping and overlapping samples</i>	18
<i>Figure 3.2: A log-log plot of Allan Deviation possible noise items</i>	19
<i>Figure 3.3: $\sigma - \tau$ Plot for ARW/VRW</i>	20
<i>Figure 3.4: $\sigma - \tau$ Plot for bias instability (for $f_0 = 1$).....</i>	21
<i>Figure 4.1: MPU-9250 Block Diagram</i>	23
<i>Figure 4.2: SparkFun MPU-9250 PCB board</i>	24
<i>Figure 4.3: Raspberry Pi 3B+ board</i>	24
<i>Figure 4.4: Example of output data format.....</i>	26
<i>Figure 4.5: Experimental setup</i>	27
<i>Figure 5.1: An example of accelerometer raw data.....</i>	28
<i>Figure 5.2: An example of gyroscope raw data</i>	29
<i>Figure 5.3: ADEV plot at 34°C.....</i>	29
<i>Figure 5.4: ADEV plot at 43°C.....</i>	30
<i>Figure 5.5: ADEV plot at 62°C.....</i>	30
<i>Figure 5.6: ADEV plot at 82°C.....</i>	30
<i>Figure 5.7: Accelerometer ADEV plots at different temperature</i>	32
<i>Figure 5.8: Gyroscope ADEV plots at different temperature</i>	34
<i>Figure 5.9: Examples of ADEV with noise parameters.....</i>	35
<i>Figure 5.10: Accelerometer VRW values plot</i>	36
<i>Figure 5.11: Accelerometer Bias Instability</i>	37
<i>Figure 5.12: Gyroscope Angle Random Walk.....</i>	38
<i>Figure 5.13: Gyroscope Bias Instability</i>	39
<i>Figure 5.14: Cooling down process temperature change</i>	39
<i>Figure 5.15: Cooling down process raw data.....</i>	40
<i>Figure 5.16: Cooling down process Accelerometer X-axis.....</i>	41
<i>Figure 5.17: Cooling down process Accelerometer Y-axis.....</i>	41
<i>Figure 5.18: Cooling down process Accelerometer Z-axis.....</i>	42
<i>Figure 5.19: Cooling down process Gyroscope X-axis.....</i>	43
<i>Figure 5.20: Cooling down process Gyroscope Y-axis</i>	43
<i>Figure 5.21: Cooling down process Gyroscope Z-axis.....</i>	44

<i>Figure 5.22: Accelerometer ADEV against temperature</i>	46
<i>Figure 5.23: Gyroscope ADEV against temperature</i>	47
<i>Figure 5.24: Accelerometer polynomial fitting evaluation</i>	48
<i>Figure 5.25: Gyroscope polynomial fitting evaluation</i>	48
<i>Figure 5.26: Compensated and uncompensated data</i>	49
<i>Figure 5.27: Compensated Accelerometer ADEV against temperature</i>	50
<i>Figure 5.28: Compensated Gyroscope ADEV against temperature</i>	52

LIST OF TABLES

<i>Table 5.1: Accelerometer Velocity Random Walk values.....</i>	36
<i>Table 5.2: Accelerometer Bias Instability values.....</i>	36
<i>Table 5.3: Gyroscope Angle Radom Walk values</i>	37
<i>Table 5.4: Gyroscope Bias Instability values.....</i>	38

ABBREVIATION

INS	Inertial Navigation System
MEMS	Micro Electro Mechanical System
IMU	Inertial Measurement Unit
UAV	Unmanned Aerial Vehicle
FOG	Fiber Optic Gyroscope
RLG	Ring Laser Gyroscope
ARW	Angle Random Walk
VRW	Velocity Random Walk
PSD	Power Spectral Density
FFT	Fast Fourier Transform
AVAR	Allan Variance
ADEV	Allan Deviation
PTH	Plated Through Hole
SPI	Serial Peripheral Interface
I²C	Inter Integrated Circuit
FIFO	First In First Out

1. INTRODUCTION

1.1 Background

Navigation is a very ancient skill or art which has become a very complex science. It is essentially about travelling and finding the way from one place to another and there are a variety of means by which this may be achieved [1]. In the early stage of history, travelers accomplished navigation by using primitive charts and observations of the Sun and Polaris to determine directions. In the 13th century, compass was invented by Chinese and became one of the earliest human-made navigational tools. The ancient Chinese people navigated successfully across the South China Sea by means of compass. This device however was difficult to use in rough weather. Around 1730, an English mathematician, John Hadley (1682-1744), and an American inventor, Thomas Godfrey (1704-1749), independently invented the sextant, which provides mariners with a more accurate way of determining the angle between the horizon and the sun, moon, or stars to calculate latitude. People also realized the importance of maps. However, maps at that time were made by means of traveler's memory, which resulted in less accuracy. The maps at that time also had no accurate longitude and latitude. Longitude and latitude at that time could only be found relatively accurately by using celestial navigation. In 1884, by international agreement, the Prime Meridian (located at 0° longitude) was established as the meridian passing through Greenwich, England [2][3]. The maps were therefore more accurate. In the 20th century, more advanced navigational technologies were brought to the world. Global Navigation Satellite System (GNSS) was developed by different countries and still keeps updating now. Another very important method is Inertial Navigation System (INS), which is based on the Sir Isaac Newton's laws. The inertial navigation is operated by using gyroscopes and accelerometers to sense the rotational and translational motion with respect to an inertial reference frame, which are able to get the current speed and orientation. The present position can be then conducted from the knowledge of initial position and the knowledge of current speed and orientation.

Inertial Navigation Systems were originally designed for rockets. American rocketry pioneer Robert Goddard experimented with rudimentary gyroscopic systems. Dr. Goddard's systems were of great interest to contemporary German pioneers including Wernher von Braun. The systems entered more widespread use with the advent of spacecraft, guided missiles, and commercial airliners. Early German World War II V2 guidance systems combined two gyroscopes and a lateral accelerometer with a simple analog computer to adjust the azimuth for the rocket in flight [4]. In the early 1950s, a stable platform inertial navigation system was manufactured and aircraft first crossed the United States of America using full inertial navigation. In the 1960s, the inertial navigation system became standard equipment for military aircrafts, ships and submarines [5]. In the past decades, with the development of manufacturing process and micro-computers, the cost and size of inertial sensors were reduced significantly. Due to this, inertial navigation systems can also be applied in small projects. One of these technologies that is widely applied to reduce the price and size of inertial sensors is MEMS.

MEMS are tiny electro-mechanical devices made by some of the same methods as integrated circuits. MEMS were first proposed in the 1960s, but not commercialized until the 1980s. Engineers and scientists wanted to use the integrated circuit fabrication techniques to make tiny mechanical systems, which could be connected to electronic circuits on the same

chip [6]. MEMS based IMUs nowadays are ensembled in smart phones, UAVs, robots, autonomous driving cars and many other devices. Recently, the concept of IoT (Internet of Things) is becoming more and more popular and widely accepted. In the sensor level, the small size, low cost and low power consumption of MEMS devices make them an ideal field for IoT hardware innovation. Moreover, the performance of MEMS can be achieved is approaching inertial grade [5]. All in all, MEMS based inertial navigation system will become more important and have a bright future and the market is keep growing as shown in Figure 1.1.



Figure 1.1: Growing Market of MEMS Inertial Sensor (From:conventor.com)

1.1 Objectives

Developments in MEMS based IMUs have resulted in significant advancements in inertial navigation systems. MEMS IMUs have the advantages of lower power consumption, smaller size and lower weight. However, in practice there are many challenges in obtaining a precise navigation solution with MEMS devices. IMUs always have both systematic errors and stochastic errors as every sensor, regardless of technology or manufacturing quality. Low-cost MEMS sensors in particular exhibit larger bias drifts and more noise comparing with the traditional mechanical sensors. Due to this, characterization of stochastic errors becomes very important. As the temperature is one important factor that may influences the performance of MEMS based IMU, this master project aims to find the correlation between temperature and the performance of low-cost MEMS IMUs.

The effect of temperature on inertial sensors are different from device to device, so this thesis only evaluates MPU-9250, which is a very popular low-cost MEMS IMU in the market. In order to evaluate MPU-9250, this master thesis makes use of a very common frequency stability evaluation method, Allan Variance, which was proposed by David Allan in 1966. By using Allan Variance, we're able to characterize main noise terms at different temperature and may find a relationship between temperature and noise terms.

1.2 Thesis Organization

As mentioned above, this master thesis will mainly focus on the influence of temperature on the noise terms of MPU-9250. Apart from the final data analysis, this thesis will provide a brief introduction to inertial navigation systems and sensors in the beginning. Allan Variance will then be brought into discussion. And more importantly, experiment setup and final data analysis will be described and discussed. Hence, this master thesis is structured as follows:

Chapter 2 introduces two inertial navigation systems, an overview of gyroscopes and accelerometers, main noise and error terms of MEMS sensors.

Chapter 3 introduces history and principle of Allan Variance and how the Allan Variance characterize the noise terms for MEMS inertial sensors.

Chapter 4 introduces an overview of hardware applied in the experiment, software design for data collection and the experimental setup.

Chapter 5 introduces the analysis of data and discussion of results.

Chapter 6 introduces the conclusion and proposes some future works to be done.

2. INERTIAL NAVIGATION

Inertial navigation is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of an object relative to a known starting point, orientation and velocity. IMUs basically integrate three orthogonal axes accelerometer and three orthogonal axes gyroscopes, measuring linear acceleration and angular velocity. By some computational processing, the position and orientation can be obtained. Additionally, some IMUs also integrates three orthogonal axes magnetometers, measuring the strength and direction of the local magnetic field, in order to find the north direction. This section will only introduce the principles of different types of accelerometers and gyroscopes and especially MEMS technology.

2.1 Inertial Systems

There are basically two categories of inertial systems for nearly all IMUs, which are stable platform system and strapdown system. The difference in between is the frame of reference in which accelerometers and gyroscopes operate.

2.1.1 Stable platform system

Stable platform system means that inertial sensors are mounted on a platform that is isolated from any other external motion. The stable platform is aligned with the global frame, which is achieved by mounting the stable platform using gimbals which allows the platform freedom in all three axes, as shown in Figure 2.1. The accelerometers detect platform translations and the gyroscopes detect platform rotations. These signals are fed back to a torque motor that rotates the gimbal to counteract this rotation, hence keeping the platform consistent with the global frame. In order to track the direction of the device, the algorithm for the stable platform system is shown in Figure 2.2. The angles between adjacent gimbals can be read by angle pick-offs. The accelerometer signals also need to subtract the effect of gravity from the vertical channel and then perform two times of integration.

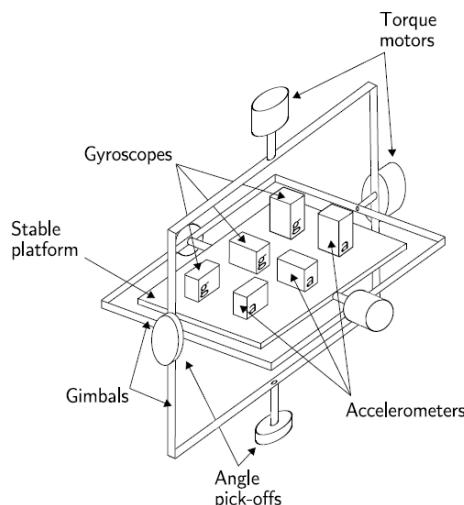


Figure 2.1: A stable platform IMU [7]

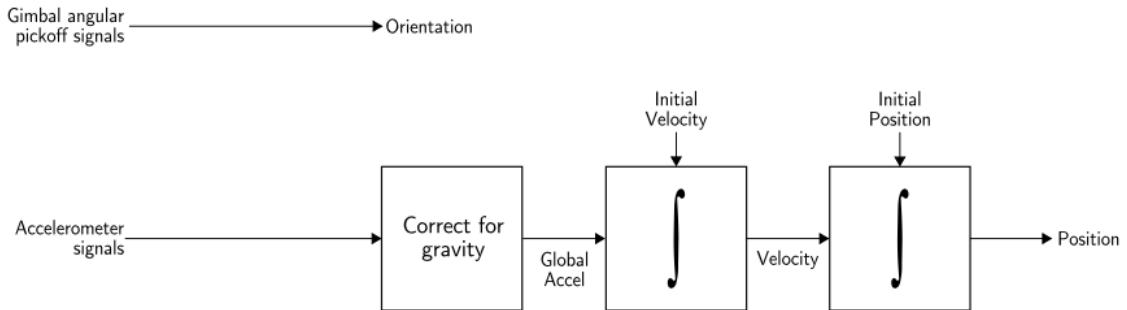


Figure 2.2: Stable platform inertial navigation algorithm [7]

2.1.2 Strapdown system

In strapdown systems, the inertial sensors are mounted rigidly onto the device. The outputs of the sensors are therefore measured in the body frame. In order to track the orientation, the rate-gyroscope signals are integrated firstly. After knowing the orientation, a projection of acceleration from body frame to global frame can be performed. The later integration processes are the same for stable platform systems. Strapdown platform systems share the same underlying principles as stable platform systems. Strapdown platform systems have less complexity and tend to be physically smaller than stable platform systems. But comparing with the algorithm of stable platform system, the strapdown platform inertial navigation algorithm (as shown in Figure 2.3) is more complex. However, with the development of micro-computers, the cost of computation has decreased significantly, hence the strapdown platform inertial navigation systems have become the mainstream of low-cost inertial navigation system nowadays.

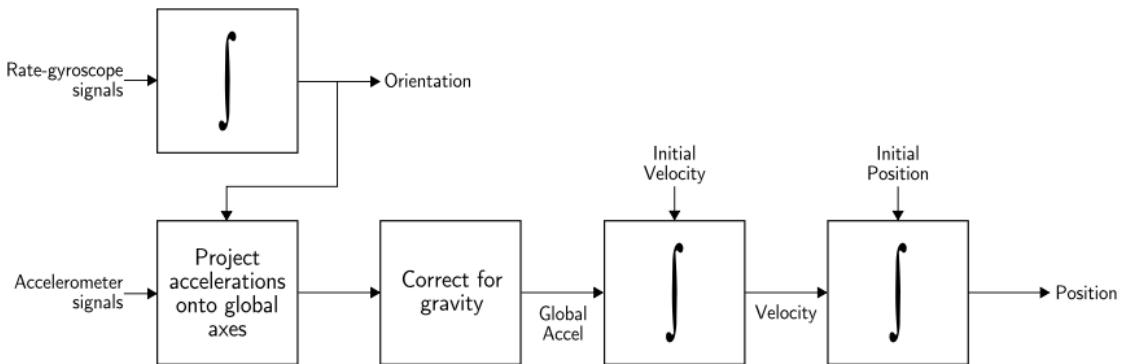


Figure 2.3: Strapdown inertial navigation algorithm [7]

2.2 Gyroscopes Principles of Operation

There are different types of gyroscopes that are used for different applications. This section will give an overview of gyroscopes and their principles of operation.

2.2.1 Mechanical gyroscope

One of the most basic and conventional form of mechanical gyroscopes is shown in Figure 2.4. It is designed to have a rotating wheel mounted on two gimbals, which allow the wheel to rotate in all three axes. The rotating wheel will resist changes in orientation due to the effect of maintaining angular momentum. Therefore, when the mechanical gyroscope is rotated, the

wheel will remain at a constant global orientation and the angles between adjacent gimbals will change. The angles can be measured by pick-off devices, which are mounted on gimbals. The orientation of the device thereby can be measured.

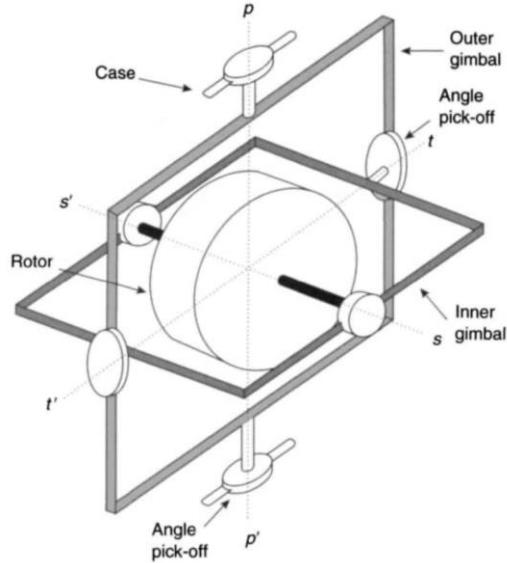


Figure 2.4: Conventional two-axis gyroscope [7]

There are many types of modern mechanical gyroscopes in the market, such as rate integrating gyroscopes, flex gyroscopes, rate gyroscopes, vibratory gyroscopes, tuning fork gyroscopes, quartz gyroscopes, nuclear magnetic resonance (NMR) gyroscopes, electrostatic gyroscopes (ESGs), etc. Some of these gyroscopes are shown as Figure 2.5. More detailed descriptions of these mechanical gyroscopes can be found in the literature [5].

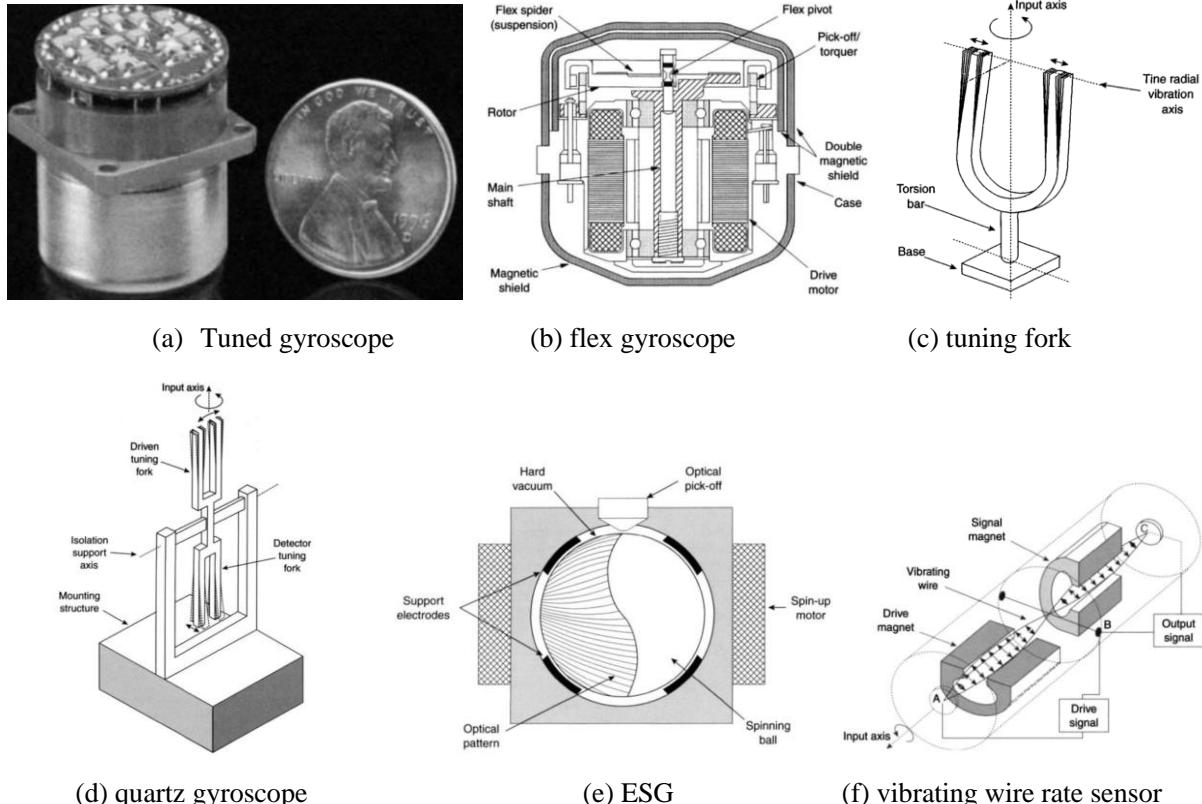


Figure 2.5: Different types of mechanical gyroscopes [5]

The main disadvantage of mechanical gyroscopes is that they contain moving parts. Moving parts create friction, which in turn causes the output to drift over time. To minimize friction, high-precision bearings and special lubricants are used, but it will increase equipment costs at the same time. Moreover, mechanical gyroscopes take a few minutes to warm up, which makes it not ideal in practice.

2.2.2 Optical gyroscope

Optical gyroscope refers to gyroscopes that use electromagnetic radiation characteristics to sense rotation. These gyroscopes mostly make use of visible wavelengths, but also some are operated in the near infrared. Notice that some mechanical gyroscopes which use optical angle pick-off sensors are usually not treated as optical gyroscopes.

Typically, all types of optical gyroscopes are suitable for different strapdown systems. The performance of optical gyroscopes ranges from very precise deviations (typically less than 0.001 ° / h, usually ring lasers gyroscopes) to tens of degrees per hour (usually simple fiber optic gyroscopes). Hence, the performance range of optical gyroscopes can cover the range of mechanical gyroscopes or relatively better. When comparing to mechanical gyroscopes, it appears that applying optical technology to sense angular rate has many advantages. The advantages of optical gyroscopes are listed below [5]:

- wide dynamic range;
- instant start-up;
- digital output;
- output independent of some environmental conditions (acceleration, vibration or shock);
- high rate capability;
- easy self-test;
- system design flexibility;
- extended running life.

In 1913, the Sagnac effect was proposed and became the fundamental principle for optical gyroscopes [5]. The fiber optic gyroscope (FOG) typically utilizes the Sagnac effect to measure angular velocity. Generally, a FOG consists of fiber optic coils. In order to measure rotation, two beams of light are emitted to the coil in clockwise and counter-clockwise directions. As shown in Figure 2.6, the dashed line represents the beam travelling in clockwise direction and the solid line represents the beam travelling in counter-clockwise direction. θ is the angle that the gyro turns whilst the beams are in flight. When the FOG starts to rotate in clockwise direction, the beam traveling in clockwise direction will have a longer path to the end of fiber than the beam traveling in counter-clockwise direction. Hence, the travel time of two opposite directions are different, which are computed as follows:

Clockwise path:

$$\Delta t_1 = \frac{2\pi r + r \cdot \theta}{c} \quad (2.1)$$

Counter-clockwise path:

$$\Delta t_2 = \frac{2\pi r - r \cdot \theta}{c} \quad (2.2)$$

The angular velocity ω is thereby computed from the difference of travel time:

$$\omega = \frac{c^2(\Delta t_1 - \Delta t_2)}{4\pi r^2} \quad (2.3)$$

Where r is the radius of fiber optic coils and c is the speed of light.

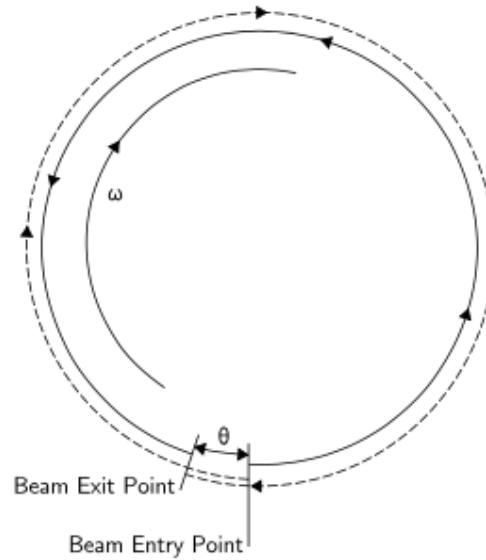


Figure 2.6: The Sagnac effect [7].

The ring laser gyroscope (RLG) is also based on the principle of Sagnac effect. The typical difference between FOG and RLG is that in RLG, the laser beam is directed around a closed path using mirrors rather than an optical fiber, as shown in Figure 2.7.

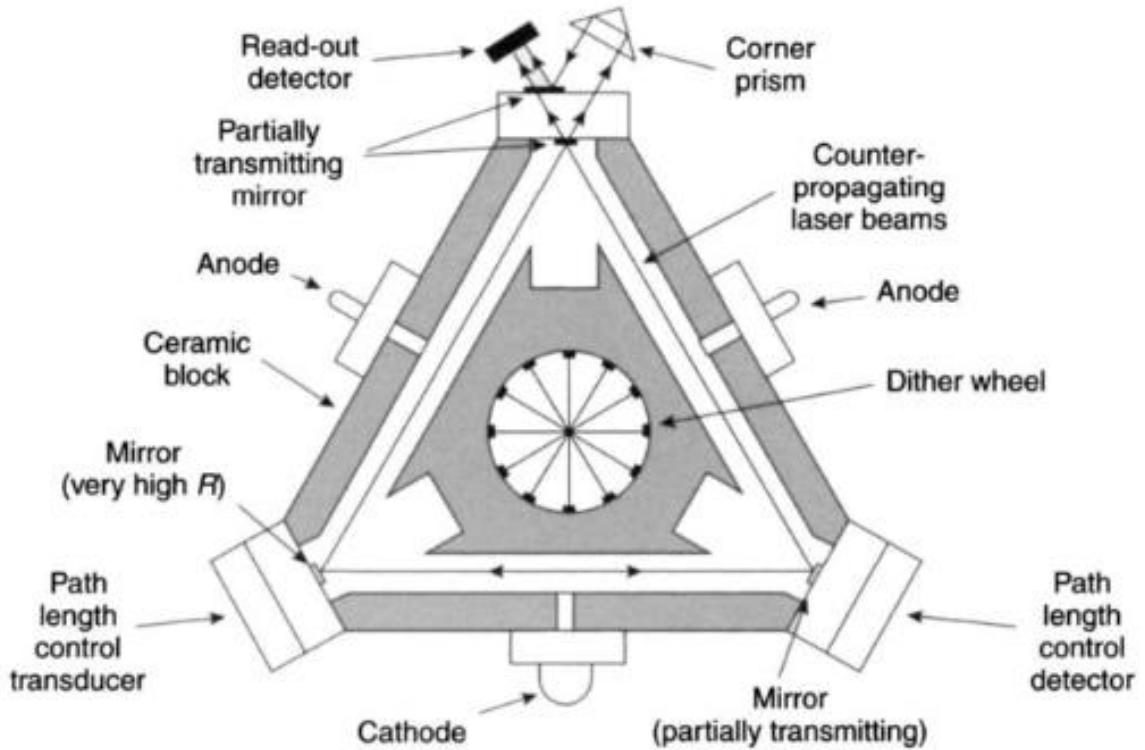


Figure 2.7: Schematic diagram of a ring laser gyroscope [5]

Unlike mechanical gyroscopes, optical gyroscopes contain no moving parts and can be woken up just in few seconds. The accuracy of an optical gyroscope depends largely on the length of the optical transmission path (usually the larger the better), which is usually limited by the size of the device.

2.2.3 MEMS gyroscope

Silicon-based MEMS gyroscopes are commercially available devices [8]. In particular, MEMS gyroscopes with their reduced cost, size, weight, and power consumption have a lot of applications especially in automotive and consumer electronics [9]. In the past few years, MEMS gyroscopes has been widely used in automotive control and safety systems in the automotive industry, such as vehicle navigation system, rollover detection and vehicle dynamic control system. Recently, the development of autonomous driving car also makes use of MEMS gyroscopes widely. The automotive industry has become a very important driver for the development of MEMS sensors. In consumer electronics field, micromachined inertial sensors are used in Optical Image Stabilization (OIS) systems for cameras and camcorders, in dead reckoning personal navigation systems (integrated in cell-phones), in 3D-pointing devices such as game controller and virtual reality headsets [10].

MEMS gyroscopes mostly don't have rotating parts that require bearings and hence are able to be manufactured through micromachining technology easily. Almost all MEMS gyroscopes integrate vibrating elements to measure the Coriolis effect, which is the basic operating principle. The angular velocity of rotation is able to be calculated afterward. These sensors rely on the detection of the force acting on a vibrating mass that is subject to linear vibratory motion in a frame of reference which is rotating about an axis perpendicular to the axis of linear motion. The resulting force, the Coriolis force, acts in a direction, which is perpendicular to both the axis of vibration and the axis about which the rotation is applied [5]. The generation of Coriolis force is shown in Figure 2.8.

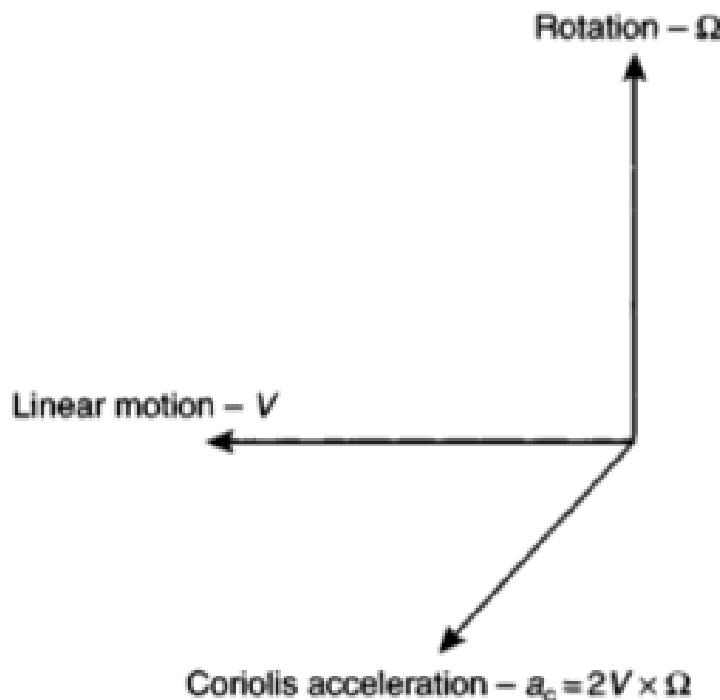


Figure 2.8: Generation of Coriolis force [5]

A very early prototype of silicon MEMS gyroscope was demonstrated by the Charles Stark Draper Laboratory in 1991[11]. The structure is shown as Figure 2.9. It was designed as a double gimbal vibratory gyroscope supported by torsional flexures. The vibrating mechanical part is made of p++ silicon. The inner gimbal is a platform that can vibrate around the x-axis and is connected to the outer gimbal by torsional flexures. The gyro element is a gold inertial mass that electroplated onto the inner gimbal. The outer gimbal is a rectangular frame that is connected to the support base plate by a thin beam, allowing it to vibrate around the y-axis. The electrodes placed on the outer gimbal drive it electrostatically at a constant amplitude, and the oscillation is transferred to the inner gimbal along the stiff axis of the inner flexures. When the sensor starts to rotate around z-axis, the resulting Coriolis force will cause the inner gimbal to oscillate about its weak axis and generate the same frequency as the drive frequency. By measuring the differential change in capacitance between the inner gimbal and a pair of electrodes, the readout of the output motion is recorded.

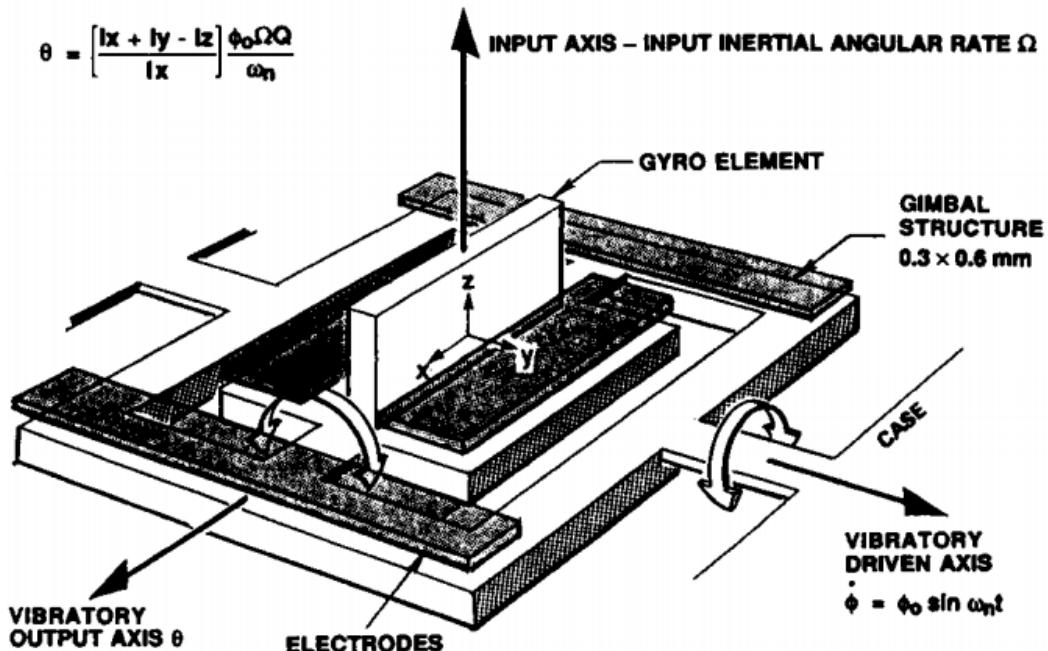


Figure 2.9: Early prototype of MEMS gyro [11]

With several years of development, a number of MEMS gyroscopes has been designed and manufactured. MEMS gyroscopes nowadays can be classified as: tuning fork, vibrating beams, vibrating shells and vibrating plates. All of these MEMS gyroscopes share the same principles. Figure 2.10 and Figure 2.11 show two examples of these MEMS gyroscopes. More information about MEMS gyroscopes are discussed in literature [5][11][12][13].

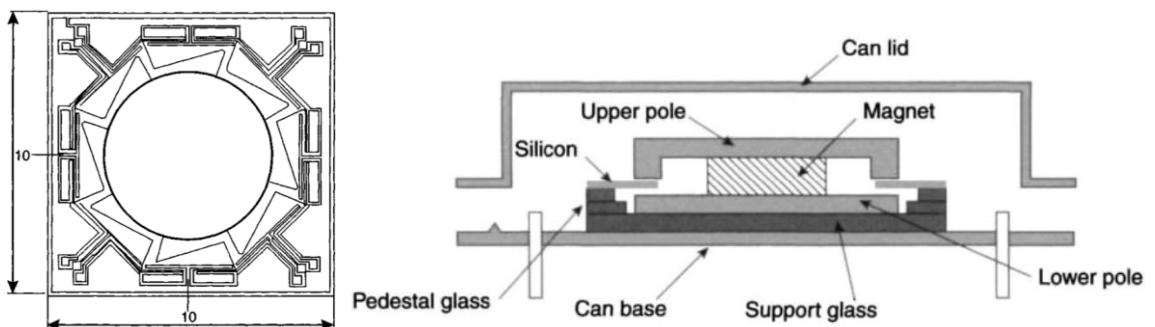


Figure 2.10: BAE System's silicon vibrating structure gyroscope [5]

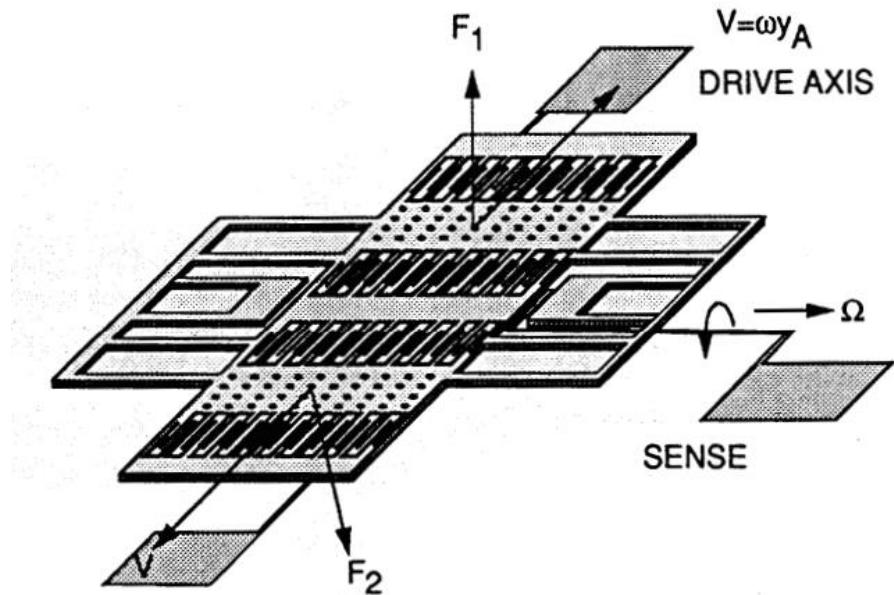


Figure 2.11: Tuning fork MEMS gyroscope [12]

MEMS gyroscopes currently are still far less accurate than traditional mechanical and optical gyroscopes, but the performance of MEMS gyroscopes has been greatly improved in the past decades. In case that the accuracy of MEMS gyroscopes can be achieved is similar to conventional gyroscopes, the advantages of MEMS gyroscopes will drive them to be more widely accepted by the industry. The main advantages of MEMS gyroscopes includes [5]:

- Small size;
- Low weight;
- Rugged construction;
- Low power consumption;
- Short start-up time;
- Inexpensive to produce (in high volume);
- High reliability;
- Low maintenance ;
- Compatible with operations in hostile environments;

2.3 Accelerometer Principle of Operation

There are different types of accelerometers that are used for different applications. This section will give an overview of accelerometers and their principles of operation.

2.3.1 Mechanical accelerometer

The mechanical accelerometer shown as Figure 2.12 has a mass suspended by springs. The displacement can be measured by the displacement pick-off. If there's movement happens, the displacement pick-off will sense the movement and give a signal that is proportional to the force F acting in that direction. The acceleration in that direction can be computed by applying Newton's second law formula $F = ma$.

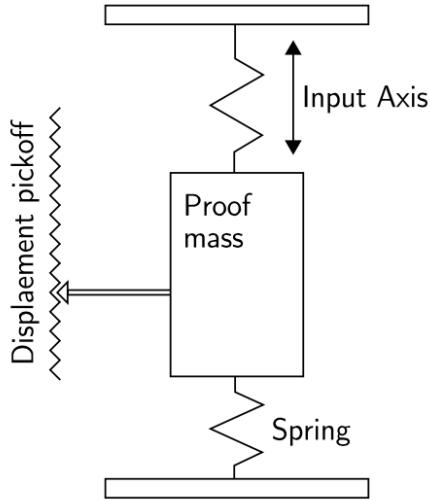


Figure 2.12: A spring accelerometer [5]

Another kind of mechanical accelerometer can be manufactured in the form of pendulum as shown in Figure 2.13. The principle is very similar to spring accelerometer. It detects the displacement in the moving direction. The relation between acceleration and measured displacement is shown as formula below.

$$\frac{d}{L} \approx \alpha = \frac{m}{k} \cdot a \quad (2.4)$$

Where

k is hinge parameter

L is pendulum length

m is mass

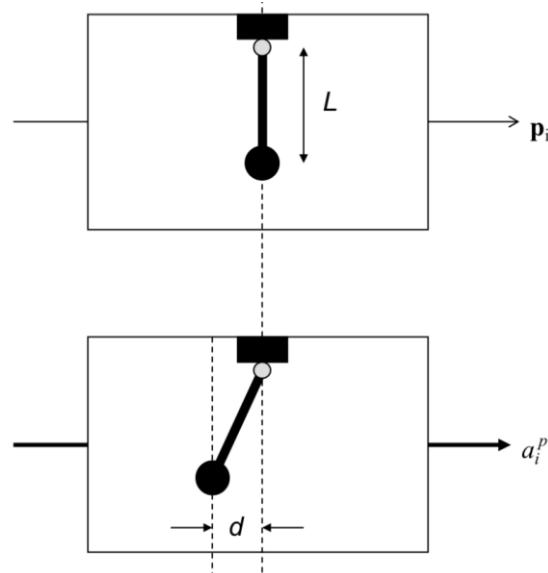


Figure 2.13: Pendulum accelerometer

2.3.2 Optical accelerometer

An example of fiber optic accelerometer as shown in Figure 2.14 shares the identical principle of operation as mechanical ones. The pendulum of the example is made up of optical fiber with a mass attached and a solid-state laser coupled into the top. Moreover, the bottom of the mass is a micro-lens. If an acceleration is applied to the sensor, the bottom will deflect and the displacement will be detected and measured by laser light that passes through the optical fiber and is focused onto a two-dimensional photosensitive array. The array in Figure 2.14 is a charge coupled imaging device (CCID), which is able to detect the displacement in both x and y directions. Due to this, the accuracy is limited by the pixel density of the photosensitive array[5]. There are more types of optical accelerometer are introduced in [5][14].

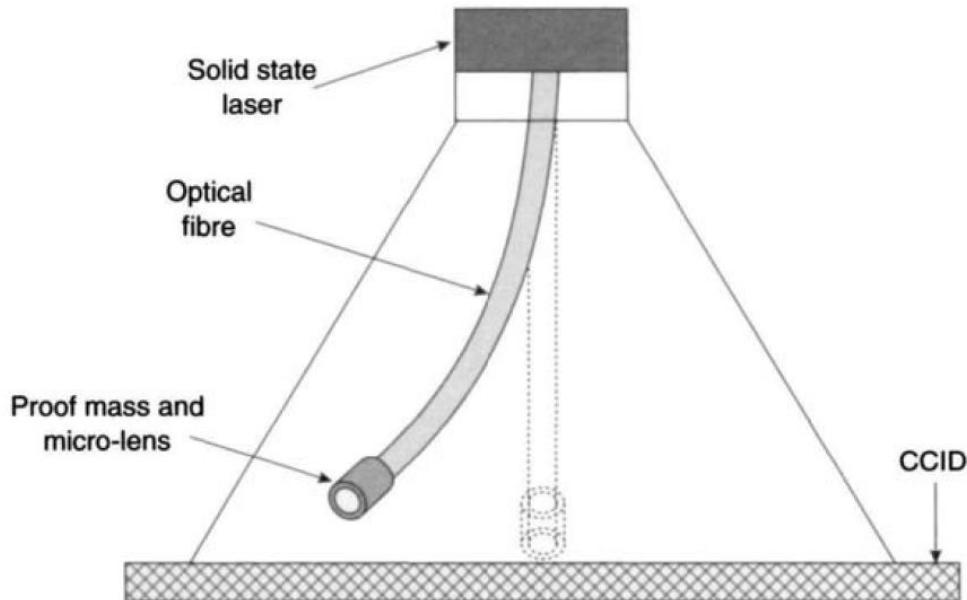


Figure 2.14: Pendulous fiber optic accelerometer [5]

2.3.3 MEMS accelerometer

MEMS accelerometers share the same advantages of MEMS gyroscopes. MEMS accelerometers are nowadays becoming more and more widely used in different application fields. MEMS accelerometers basically share the same principles as mechanical and state-solid accelerometers. They are designed and manufactured in tiny size and cheaper cost. There are mainly two kinds of MEMS accelerometers, which are different in the way of sensing acceleration [15]:

- Detecting the mass displacement by hinges or flexures in the presence of acceleration.
- Detecting the change in frequency of a vibrating element caused by the change in tension as a result of acceleration.

Based on the different way of sensing acceleration, various types of MEMS accelerometers are in the market now, including pendulous mass, resonant, tunneling and electrostatically levitated MEMS accelerometers.

An example of in-plane (X-axis or Y-axis) MEMS accelerometer is shown in Figure 2.15. The acceleration is detected by means of comb electrodes (rotors) attached to the proof mass and comb electrodes fixed on the substrate (stators). The rotors in our example are in blue and stators are in red and green. All the red stator fingers are wired together to form a single capacitance plate and all the green stator fingers are wired together to form another single

capacitance plate. When an acceleration is applied to the sensor, the proof-mass starts to move and will change the gap spacing between rotors and stators. The capacitance will thereby change inversely proportional to the square of the gap spacing. Due to this, the displacement of proof-mass is converted to capacitance change, which will also be converted and amplified into a voltage signal later on.

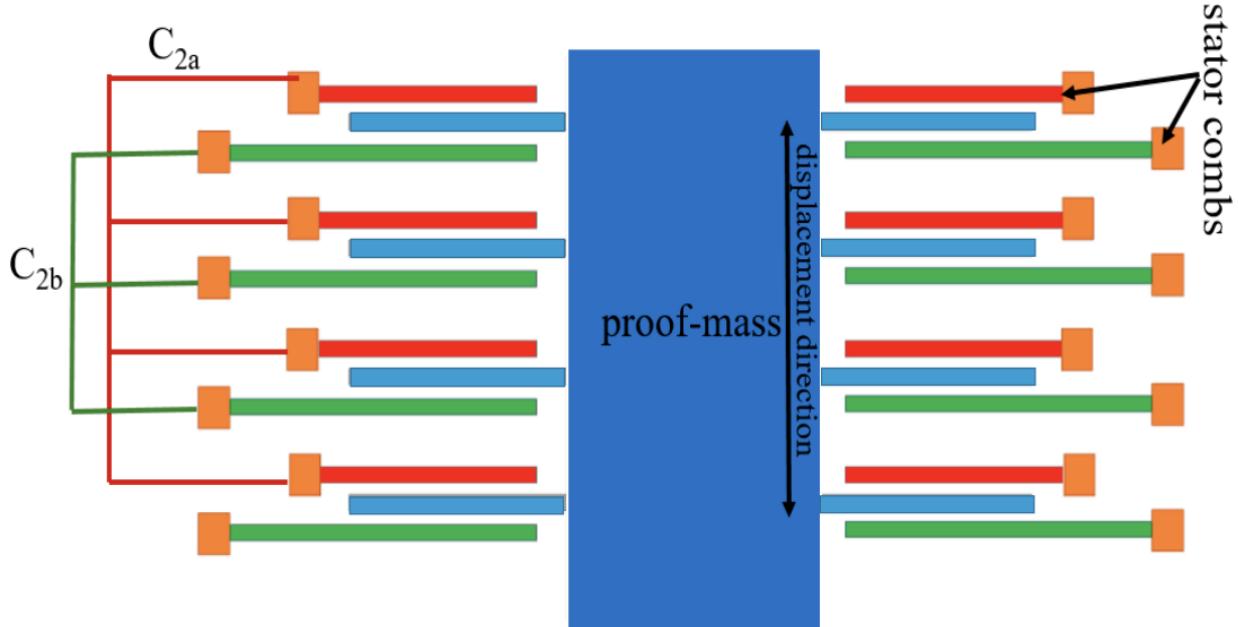


Figure 2.15: In-plane pendulous accelerometer [16]

An example of structure designed for sensing out-of-plane acceleration (Z-axis) is shown in Figure 2.16. In this case, the whole proof-mass is designed to be an electrode. Moreover, the top and bottom act as two electrode plates. When an acceleration is applied in the vertical direction as shown in Figure 2.16, the gap spacing between top electrode and proof-mass decreases, thus the capacitance C_+ will increase. In contrast, the capacitance C_- will decrease. The change of capacitance can be converted to a voltage signal later.

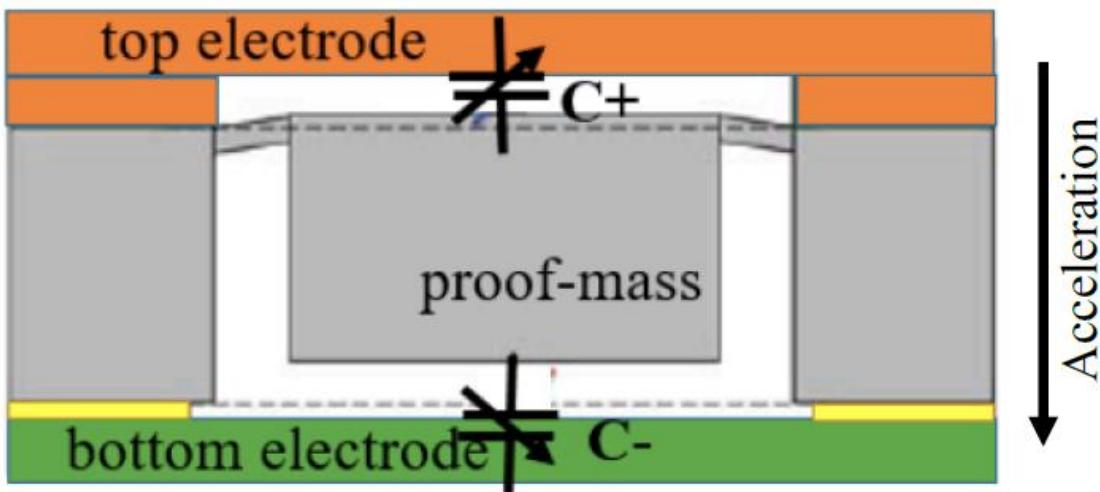


Figure 2.16: Z-axis accelerometer [16]

2.4 MEMS Sensors Noise Characteristics

Various kinds of errors and noises may exist in the MEMS inertial sensors. This section will introduce the main types of errors and noises, including constant bias error, bias instability, angle/velocity random walk, temperature effects, calibration errors, etc.

2.4.1 Constant bias

The constant bias of MEMS inertial sensors refers to the offset of the sensor's output signal data from the true value. It is usually estimated by taking a long-term average of the sensor's output signal when the sensor keeps static. The constant bias for gyroscopes and accelerometers is usually expressed in ($^{\circ}/h$) and (m/s^2). Afterwards, the constant bias can be compensated by simply subtracting from the output. Additionally, when compensating for accelerometer, the effect of gravity should be also taken into consideration, which can be achieved by sensor calibration.

2.4.2 Angle/Velocity random walk

The “Random Walk” is used in statistics to describe a situation where the output of a system is driven by random, uncorrelated “steps” [17]. The output of the MEMS inertial sensor should keep zero when at rest, however, due to the perturbation of so called thermo-mechanical noise, the output is sometimes above zero and sometimes below zero. Angle/Velocity random walk (ARW/VRW) is an independent noise specification given in unit of ($^{\circ}/\sqrt{h}$) for gyros and ($m/s/\sqrt{h}$) for accelerometers. It is typically applied to angle/velocity calculation to describe the average deviation or error that will occur when integrating output signal. The ARW/VRW will increase when integrates longer, for instance, a gyro has an ARW of $0.1^{\circ}/\sqrt{h}$, it means that the angle deviation will be 0.1° after 1 hour, and it will be $\sqrt{2} \cdot 0.1 \approx 0.14^{\circ}$ after 2 hours of operation. Moreover, the noise remains the same whether the sensor is static or not.

In general, the noise specifications are quoted in different ways by different manufacturers. For example, for the case of ARW, some of manufacturers prefer to quote an ARW; some of manufacturers prefer to quote a PSD ($(^{\circ}/h)^2/Hz$) or FFT noise density ($^{\circ}/h/\sqrt{Hz}$); some of manufacturers prefer to quote a total noise, one or three sigma variation in the output of the sensor. It is possible to convert between the various specifications by the following equations [17]:

$$ARW(^{\circ}/\sqrt{h}) = \frac{1}{60} \cdot \sqrt{PSD((^{\circ}/h)^2/Hz)} \quad (2.5)$$

$$ARW(^{\circ}/\sqrt{h}) = \frac{1}{60} \cdot FFT(^{\circ}/h/\sqrt{Hz}) \quad (2.6)$$

$$ARW(^{\circ}/\sqrt{h}) = \frac{1}{60} \cdot \sigma(^{\circ}/h) \cdot \frac{1}{\sqrt{BW(Hz)}} \quad (2.7)$$

Where σ is the standard deviation of the signal and BW is the bandwidth of the sensor in Hz.

2.4.3 Bias instability

As stated before, bias refers to a long-term average of inertial sensor's output. However, due to the effect of flicker noise, the bias wanders over time. Bias instability is a specific noise that refers to changes in the bias measurement [18]. Generally, bias instability can be modeled

with a second order random walk in angle/velocity. In reality, it is not a real random walk, but a good approximation for short periods of time.

2.4.4 Temperature effects

In reality, the bias of the sensor's output signal is fluctuated while the working condition is changed. Temperature is one of the condition changes that influences the bias significantly. Normally, the relationship between temperature and bias is different from device to device, which is mostly non-linear. If the MEMS IMU integrates the temperature sensor on the chip, it is therefore possible to apply temperature compensation to the output data. This will be the focus of this master thesis.

2.4.5 Calibration errors

Calibration errors refer to errors in the scale factors, alignments and output linearities, which appear as bias errors while the inertial sensor is running. Such errors lead to the accumulation of additional drift in the integrated signal, the magnitude of which is proportional to the rate and duration of the motions [19].

3. ALLAN VARIANCE

Allan Variance was proposed by David W. Allan in 1996 [20]. In 1998, the IEEE standard accepted the Allan Variance as a noise identification method for linear, single, non-gyroscopic accelerometer analysis (IEEE Std. 1293-1998) [21]. In 2003, H.Hou and El-Sheemy published the paper “Inertial Sensors Error Modeling Using Allan Variance”, which was the first time that applied the Allan Variance to the MEMS inertial device’s noise identification [22].

Allan Variance is now the most common time-domain measure of frequency stability. Similar to standard deviation, it is a measure of fractional frequency fluctuations, but has the advantage of converging against most types of clock noise. Several versions of Allan Variance can provide better statistical confidence, can distinguish white noise from flicker phase noise, and can describe temporal stability [23]. In this thesis, Allan Variance method will be applied to do statistical analysis.

3.1 Allan Variance Principles

The original non-overlapped Allan Variance, also called two-sample variance, is the standard time domain measure of frequency stability. It is defined as [20]:

$$\sigma_y^2(\tau) = \frac{1}{2(M-1)} \sum_{i=1}^{M-1} [y_{i+1} - y_i]^2 \quad (3.1)$$

Where y_i is the i th of M fractional frequency values averaged over the measurement interval τ . Note that these y symbols are sometimes shown with a bar over them to denote the averaging.

In terms of phase data, AVAR is calculated as [20]:

$$\sigma_y^2(\tau) = \frac{1}{2(N-2)\tau^2} \sum_{i=1}^{N-2} [x_{i+2} - 2x_{i+1} + x_i]^2 \quad (3.2)$$

Where x_i is the i th of the $N = M + 1$ phase values spaced by the measurement interval τ .

3.2 Overlapping Allan Variance

The overlapping Allan Variance is more commonly used than the standard method. The overlapping method utilize overlapping samples, whereby the calculation is performed by utilizing all possible combinations of the dataset, as shown in Figure 3.1.

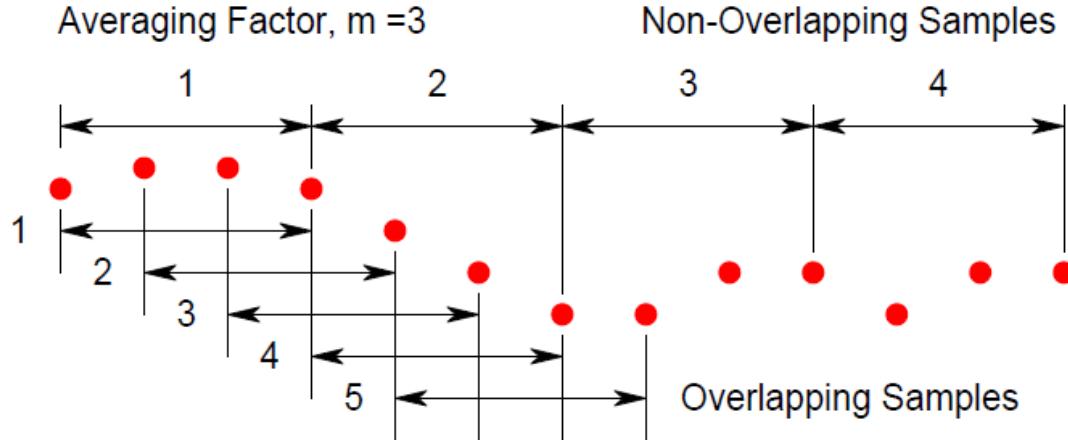


Figure 3.1: Comparison of non-overlapping and overlapping samples[23]

In case of the non-overlapping method, the stride τ is equal to the averaging period $m\tau_0$. For the fully overlapping method, the stride is equal to sample period τ_0 . Keep the sensor static and collect M samples of frequency data with the sample period as τ_0 . The averaging time $\tau = m\tau_0$, where m is the averaging factor. The value of m can be chosen arbitrarily if $m < (M - 1)/2$. Hence, AVAR can be estimated by the expression[23]:

$$\sigma_y^2(\tau) = \frac{1}{2m^2(M - 2m + 1)} \sum_{i=1}^{M-2m} \left\{ \sum_{j=i}^{j+m-1} [y_{i+m} - y_i] \right\}^2 \quad (3.3)$$

Where y_i is the i th of M fractional frequency values averaged over the measurement interval τ .

However, the intensive inner summation of expression above takes much computational effort. Due to this, the expression above is seldom used when the data sets are large. In terms of phase data, AVAR can also be estimated by a faster expression[23]:

$$\sigma_y^2(\tau) = \frac{1}{2(N - 2m)\tau^2} \sum_{i=1}^{N-2m} [x_{i+2m} - 2x_{i+m} + x_i]^2 \quad (3.4)$$

Where x_i is the i th of the $N = M + 1$ phase values spaced by the measurement interval τ .

Moreover, the fractional frequency datasets can also use the faster formula, while the datasets are firstly integrated.

The use of overlapping method improves the confidence of the resulting stability estimate, but in the other hand takes more computational time. Despite the fact that overlapping samples are not completely independent, the effective number of degrees of freedom are nevertheless increased. Hence, the confidence in the estimation is improved. Since the overlapping method is used more often, the term AVAR has also come to be used as the abbreviation of overlapping Allan Variance, and ADEV for its square root.

Finally, take the square root of Allan Variance and obtain the value of Allan Deviation for a particular value of τ by the expression[24]:

$$ADEV(\tau) = \sqrt{AVAR(\tau)} \quad (3.5)$$

In order to plot the ADEB curve, the computation should be repeated for several times for a sequence of τ values, which can be chosen arbitrarily. The Allan Deviation plots are usually plotted as values of Allan Deviation over τ on a log-log plot.

3.3 Noise Identification

The AVAR $\sigma^2(\tau)$ is related to the PSD of the noise items and the relation between AVAR and the two-sided PSD is given by the expression:

$$\sigma^2(\tau) = 4 \int_0^\infty S_\Omega(f) \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} df \quad (3.6)$$

Where

$S_\Omega(f)$ is the PSD of the inertial sensor's rate output data.

f is sideband frequency (Hz).

The equation above is used to calculate the AVAR from the rate noise PSD. An explanation is that the AVAR is proportional to the total noise power of the rate output when passed through a filter with the transfer function of the form $\sin^4(x)/(x)^2$. This particular transfer function is the result of the method used to create and operate on the clusters[25]. It is also evident that the filter bandwidth depends on τ , which means that different types of random processes can be examined by adjusting the filter bandpass, namely by varying τ . Hence, the AVAR can be used as a method to identify and quantify different noise terms that exist in the data.

Different types of random process normally cause slopes with different gradients to appear on the Allan deviation plot and make it possible to read its numerical parameters directly from the plot. These noise items basically include quantization noise, angle/velocity random walk, bias instability, rate random walk and rate ramp, which are shown in the Figure 3.2 as a combination of all possible noise terms.

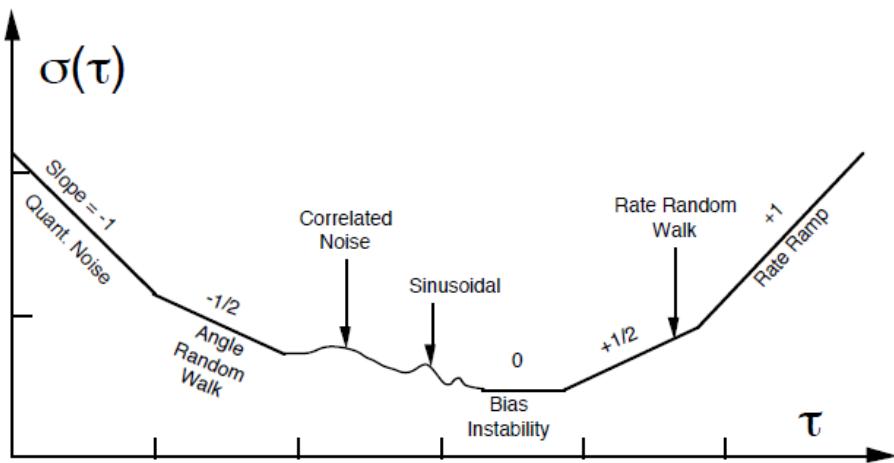


Figure 3.2: A log-log plot of Allan Deviation possible noise items [25]

For the MEMS low-cost IMUs, angle/velocity random walk and bias instability are more frequently used to evaluate their performance. The other noise items sometimes are hardly to be characterized on the plot. In this case, the subsection will only discuss how to get the angle/velocity random walk and bias instability from the plot of Allan deviation.

3.3.1 Angle/Velocity random walk

The associated rate noise PSD of ARW/VRW is represented by[25]:

$$S_\Omega(f) = N^2 \quad (3.7)$$

Where N is the coefficient of ARW/VRW.

Substitution of equation (3.7) in equation (3.6) and performing the integration yields[25]:

$$\sigma^2(\tau) = \frac{N^2}{\tau} \quad (3.8)$$

In log-log plot of $\sigma(\tau)$ versus τ has a slope of $-1/2$ and the numerical value of N can be obtained directly by reading the slope line at $\tau = 1$, shown as Figure 3.3.

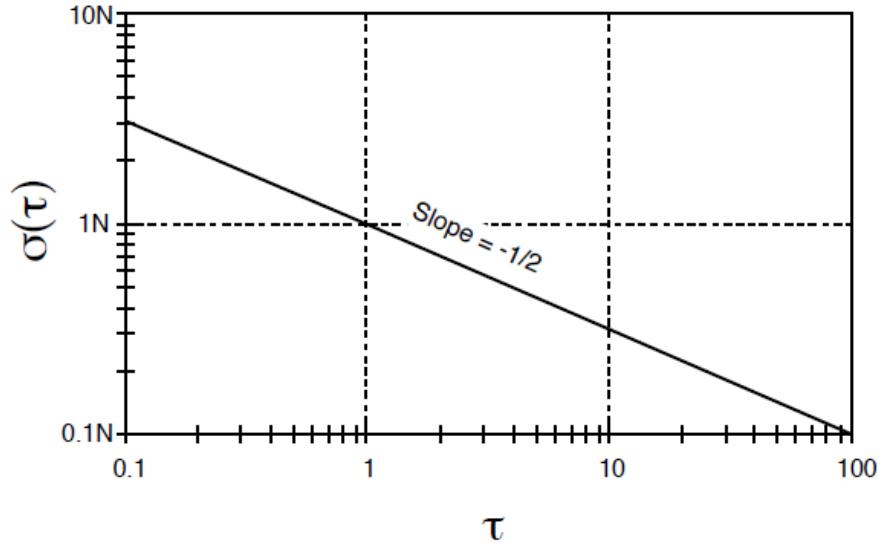


Figure 3.3: $\sigma(\tau)$ Plot for ARW/VRW [25]

3.3.2 Bias instability

The associated rate noise PSD of bias instability is represented as[25]:

$$S_\Omega(f) = \begin{cases} \left(\frac{B^2}{2\pi}\right)\frac{1}{f} & f < f_0 \\ 0 & f > f_0 \end{cases} \quad (3.9)$$

Where

B is the bias instability coefficient

f_0 is the cutoff frequency

Substitution of equation (3.9) in equation (3.6) and performing the integration yields[25]:

$$\sigma^2(\tau) = \frac{2B^2}{\pi} \left[\ln 2 - \frac{\sin^3 x}{2x^2} (\sin x + 4x \cos x) + C_i(2x) - C_i(4x) \right] \quad (3.10)$$

Where

x is $\pi f_0 \tau$

C_i is the cosine-integral function

When τ is much longer than the inverse of the cutoff frequency, the PSD equation is[25]:

$$\sigma^2(\tau) = \frac{2B^2}{\pi} \ln 2 \quad (3.11)$$

The Figure 3.4 shows that the AVAR for bias instability reaches a plateau for τ much longer than the inverse cut off frequency. Hence, the flat region of the plot can be examined to estimate the limit of the bias instability and the cutoff frequency of the underlying flicker noise. The bias instability value is therefore the ADEV at the region where the slope equals zero. The value of bias instability can be read directly off with a scaling of $\sqrt{\frac{2 \ln 2}{\pi}} \approx 0.664$.

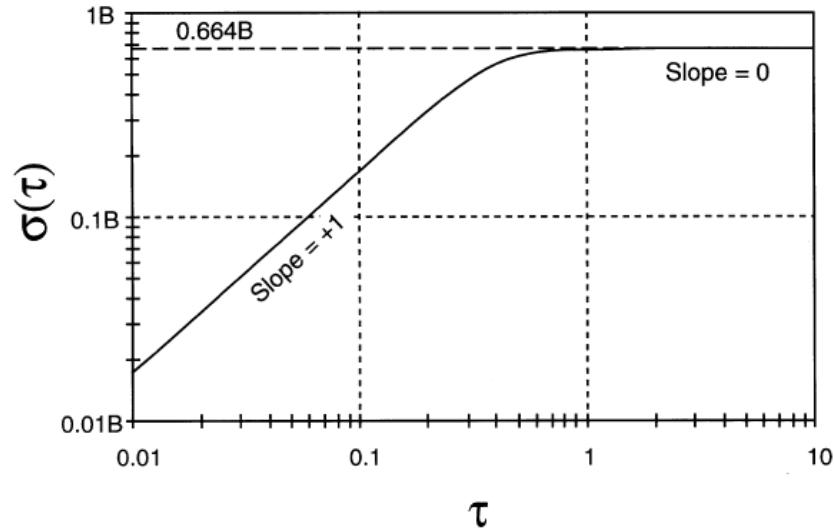


Figure 3.4: $\sigma(\tau)$ Plot for bias instability (for $f_0 = 1$) [25]

4. HARDWARE IMPLEMENTATION

In order to evaluate how the temperature influences the performance of the low-cost IMU, this master thesis selects the MEMS based MPU-9250 as the test hardware. It is a very cheap MEMS IMU product designed by InvenSense. The manufacturer is SparkFun. In order to get a better shape of ADEV plot, this thesis also writes a specific Python driver to read out the data from MPU-9250. The driver allows the data to be written to FIFO buffer at the sample rate, in the other word, the time interval between two neighboring samples are the same. This chapter will not only give an overview of hardware and software, but also introduces how experiment was set up to keep sensor static and control working temperature.

4.1 Hardware Overview

4.1.1 MPU-9250 MEMS IMU

The sensor MPU-9250 is a 9 degrees of freedom multi-chip module IMU that consists of two dies integrated into a single QFN package. One die houses the 3-Axis gyroscope and the 3-Axis accelerometer. The other die houses the AK8963 3-Axis magnetometer from Asahi Kasei Microdevices Corporation. This thesis will only focus on the 3-Axis gyroscope and the 3-Axis accelerometer.

MPU-9250 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and another three for digitizing the accelerometer outputs. For precision tracking of motions' change, the sensor features a user programmable gyroscope and a user programmable accelerometer. The full-scale range of gyroscope can be chosen from ± 250 , ± 500 , ± 1000 , and $\pm 2000^{\circ}/\text{sec}$ (dps). The full-scale range of accelerometer can be chosen from $\pm 2g$, $\pm 4g$, $\pm 8g$, and $\pm 16g$.

The device features both I²C and SPI serial interfaces, the communication between all registers of the device is performed using either I²C up to 400kHz or SPI up to 1MHz. For applications requiring faster communications, the sensor and interrupt registers may be read using SPI at 20MHz. Moreover, the auxiliary master I²C bus provides the possibility for reading data from external sensors, for example, pressure sensor. The VDD operating range is from 2.4V to 3.6V, and a separate digital IO supply, VDDIO is from 1.71V to VDD.

Other industry-leading features include programmable digital filters, a precision clock with 1% drift from -40 °C to 80°C, a digital-output temperature sensor and programmable interrupts. A 512-byte FIFO buffer is also integrated on the chip, which enables the applications processor to read the data in bursts.

The block diagram of MPU-9250 is shown as Figure 4.1 and more hardware information can be seen from the "MPU-9250 Product Specification Revision 1.1" [26] and "MPU-9250 Register Map and Descriptions Revision 1.6" [27].

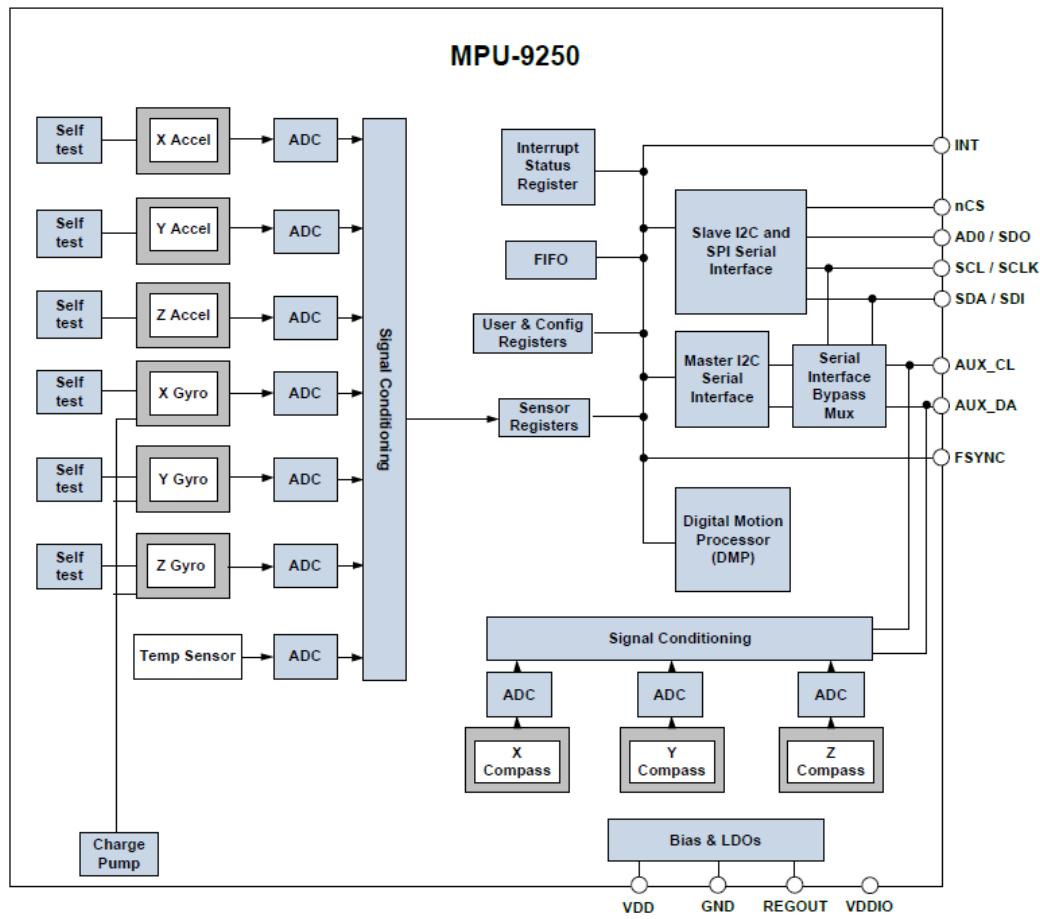
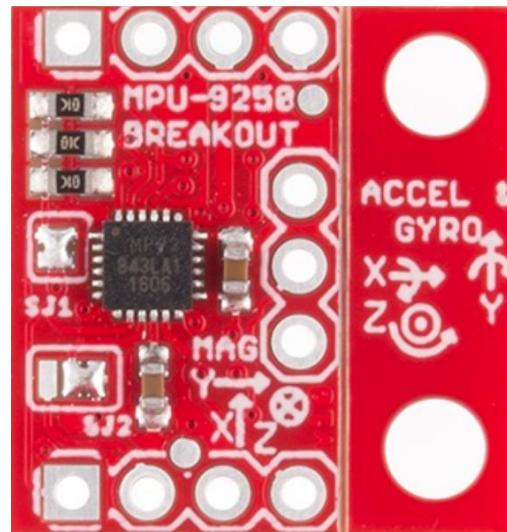
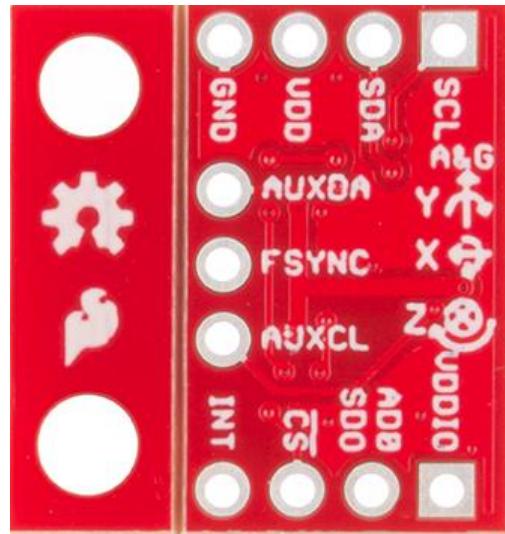


Figure 4.1: MPU-9250 Block Diagram [26]

The PCB board of MPU-9250 used in this thesis is shown in Figure 4.2. The PTHs are distributed at the border of the PCB in three rows of three or four. The top row is all one needs to get most of the functionality of IMU, including the I²C and power interface. The second mostly used is the PTHs along the bottom, including the address pin AD0/SDO, the interrupt pin INT, the chip select pin for SPI and the IO power supply pin VDDIO. The non-breadboard-compatible row is used for features like running other I²C devices, which will not be used here.



(a)



(b)

Figure 4.2: SparkFun MPU-9250 PCB board (a) Top view (b) Bottom view

4.1.2 Single-board computer

Only a single IMU chip is not able to collect the data in a file, hence a single-board computer needs to be connected with MPU-9250. The very popular and cheap single-board computer Raspberry Pi 3 Model B+ shown as Figure 4.3 is used in the experiment, which provides 1.4GHz 64-bit quad-core processor and 1GB LPDDR2 SDRAM. The operating system installed is Raspbian OS, which is based on Debian with Linux core. The micro SD port for operation system and storing data is 64GB. More importantly, it provides 40-pin GPIO header, which supports both I²C and SPI communication. In this case, the MPU-9250 is able to build connection with the Raspberry Pi 3 Model B+. Afterwards, the data collecting software can be run on Raspberry Pi 3 Model B+ by using Linux commands in the terminal window.

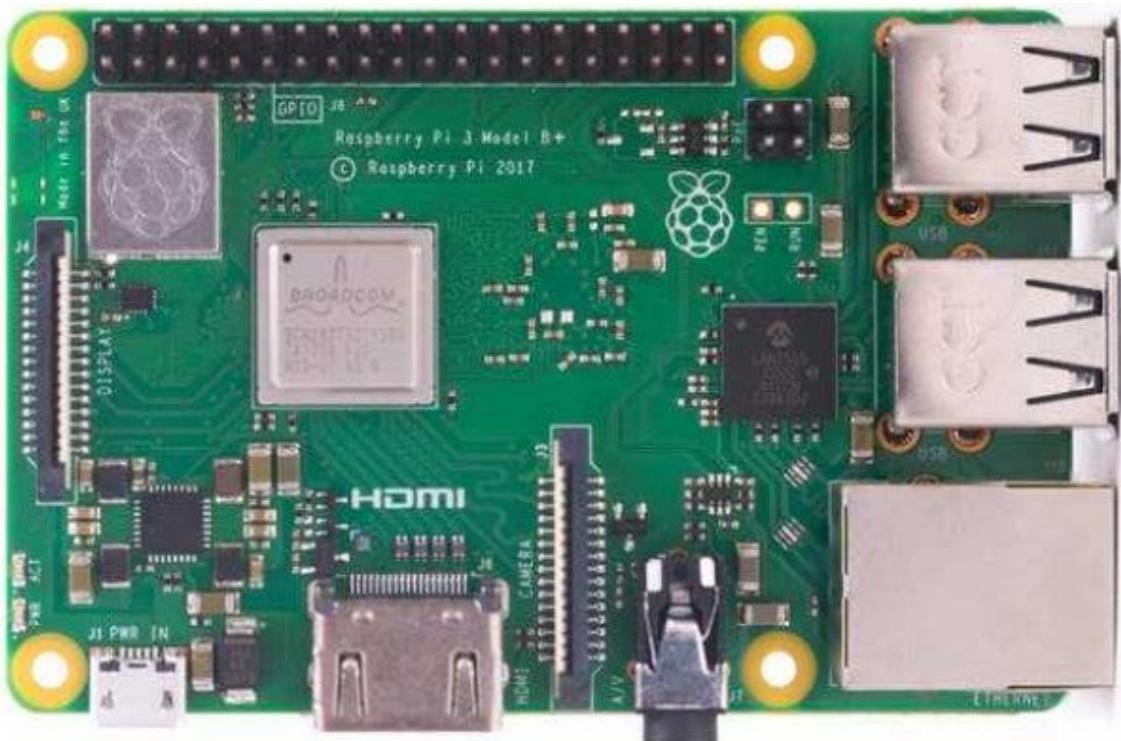


Figure 4.3: Raspberry Pi 3 Model B+ board

4.2 Software Design

The software serves two purposes: communication with the MPU-9250 and collecting data. As it mentioned before, MPU-9250 supports both I²C and SPI communication. In this thesis, I²C communication is used, which is fast enough for the experimental purpose. The software is a driver that written in Python 3.7. A Python toolbox called *smbus2* is used to build connection between MPU-9250 and Raspberry Pi 3 Model B+ and read block data from MPU-9250.

Normally, MPU-9250 driver reads the gyroscope data, accelerometer data and temperature data from the output registers directly. For example, registers 59 to 64 are used for 3-Axis accelerometer measurements, registers 65 and 66 are used for temperature measurement, registers 67 to 72 are used for 3-Axis gyroscope measurements. However, these drivers cannot guarantee that all of the time interval between neighboring samples are the same. In this case, the shape of ADEV plots maybe distorted, because the true averaging time of different clusters are not the same. A 512-byte FIFO buffer is therefore used for reading data. FIFO is the abbreviation of “First In, First Out”, which means that the first data come into the FIFO buffer will be read first as well. Once the data is read, it will be cleared from the FIFO buffer at once. The data comes into FIFO buffer at sample rate, so if there’s no FIFO overflow happens, the time interval between adjacent samples remains the same. MPU-9250 provides entire FIFO control and read processes. The FIFO configuration register determines which data is written into the FIFO, including accelerometer data, temperature readings, gyroscope data. A FIFO counter register can keep track of the number of bytes of valid data are contained in the FIFO buffer. The number of bytes should be less than 512 to avoid FIFO overflow. If the FIFO buffer happens to overflow, the time interval therefore is not the same.

When the MPU-9250 is powered on, the Python driver first does the initialization. Steps in the initialization process include setting the sampling rate, digital low-pass filter, FIFO mode and full-scale range of both the accelerometer and gyroscope. MPU-9250 sample rate divider register allows user to set the sample rate as they want, but only able to be used for 1kHz internal sampling. In this case, digital low-pass filter is activated and set to a relatively highest allowable bandwidth. While the performance of single board computer cannot support much computation, if the sampling rate is too high, the single board computer will be stuck sometimes and result in FIFO overflow. Hence, the sampling rate is set as 20Hz for both accelerometers and gyroscopes. When selecting the full-scale range for gyroscope, it is should be bigger than the expecting maximum angular velocity. For accelerometer, in most cases, a smaller full-scale range will give more sensitive output. The full-scale range in our case is set to +500dps for gyroscope and $\pm 4g$ for accelerometer.

After initialization, the Raspberry Pi 3 Model B+ starts to continuously collecting data from MPU-9250 over I²C bus for around 1h. Since 1h is enough to characterize all noise terms through ADEV plot. The data comes into FIFO in the order of x, y, z-axis acceleration (m/s^2), temperature ($^{\circ}C$), x, y, z-axis angular rate (rad/s). But here the software writes in the order of x, y, z-axis acceleration (m/s^2), x, y, z-axis angular rate (rad/s), temperature ($^{\circ}C$) into a *.txt file named after the operating time and sampling rate. An example of the format of the output data files is shown as Figure 4.4. The number 1,2,3 stands for x,y,z-axis in the header.

MASTER THESIS: TEMPERATURE DEPENDENCY OF A LOW-COST IMU

```

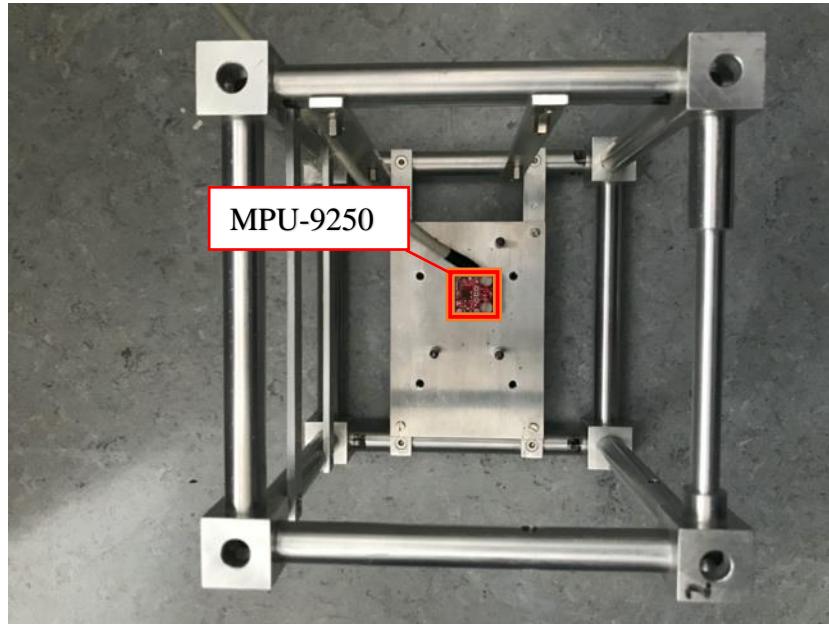
mpu_accel_1[m/^2], mpu_accel_2[m/^2], mpu_accel_3[m/^2], mpu_gyro_1[rad/s], mpu_gyro_2[rad/s], mpu_gyro_3[rad/s], temp[degC]
0.8786720092773437, 0.19632453613281248, 9.344569079589844, -0.03677179134860051, 0.022649291772688717, 0.01891882018659881, 4.062269745709408
0.8810662109374999, 0.22026655273437498, 9.363722692871093, -0.036238866836301946, 0.023182216284987275, 0.01945174469889737, 4.074250456764609
0.8810662109374999, 0.22505495605468748, 9.425971936035156, -0.037038253604749785, 0.023448678541136553, 0.02078405597964376, 4.053284212418006
0.9169792358398436, 0.2346317626953125, 9.397241516113281, -0.03543948006785411, 0.02424806530958439, 0.019984669211195925, 4.071255279000809
0.8810662109374999, 0.22266075439453123, 9.349357482910156, -0.034640093299406274, 0.02424806530958439, 0.019984669211195925, 4.0682601012370085
0.8930372192382812, 0.20350714111328125, 9.37808790283203, -0.03543948006785411, 0.02184990500424088, 0.01865235793044953, 4.0682601012370085
0.8834604125976562, 0.2059013427734375, 9.392453112792968, -0.03677179134860051, 0.023715140797285834, 0.01918528244274809, 4.056279390181807
0.9121908325195311, 0.229843359375, 9.404424121093749, -0.035173017811704836, 0.023715140797285834, 0.020251131467345206, 4.071255279000809
0.8307879760742187, 0.22026655273437498, 9.414000927734374, -0.035705942324003384, 0.022649291772688717, 0.01865235793044953, 4.07724563452841
0.9050082275390624, 0.22744915771484372, 9.3852705078125, -0.03543948006785411, 0.02211636726039016, 0.01945174469889737, 4.071255279000809
0.8882488159179687, 0.2059013427734375, 9.332598071289063, -0.036238866836301946, 0.023448678541136553, 0.01945174469889737, 4.074250456764609
0.86191259765625, 0.22266075439453123, 9.370905297851563, -0.03597240458015267, 0.023448678541136553, 0.01891882018659881, 4.08024081229221
0.8571241943359375, 0.22026655273437498, 9.390058911132812, -0.036238866836301946, 0.02424806530958439, 0.019984669211195925, 4.0682601012370085
0.8786720092773437, 0.21068974609375, 9.279925634765624, -0.03677179134860051, 0.023981603053435115, 0.019718206955046648, 4.07724563452841
0.8930372192382812, 0.20829554443359374, 9.344569079589844, -0.03490655555555555, 0.021050518235793042, 0.020251131467345206, 4.083235990056011
0.8212111694335937, 0.2537853759765625, 9.373299499511718, -0.035705942324003384, 0.02424806530958439, 0.01865235793044953, 4.059274567945607
0.852335791015625, 0.20350714111328125, 9.370905297851563, -0.03650532909245122, 0.022915754028837994, 0.018385895674300254, 4.086231167819811
0.9265560424804687, 0.19871873779296872, 9.390058911132812, -0.03677179134860051, 0.023715140797285834, 0.020517593723494487, 4.065264923473208
0.8738836059570312, 0.19153613281249998, 9.373299499511718, -0.035173017811704836, 0.022382829516539436, 0.020517593723494487, 4.065264923473208
0.8882488159179687, 0.21068974609375, 9.342174877929686, -0.03650532909245122, 0.02451452756573367, 0.02078405597964376, 4.059274567945607
0.8906430175781249, 0.22744915771484372, 9.313444458007812, -0.035173017811704836, 0.022915754028837994, 0.02078405597964376, 4.074250456764609
0.8834604125976562, 0.22026655273437498, 9.361328491210937, -0.03597240458015267, 0.022382829516539436, 0.01945174469889737, 4.071255279000809
0.8930372192382812, 0.2346317626953125, 9.361328491210937, -0.035173017811704836, 0.022915754028837994, 0.01918528244274809, 4.071255279000809
0.8858546142578124, 0.23702596435546874, 9.370905297851563, -0.03597240458015267, 0.02211636726039016, 0.017586508905852418, 4.071255279000809
0.9097966308593749, 0.21068974609375, 9.363722692871093, -0.03650532909245122, 0.023448678541136553, 0.019718206955046648, 4.062269745709408
0.8738836059570312, 0.2537853759765625, 9.34696328125, -0.03597240458015267, 0.023182216284987275, 0.020251131467345206, 4.0682601012370085
0.8571241943359375, 0.22744915771484372, 9.34696328125, -0.036238866836301946, 0.023448678541136553, 0.020517593723494487, 4.074250456764609
0.8978256225585937, 0.201112939453125, 9.442731347656249, -0.037038253604749785, 0.022649291772688717, 0.02078405597964376, 4.07724563452841
0.8858546142578124, 0.22026655273437498, 9.373299499511718, -0.03650532909245122, 0.02424806530958439, 0.019984669211195925, 4.062269745709408
0.9074024291992187, 0.22266075439453123, 9.363722692871093, -0.03650532909245122, 0.023448678541136553, 0.01865235793044953, 4.065264923473208

```

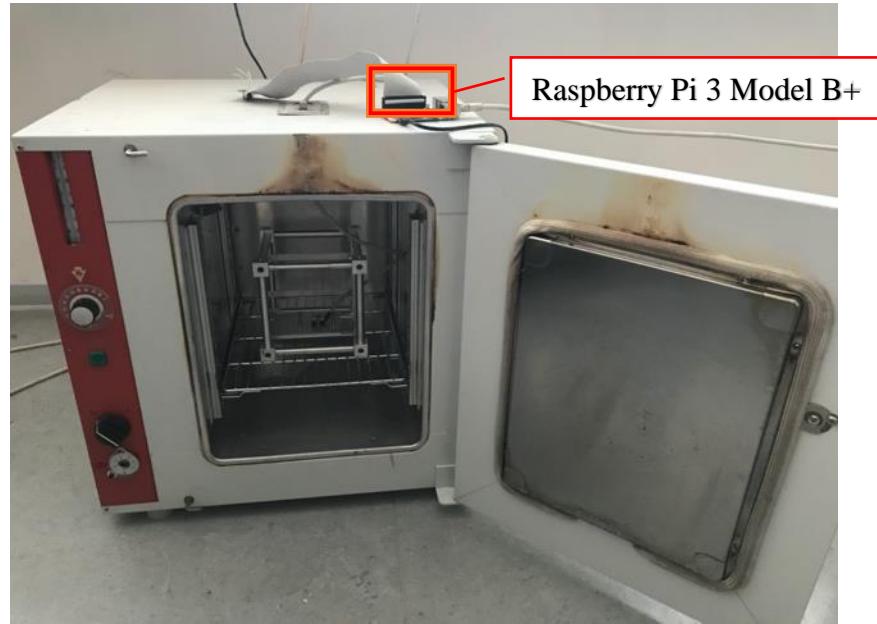
Figure 4.4: Example of output data format

4.3 Experimental Setup

The experiment was held at the laboratory of Institution for Navigation, University of Stuttgart. There are mainly three parts of experiment, including MPU-9250, Raspberry Pi 3B+ and an oven. The range of normal operating temperature that MPU-9250 is specified from -40°C to 80°C , but the oven unfortunately is only able to heat up. In this case, the temperature can only be set higher than the room temperature. In order to keep the MPU-9250 static, MPU-9250 was fixed on a cubic frame, which is shown in Figure 4.5 (a). Pinholes on the chip used for I²C communication were connected through a cable with Raspberry Pi 3B+. The Raspberry Pi 3B+ was put out of the oven to avoid the temperature influence on the single board computer. The whole experimental setup is shown in Figure 4.5.



(a)



(b)

Figure 4.5: Experimental setup (a)MPU-9250 (b) Devices

Different temperature points were set to evaluate the MPU-9250, the temperature controlled by the oven is however not able to reach the temperature point we want immediately. In other words, data collection started after the temperature was stabilized. In order to characterize how the dynamic change of temperature affects the performance of MPU-9250, a cooling down process was performed as well. While the temperature inside the oven reached 100°C and was stable, turned off the oven and kept acquiring data until the temperature inside the oven went down to the room temperature. The data evaluations will be introduced in the chapter 5.

5. RESULTS

The data collection method has been introduced in the last chapter. This chapter is devoted to implement the overlapping Allan Variance method to evaluate the data collected in different temperature points both statistically and graphically. Moreover, for the cooling down process, a 3D plot of ADEV over temperature change is applied to illustrate the AVAR change over temperature. Last but not least, a simple temperature compensation model is applied to find differences.

5.1 Raw Data

As last Chapter described, the data file includes both accelerometers and gyroscopes. The unit of accelerometer output is recorded as (m/s^2) and the unit of gyroscope output is saved as (rad/s).

An example of accelerometer raw data at 34°C is shown in Figure 5.1. The Z-axis of accelerometer should be aligned parallel with the direction of gravity under ideal condition. Due to this, all two horizontal axes outputs of the accelerometer are ideally expected to be zero and the vertical axis is ideally expected to be around $9.8m/s^2$ due to the effect of gravity. However, in our case, the chip is slightly tilted and the x-axis and y-axis are therefore affected by the gravity force as well. Hence, the x-axis is around $0.5m/s^2$, y-axis is around $0.2m/s^2$ and z-axis is around $9.5m/s^2$. From the plot of raw data, it also illustrates that the z-axis output is noisier than the other two axes, which therefore will be expected to have larger standard deviation and Allan Deviation.

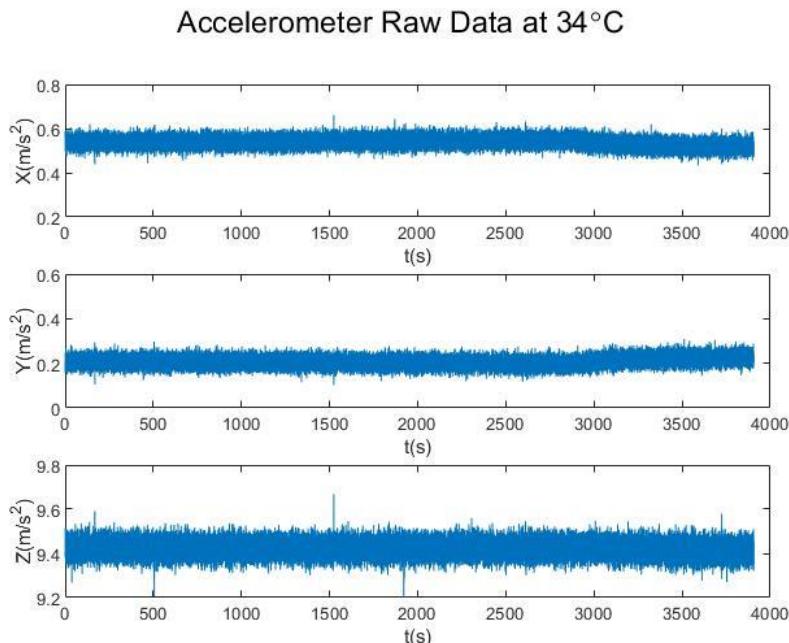


Figure 5.1: An example of accelerometer raw data

An example of gyroscope raw data at 34°C is shown in Figure 5.2. The output of gyroscope is expected to fluctuate around zero under ideal condition. However, there are biases exist in the output data.

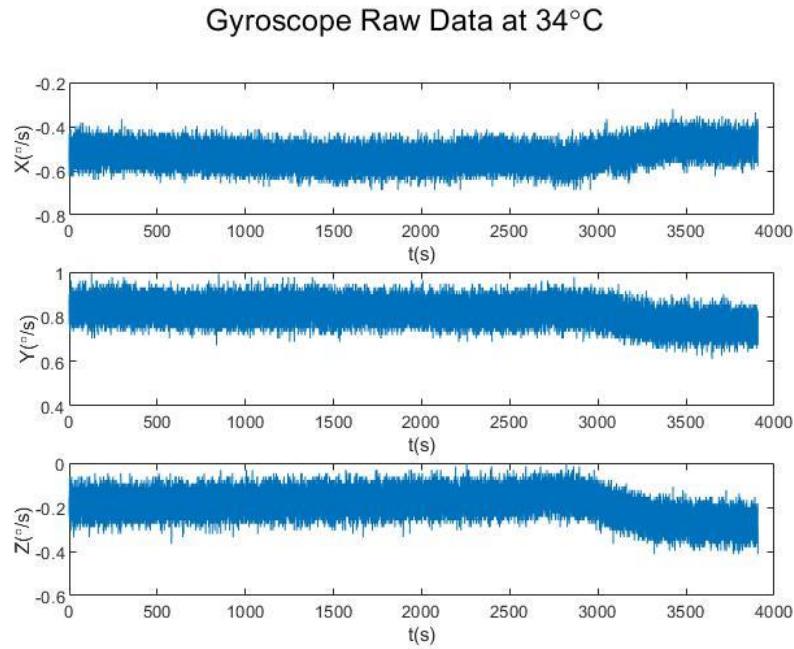


Figure 5.2: An example of gyroscope raw data

5.2 Allan Deviation Plots

The methodology to generate Allan Deviation plots has been introduced in chapter 3. The Allan Deviation plots for accelerometer and gyroscope at 34°C, 43°C, 62°C, 82°C are shown in Figure 5.3 ~ Figure 5.6.

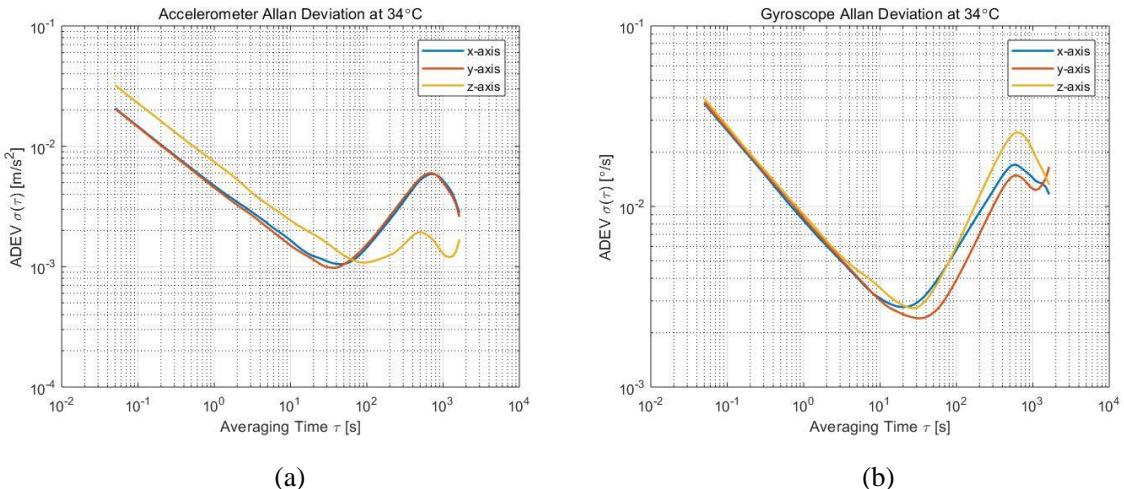


Figure 5.3: ADEV plot at 34°C (a) Accelerometer (b) Gyroscope

MASTER THESIS: TEMPERATURE DEPENDENCY OF A LOW-COST IMU

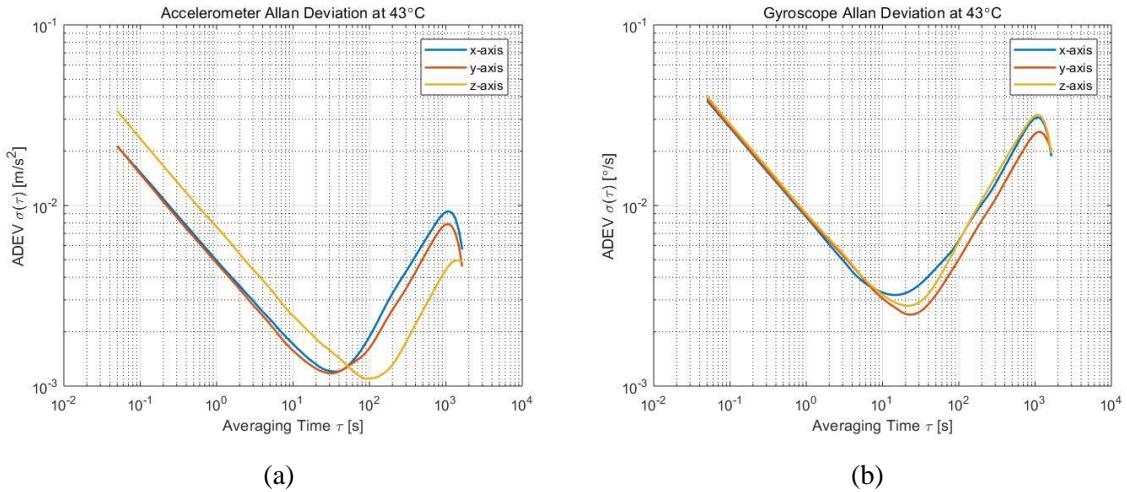


Figure 5.4: ADEV plot at 43°C (a) Accelerometer (b) Gyroscope

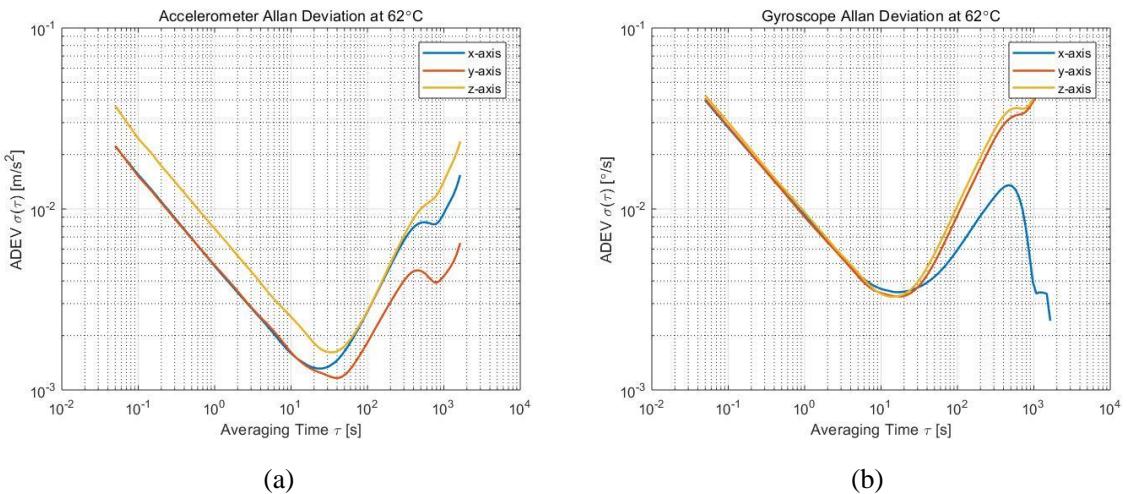


Figure 5.5: ADEV plot at 62°C (a) Accelerometer (b) Gyroscope

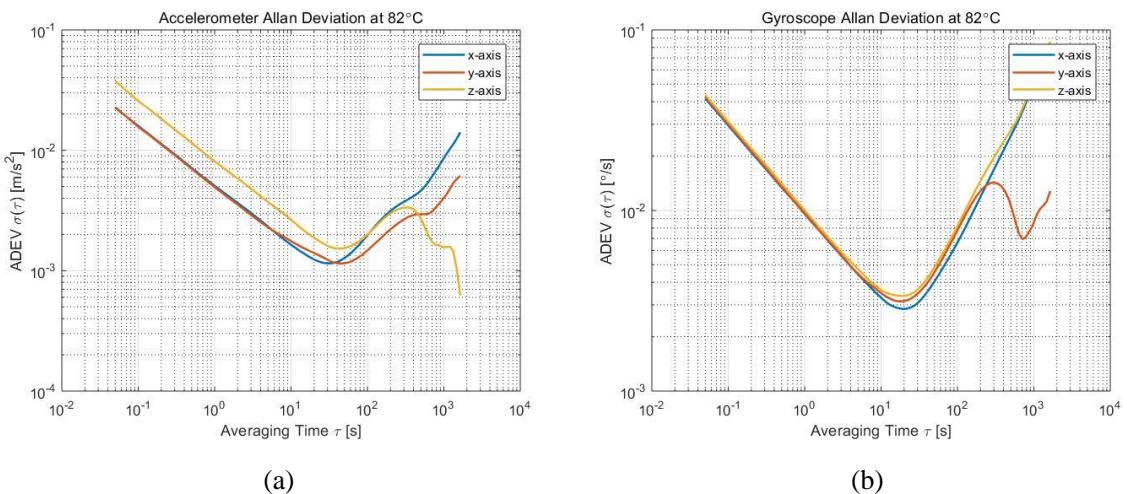


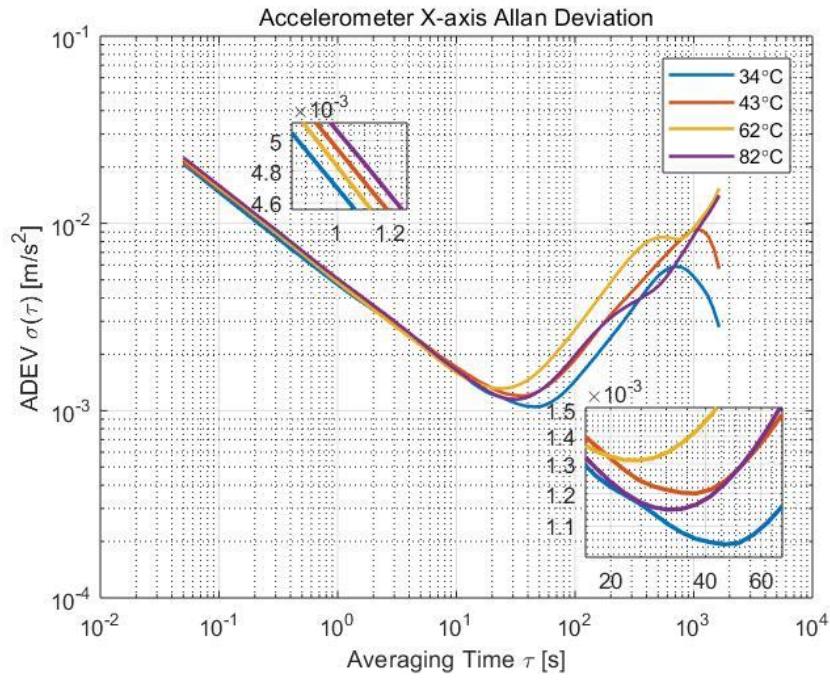
Figure 5.6: ADEV plot at 82°C (a) Accelerometer (b) Gyroscope

From figures shown above, the z-axis of accelerometer is significantly worse than the other two axes, this is due to the manufacturing process. As described in the previous chapter, the structure of z-axis is not as stable and accurate as the other two axes in plane. The capacitance

changes between proof mass and electrodes are not as sensitive as the other in-plane axes. Moreover, the shapes of ADEV at different temperature are different.

In order to find the effect of temperature, accelerometer and gyroscope ADEV plots at different temperature in the same plot are generated simply shown in Figure 5.7 and Figure 5.8, which includes all 6 axes of IMU. Generally, when we are going to evaluate the IMU performance through Allan Deviation plot, the lower the curve position is, the better performance it has.

From the plot of accelerometer's X-axis, we can infer that 82°C is above all the other curves at $\tau = 1\text{s}$, which indicates that the largest VRW it has. In our case, we can also infer that the VRW values follow the sequence of 82°C > 43°C > 62°C > 34°C. As for the bottom of the curve, which is related to bias instability, we can infer that the bias instability values follow the sequence of 62°C > 43°C > 82°C > 34°C. In the plot of accelerometer's Y-axis, the VRW follows the same sequence of 82°C > 62°C > 43°C > 34°C, the bias instability follows the sequence of 43°C > 62°C > 82°C > 34°C. In the plot of accelerometer's Z-axis, the VRW follows the same sequence as Y-axis, the bias instability follows the sequence of 62°C > 82°C > 43°C > 34°C. All in all, the graphical analyzation in our case shows that the lower temperature is, the lower VRW value may have. The bias instability value is not well regulated, but 34°C always have smallest value through all three axes of accelerometer. The accelerometer of MPU-9250 has the best performance at 34°C among all our experiments.



(a)

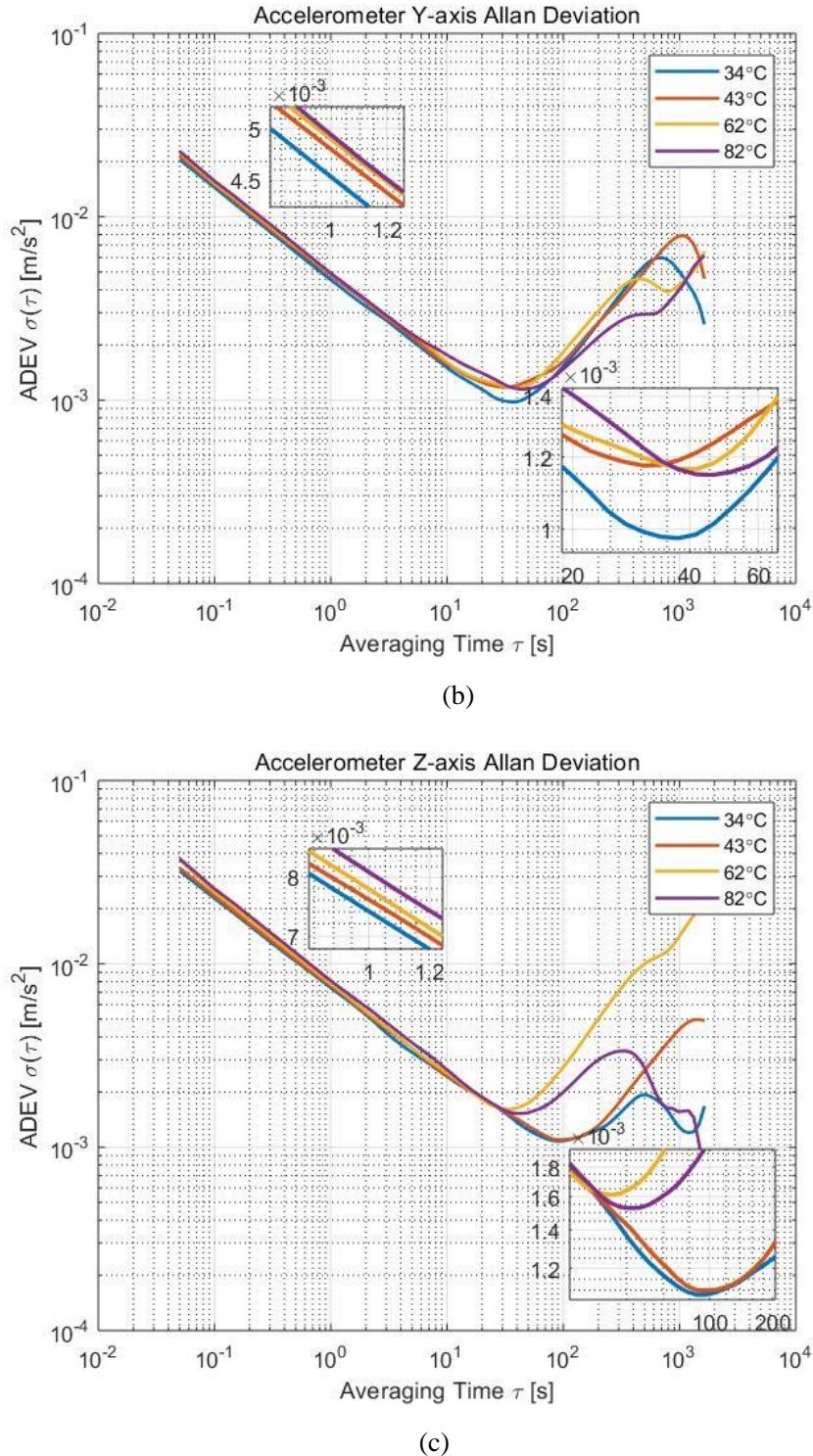
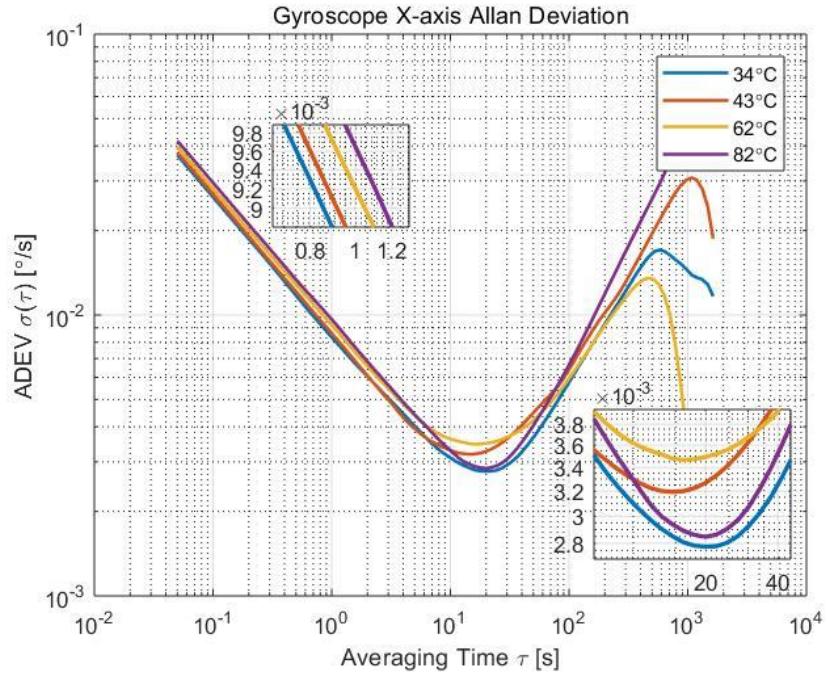


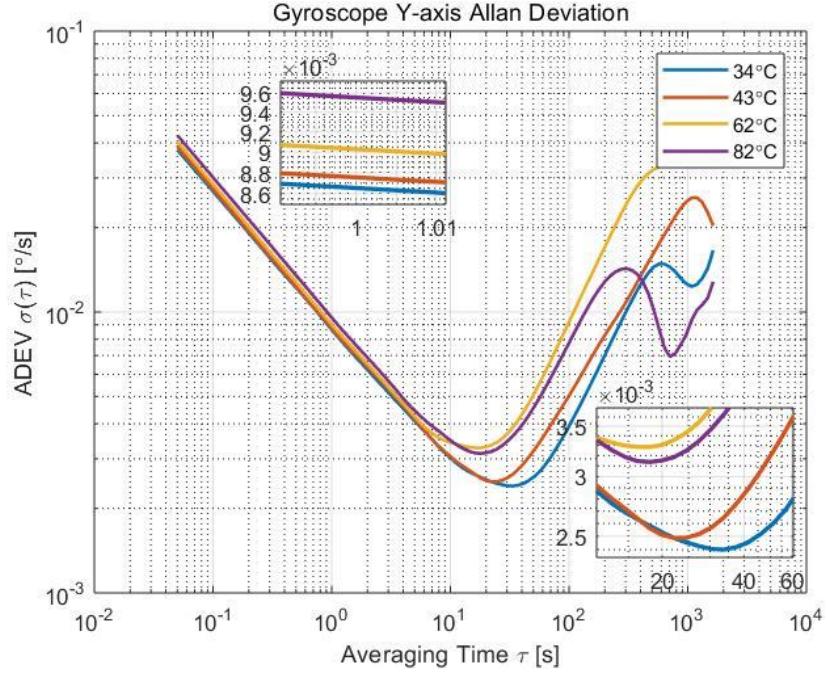
Figure 5.7: Accelerometer ADEV plots at different temperature (a) X-axis (b) Y-axis (c) Z-axis

From the plot of gyroscope Allan Deviation plot, we can also do the same analyzation as accelerometer did. In the plot of gyroscope's X-axis, the ARW follows the sequence of $82^{\circ}\text{C} > 62^{\circ}\text{C} > 43^{\circ}\text{C} > 34^{\circ}\text{C}$, the bias instability follows the sequence of $62^{\circ}\text{C} > 43^{\circ}\text{C} > 82^{\circ}\text{C} > 34^{\circ}\text{C}$. In the plot of gyroscope's Y-axis, the ARW follows the sequence of $82^{\circ}\text{C} > 62^{\circ}\text{C} > 43^{\circ}\text{C} > 34^{\circ}\text{C}$, the bias instability follows the sequence of $62^{\circ}\text{C} > 82^{\circ}\text{C} > 43^{\circ}\text{C} > 34^{\circ}\text{C}$. In the plot of gyroscope's Z-axis, the ARW follows the sequence of $82^{\circ}\text{C} > 62^{\circ}\text{C} > 43^{\circ}\text{C} > 34^{\circ}\text{C}$, the bias instability follows the sequence of $82^{\circ}\text{C} > 62^{\circ}\text{C} > 43^{\circ}\text{C} > 34^{\circ}\text{C}$. All in all, the information shows that the lower temperature in our case is, the smaller ARW has. The bias instability is not well regulated,

but the lower temperature has the tendency to have smaller bias instability. 34°C has both smallest ARW and bias instability in our case, which indicates that the gyroscope of MPU-9250 performs best at the temperature of 34°C among all tested temperature.



(a)



(b)

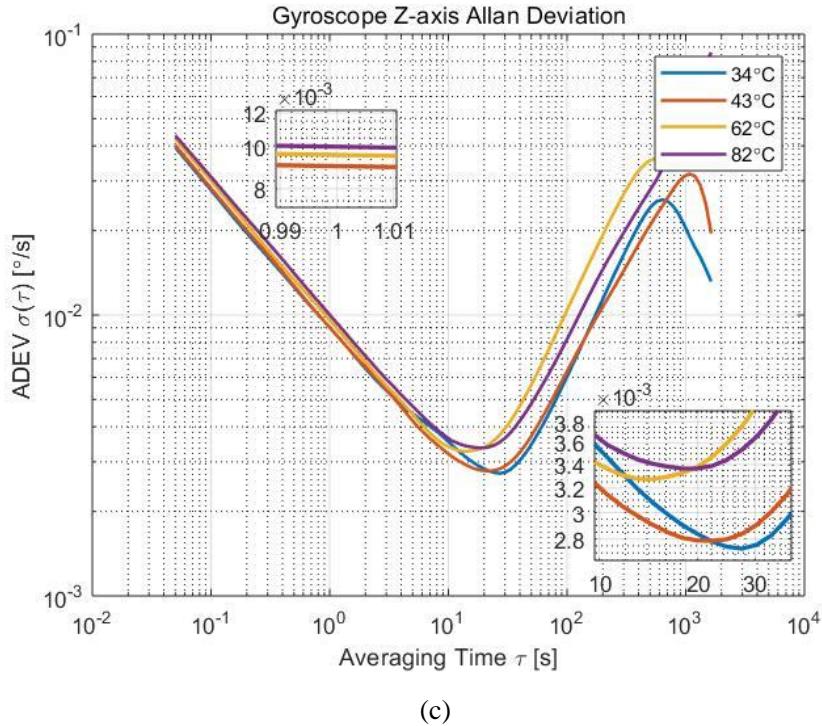


Figure 5.8: Gyroscope ADEV plots at different temperature (a) X-axis (b) Y-axis (c) Z-axis

5.3 Noise Analysis

As stated in Chapter 3, Allan Deviation also enable us to determine different terms of noise in the data, including quantization noise, angle/velocity random walk (ARW/VRW), bias instability (BI), rate random walk and rate ramp. ARW/VRW and BI are more frequently used for MEMS sensor. In section 5.3, the values of ARW/VRW and BI has been inferred through AEDV plots, however is not as accurate as statistical analysis. This section will characterize these two main noise terms statistically.

As Chapter 3 stated, different noise term has different value of slope in log-log plot of ADEV. The numerical value of ARW/VRW can be obtained directly by reading the slope of $-1/2$ line at $\tau = 1$. It is marked as N in the plot. The bias instability value is the ADEV at the region where the slope equals zero. The value of bias instability can be read directly off with a scaling of $\sqrt{\frac{2 \ln 2}{\pi}} \approx 0.664$. In our case, you may notice that some plot has more than one slope equals zero, hence, the program needs to find the first one that equals to zero.

An example of Allan Deviation with two noise parameters is shown in Figure 5.9 (a). It is an example of accelerometer's X-axis at 82°C. The Marker N is the value of VRW and the Marker 0.664B is the scaled value of bias instability. Another example shown in Figure 5.9 (b) is an example of gyroscope's X-axis at 82°C.

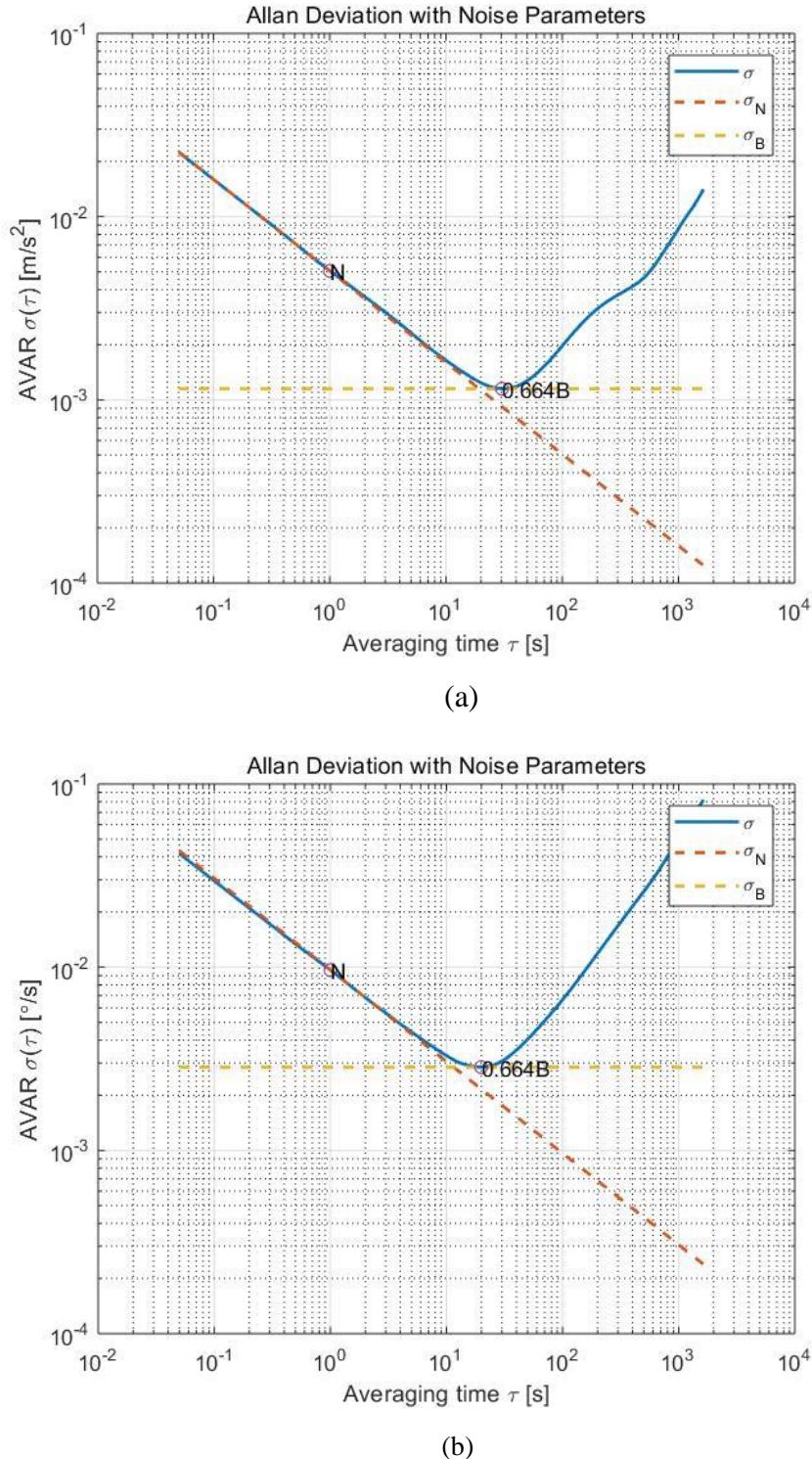


Figure 5.9: Examples of ADEV with noise parameters (a) Accelerometer (b) Gyroscope

5.3.1 Accelerometer noise analysis

By using the method stated before, the VRW and BI values at 34°C, 43°C, 62°C, 82°C are calculated and shown in Table 5.1 and Table 5.2. The statistical analysis is done by making plots of VRW and BI for different axis separately, which are shown as Figure 5.10. In our case, from the plots of VRW, it is evident that the VRW values of all three axes have the tendency to become bigger as the temperature rises. From the Table 5.2, the time when BI is reached can also be inferred. Basically, the later reaches bias instability point, the better the sensor is. In our

case, bias instability point of 34°C happens the latest or the second latest indicates that it has better performance than the others. As for the BI values, 34°C also keeps the smallest, 82°C is less than 62 °C. All in all, accelerometer of MPU-9250 has the best performance at the temperature of 34°C in our experiments. Moreover, it has the tendency to become worse when the temperature goes up too high. This can be accepted, while the MPU-9250 sensors are low-cost MEMS sensor designed for consumer market. But for the purpose of our experiment, it is obvious that the temperature does influences the performance of the accelerometer.

Table 5.1: Accelerometer Velocity Random Walk values

Velocity Random Walk ($m/s^2/\sqrt{s}$)				
	34°C	43°C	62°C	82°C
Accel X	0.0046	0.0049	0.0049	0.0051
Accel Y	0.0045	0.0049	0.0049	0.0050
Accel Z	0.0072	0.0076	0.0078	0.0083

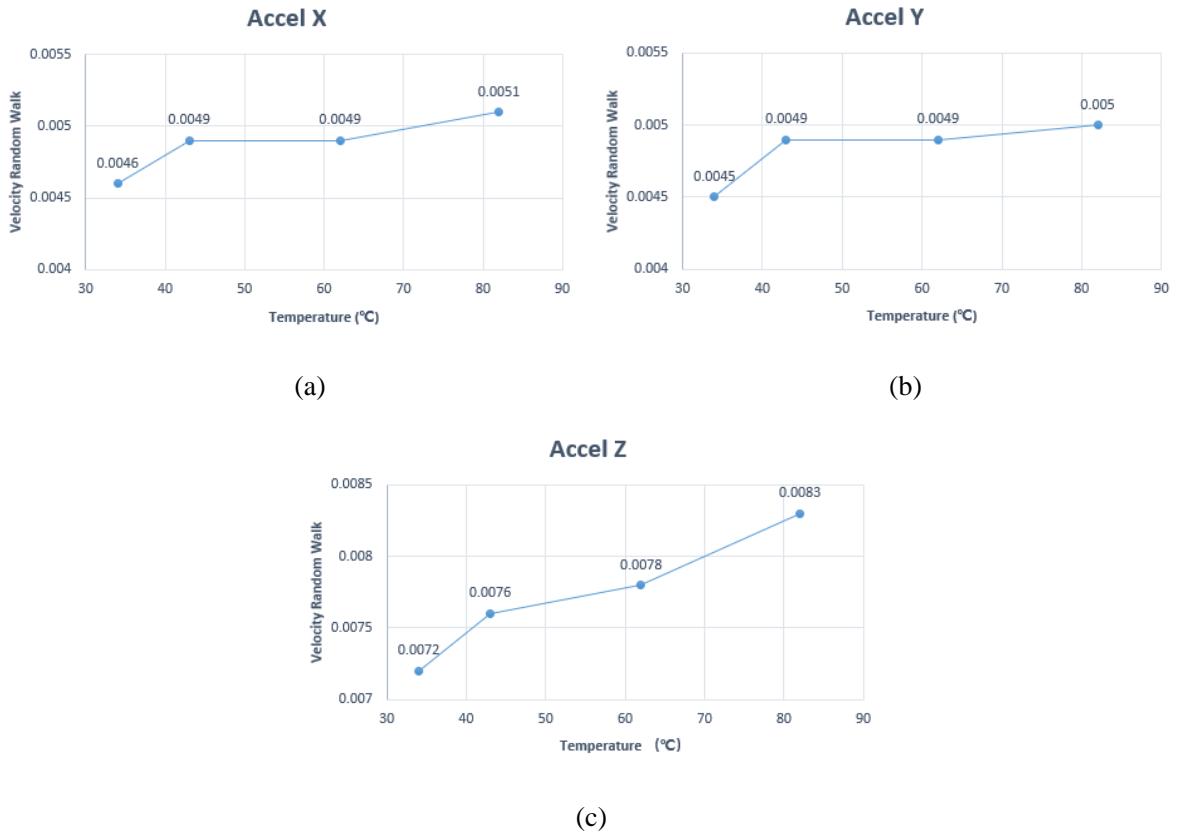


Figure 5.10: Accelerometer VRW values plot (a) X-axis (b) Y-axis (c) Z-axis

Table 5.2: Accelerometer Bias Instability values

	Bias Instability (m/s^2)			
	34°C	43°C	62°C	82°C
Accel X	0.0016 (at 46.1s)	0.0018 (at 37.4s)	0.0020 (at 24.6s)	0.0017 (at 30.3s)
Accel Y	0.0015 (at 37.4s)	0.0018 (at 30.3s)	0.0018 (at 41.6s)	0.0017 (at 46.1s)
Accel Z	0.0016 (at 86.6s)	0.0017 (at 96.2s)	0.0024 (at 33.7s)	0.0023 (at 41.6s)

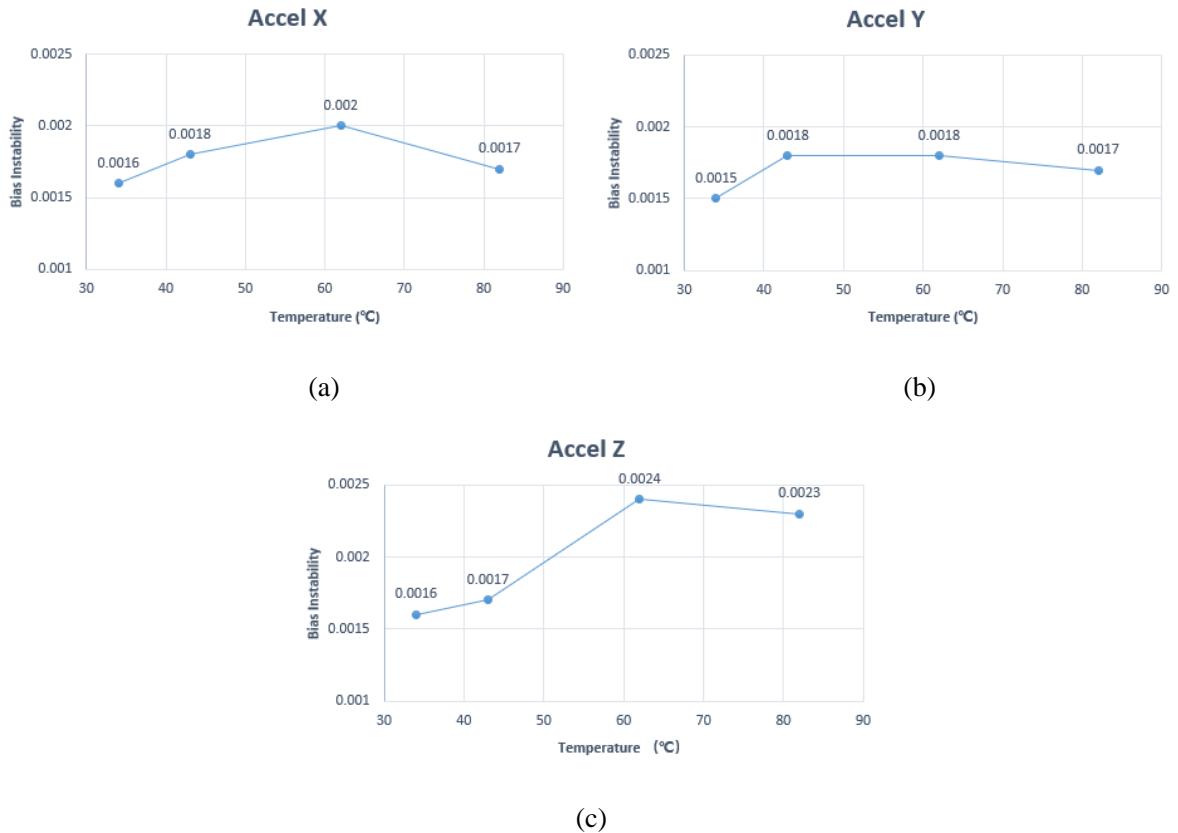


Figure 5.11: Accelerometer Bias Instability (a) X-axis (b) Y-axis (c) Z-axis

5.3.2 Gyroscope noise analysis

The noise analysis of gyroscope is similar to accelerometer. The ARW and BI values at different temperature in our experiment are shown in Table 5.3 and Table 5.4. Figure 5.12 and Figure 5.13 are generated from Table 5.3 and Table 5.4. From Figure 5.12, it is also obvious that the ARW value goes up as the temperature rises, which has the same behavior as accelerometer. It also appears that the BI point of 34°C happens later than the other datasets. 34°C has the smallest value of both ARW and BI, which also illustrates that the gyroscope at 34°C performs the best among all datasets. In this case, gyroscope holds the same response to temperature as accelerometer does.

Table 5.3: Gyroscope Angle Radom Walk values

	Angle Random Walk ($^{\circ}/\sqrt{s}$)			
	34°C	43°C	62°C	82°C
Gyro X	0.0084	0.0086	0.0092	0.0097
Gyro Y	0.0087	0.0087	0.0090	0.0096
Gyro Z	0.0088	0.0094	0.0096	0.0100

MASTER THESIS: TEMPERATURE DEPENDENCY OF A LOW-COST IMU

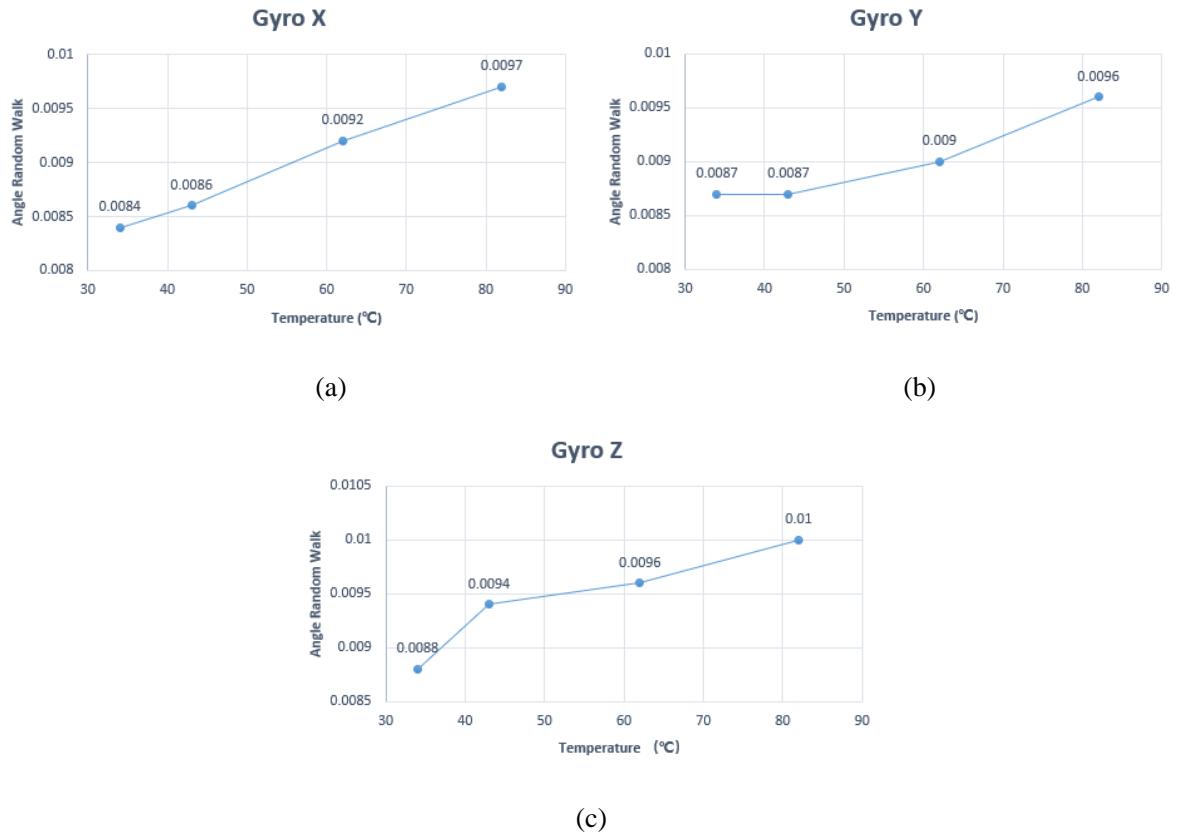
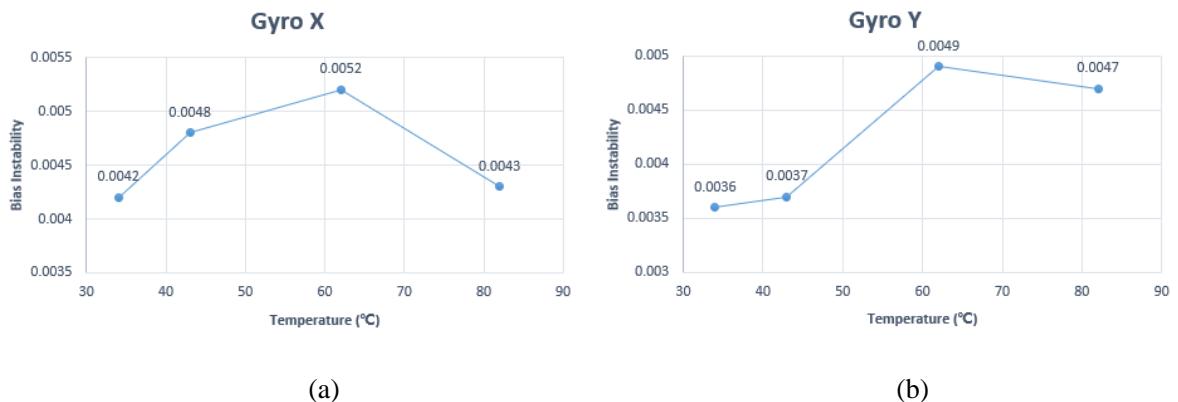
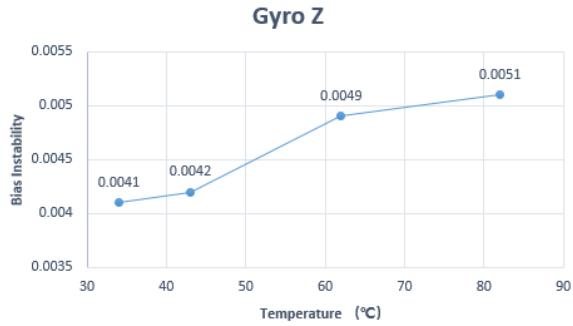


Figure 5.12: Gyroscope Angle Random Walk (a) X-axis (b) Y-axis (c) Z-axis

Table 5.4: Gyroscope Bias Instability values

Bias Instability (°/s)				
	34°C	43°C	62°C	82°C
Gyro X	0.0042 (at 19.9s)	0.0048 (at 14.6s)	0.0052 (at 16.2s)	0.0043 (at 19.9s)
Gyro Y	0.0036 (at 33.7s)	0.0037 (at 22.1s)	0.0049 (at 18.0s)	0.0047 (at 18.0s)
Gyro Z	0.0041 (at 27.3s)	0.0042 (at 22.1s)	0.0049 (at 14.6s)	0.0051 (at 19.9s)





(c)

Figure 5.13: Gyroscope Bias Instability (a) X-axis (b) Y-axis (c) Z-axis

5.4 Cooling Down Process analysis

As stated in chapter 4, in order to find how the performance of MPU-9250 reacts to dynamic change of temperature, a cooling down process was performed as well. The chip was firstly heated up to a high temperature by the oven and then kept collecting data until the temperature on the chip reached the room temperature, which took approximately 6 hours. The temperature change is characterized in Figure 5.14, which is non-linear. The temperature goes down fast at relatively high temperature, and goes down slower at low temperature part. The temperature range is approximately from 100°C to 20°C.

Figure 5.15 are plots of accelerometer's and gyroscope's raw data of cooling down process. It is obvious that these raw data of different axes are slightly changed over time. The temperature influences the data readout obviously.

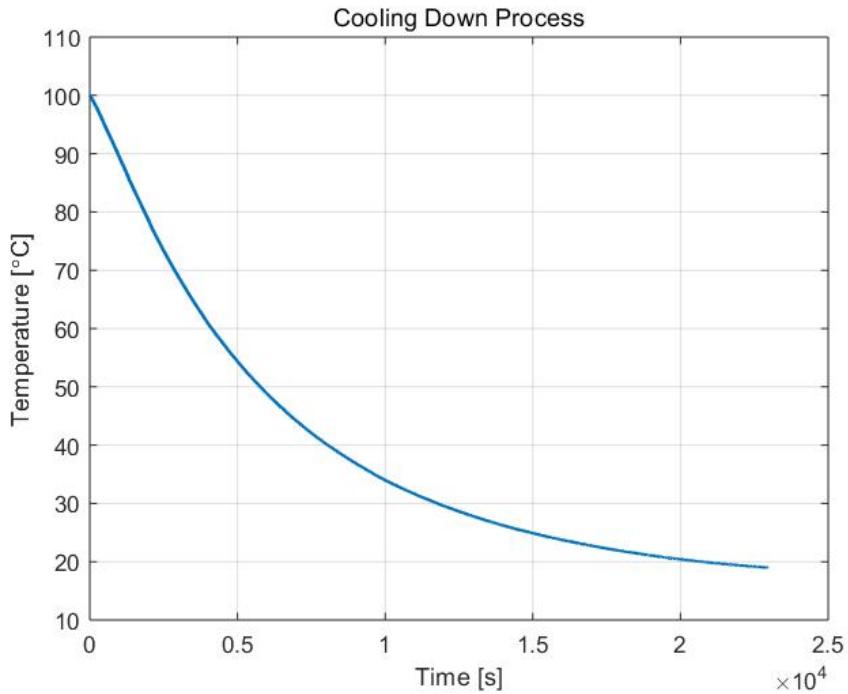


Figure 5.14: Cooling down process temperature change

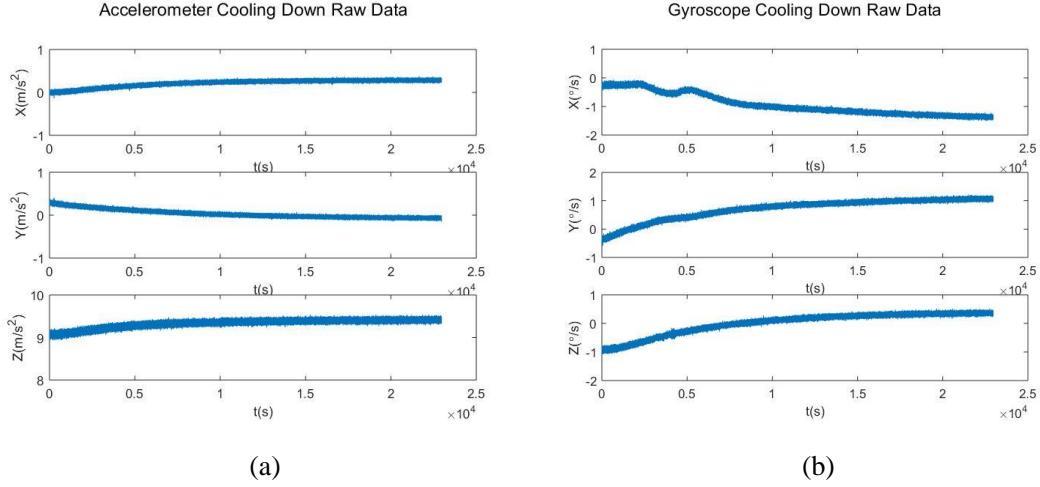
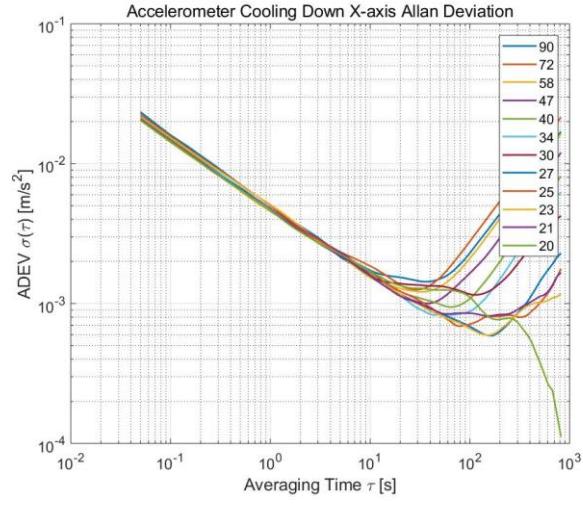


Figure 5.15: Cooling down process raw data (a) Accelerometer (b) Gyroscope

Only one Allan Deviation is not able to reveal the noise change over temperature change. In this case, the dataset is divided into several clusters. All clusters have 36000 samples, which is equivalent to half hour sampling and are enough to compute two noise terms mentioned before. The Allan plot in our case has 12 curves. ARW/VRW and BI values are also computed and plotted over temperature. Figure 5.16 ~ Figure 5.21 shown below are Allan Deviation plots of different axes of MPU-9250 and their ARW/VRW, BI values changing over temperature. From the plots below, we can find that in the temperature range experimented, both ARW/VRW and BI values hold the tendency to become smaller. There exist some fluctuations especially for accelerometer, for example, in Figure 5.16 (b). These may result from some environmental disturbance.



(a)

MASTER THESIS: TEMPERATURE DEPENDENCY OF A LOW-COST IMU

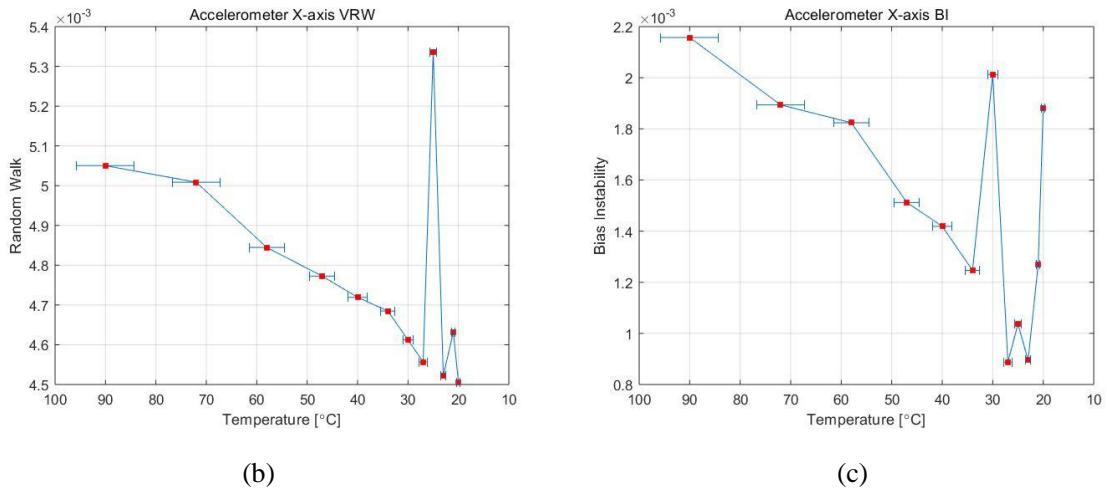


Figure 5.16: Cooling down process Accelerometer X-axis (a) ADEV (b) VRW (c) BI

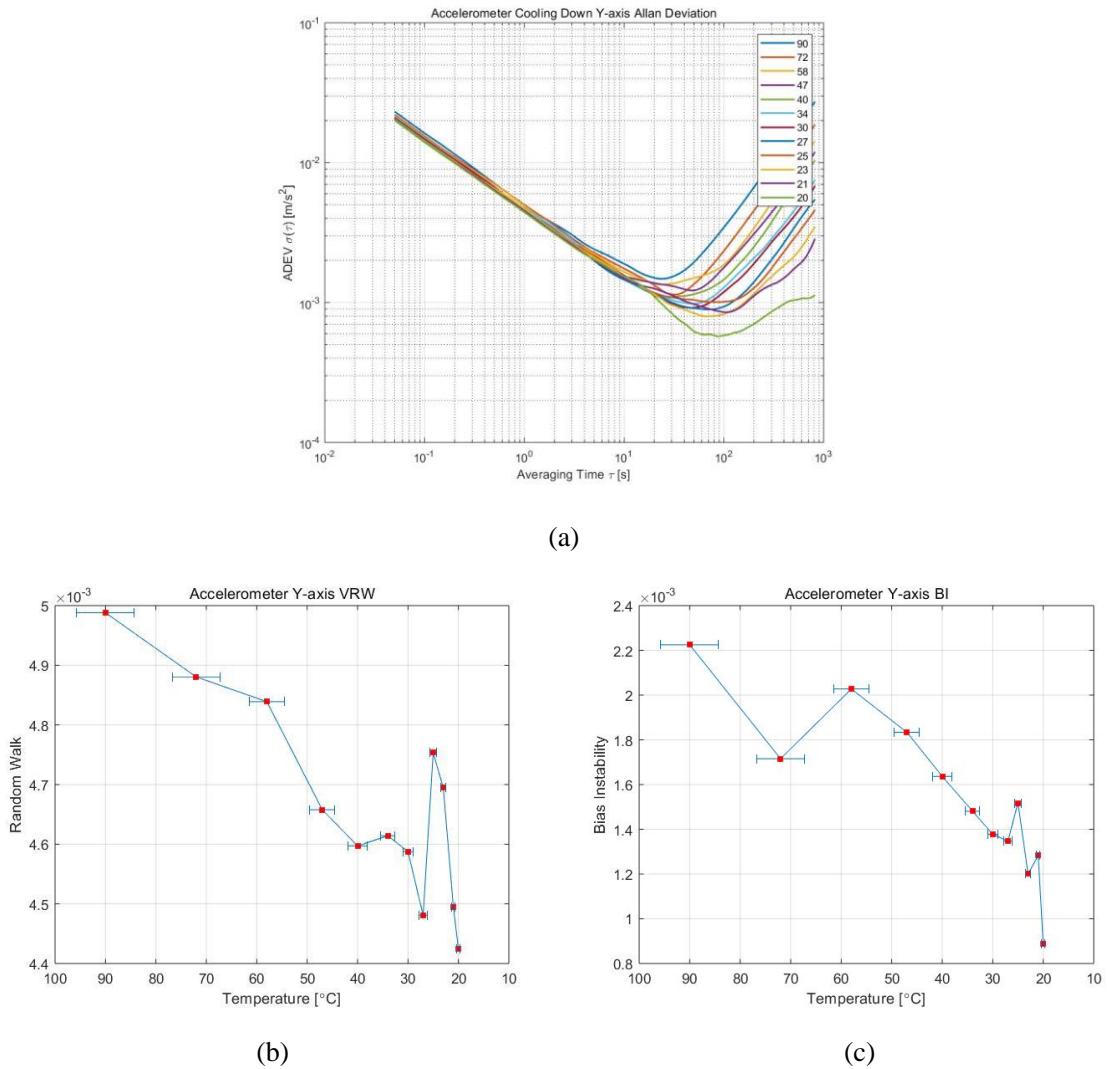


Figure 5.17: Cooling down process Accelerometer Y-axis (a) ADEV (b) VRW (c) BI

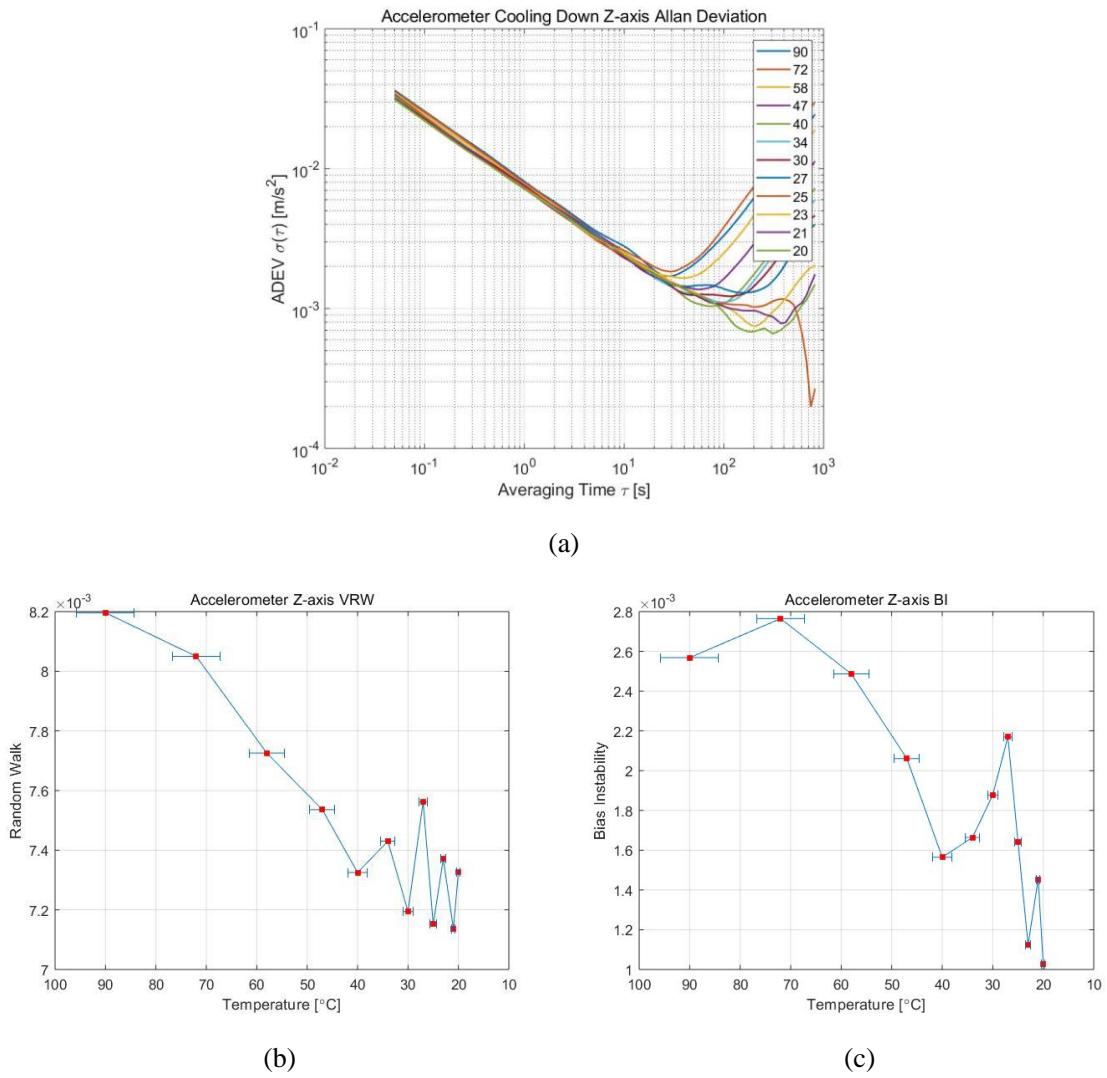
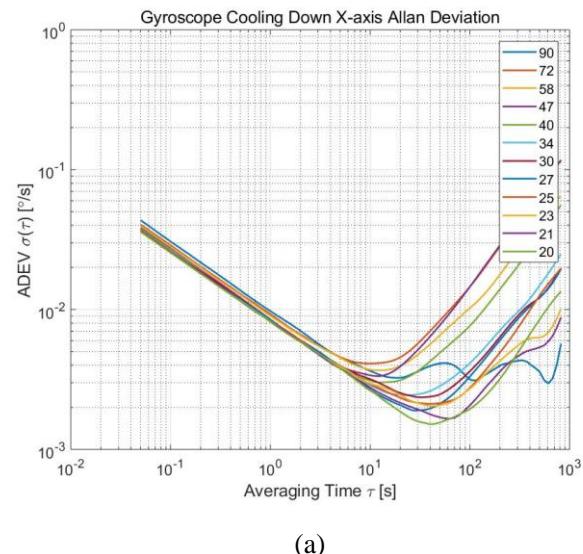


Figure 5.18: Cooling down process Accelerometer Z-axis (a) ADEV (b) VRW (c) BI



(a)

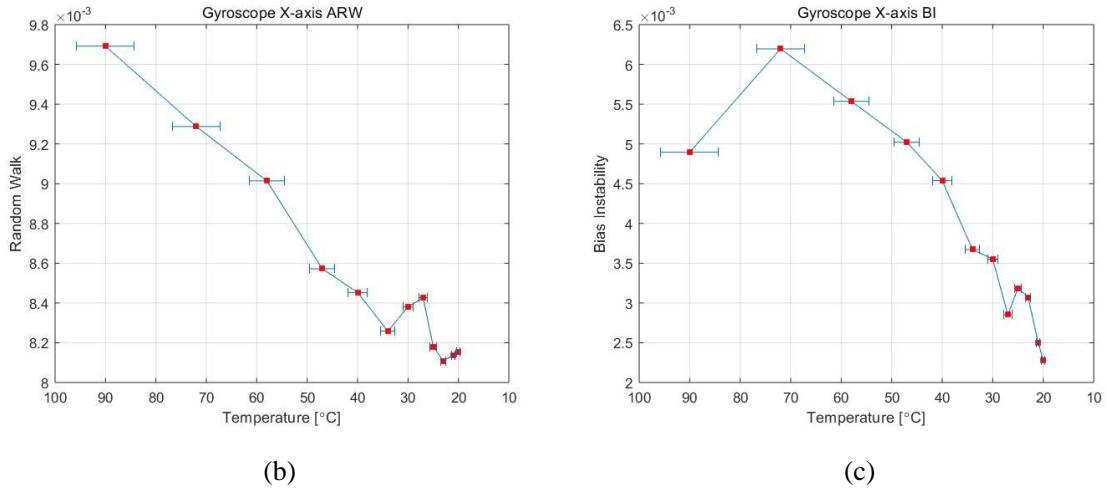


Figure 5.19: Cooling down process Gyroscope X-axis (a) ADEV (b) ARW (c) BI

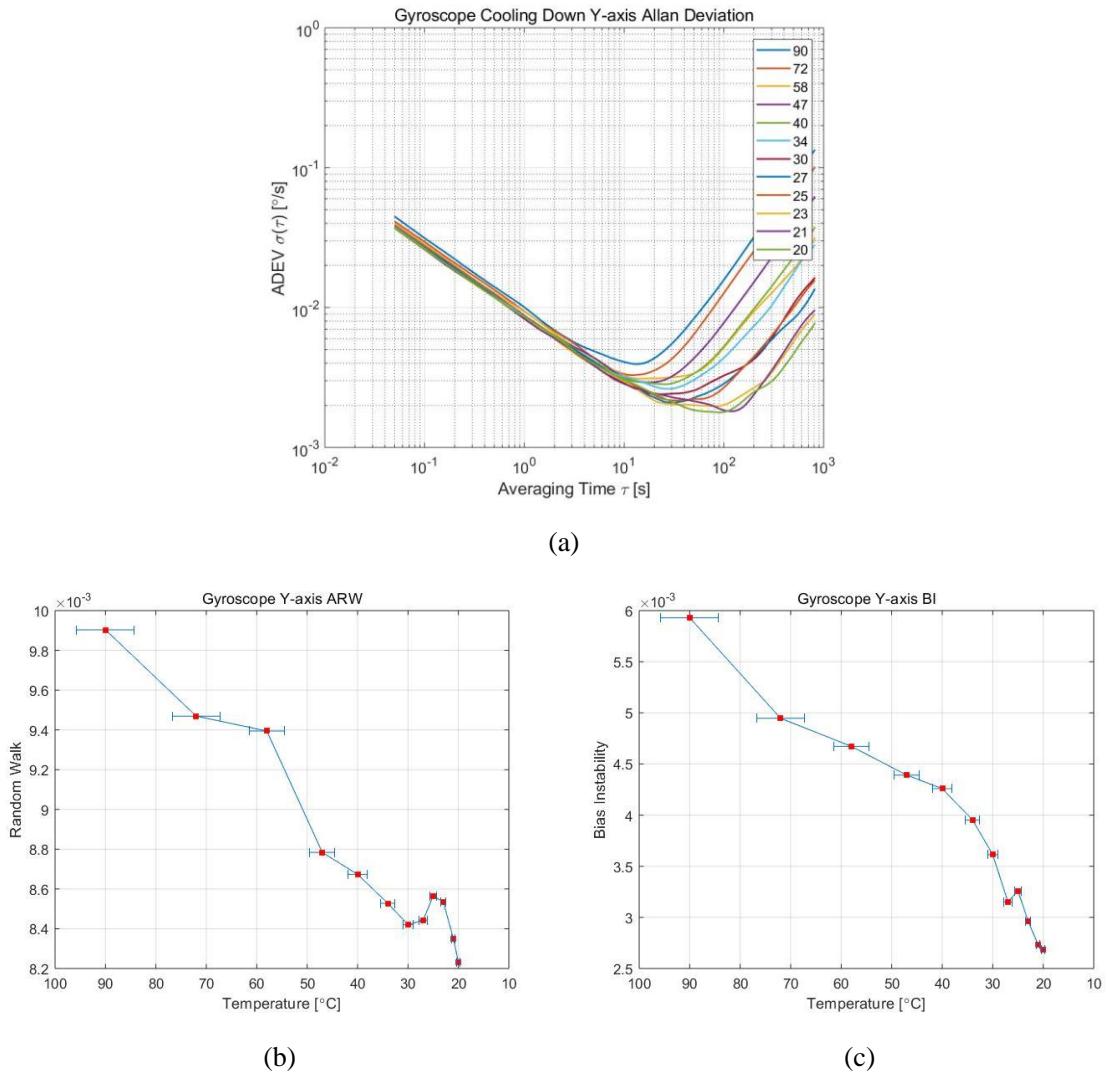


Figure 5.20: Cooling down process Gyroscope Y-axis (a) ADEV (b) ARW (c) BI

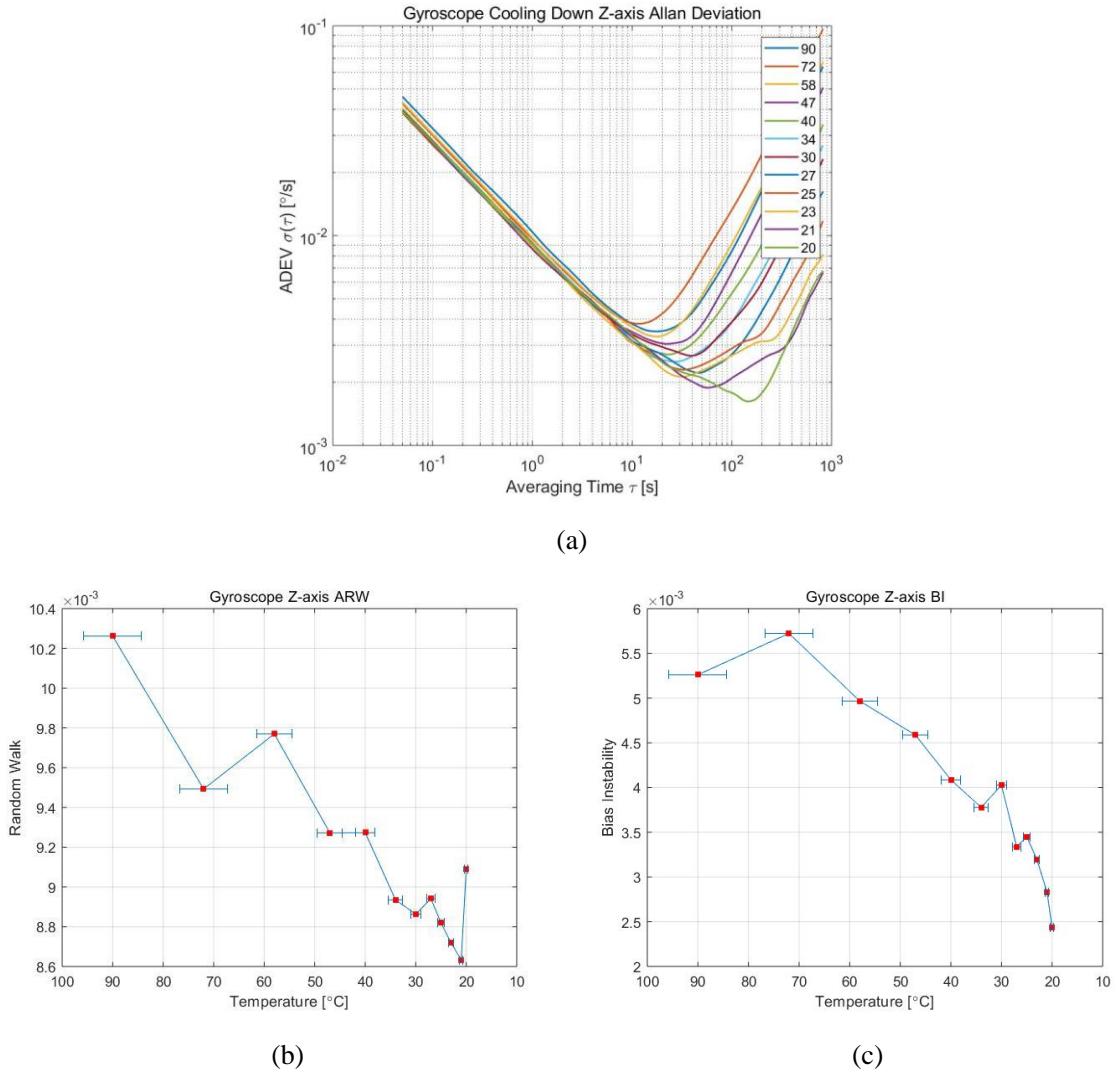
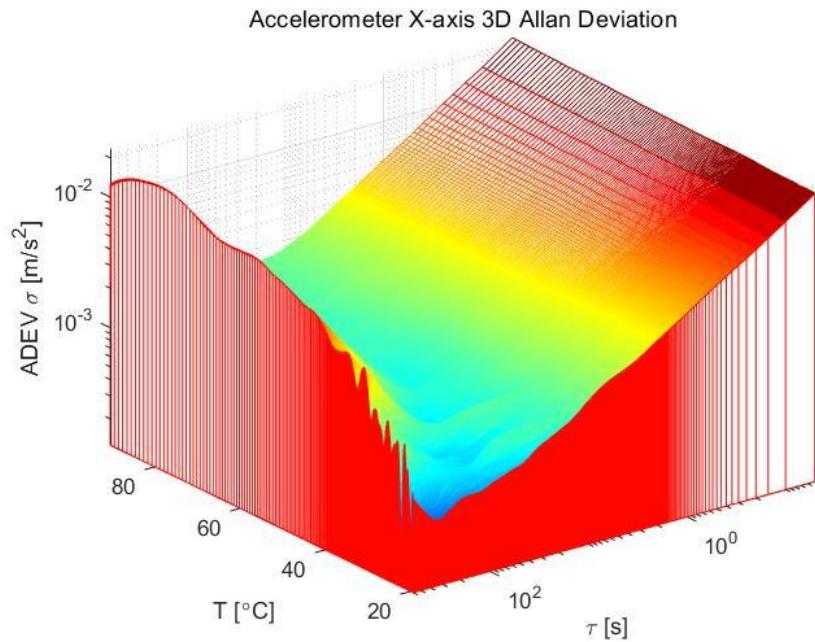


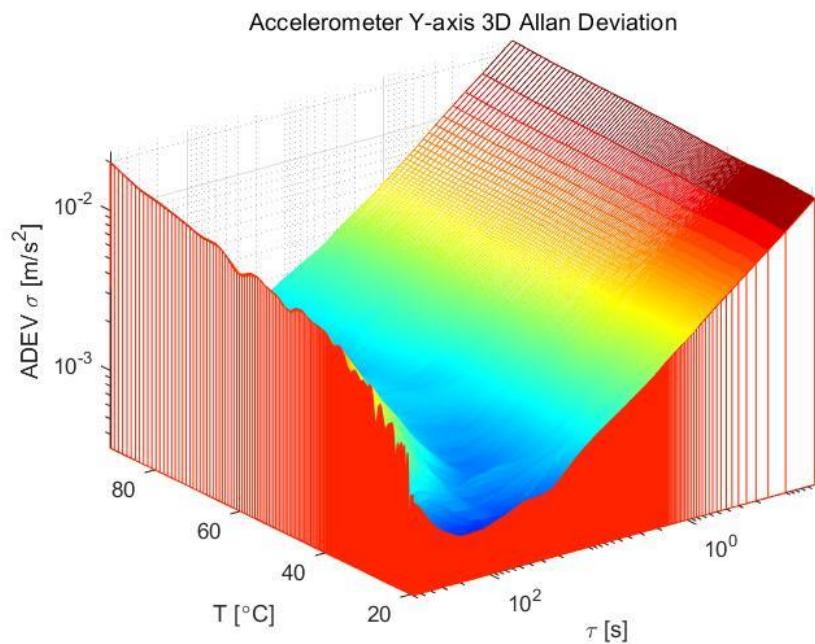
Figure 5.21: Cooling down process Gyroscope Z-axis (a) ADEV (b) ARW (c) BI

For the cooling down process, in order to find the dynamic change of Allan Deviation, 3D plots are generated as well. The 3D plots simply add averaging temperature as the third axis. In order increase the density of Allan Deviation, a moving window is applied, which includes 36000 samples in a window. The center of the window is defined, which depends on the length of input data. The step width in our plots are 1800, which means that the center of the window is defined every 1800 samples. In order to avoid zero padding at the beginning and the end of data, the center of the window starts with 36001 and end with (end - 36000). There's a method called Dynamic Allan Deviation (DAVAR), which can show the dynamic change of Allan Deviation over time. The third axis of Dynamic Allan Deviation is the actual measuring time. Here in our case, the time axis is taken place of averaging temperature of each cluster, which shows the dynamic performance change over temperature significantly.

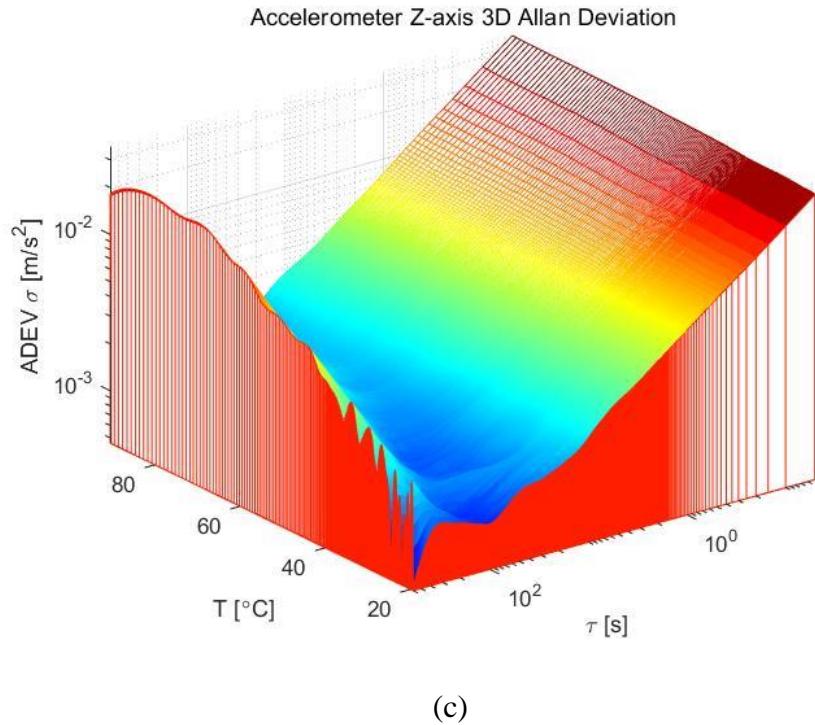
Figure 5.22 are examples of accelerometer's dynamic change of Allan Deviation over temperature. It is obvious that as the temperature goes down, the Allan Deviation curve goes down as well, the “valley” turns to become deeper as temperature decreases. The dynamic plots of Allan Deviation show the dynamic change of accelerometer's output significantly. In the cooling down experiment, the quality and stability of acquired data from accelerometer becomes better as temperature decreases from 100 $^{\circ}$ C to 20 $^{\circ}$ C, which means having smaller VRW and BI values.



(a)



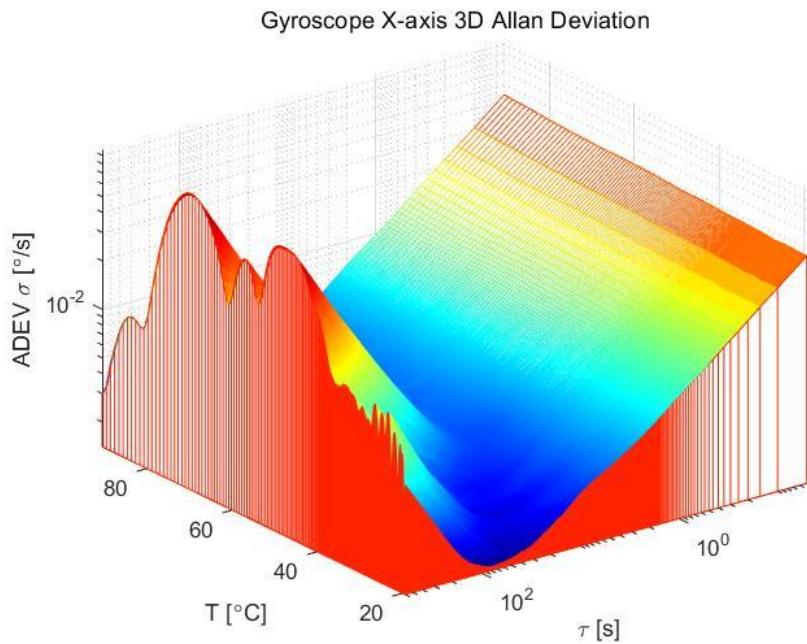
(b)



(c)

Figure 5.22: Accelerometer ADEV against temperature (a) X-axis (b) Y-axis (c) Z-axis

Figure 5.23 are examples of gyroscope's dynamic change of Allan Deviation over temperature. It shows that the gyroscope of MPU-9250 has the similar response to the temperature varying. The lower temperature is, the better performance it has. The X-axis of gyroscope is a little bit strange at high temperature, which can be interpreted from the raw data. In the raw data of X-axis, we can see a fluctuation at the beginning. This may due to some environmental disturbances. However, the whole tendency is the same as the other two axes. The “valley” turns to become deeper as well. All in all, the performance of gyroscope is getting better as the temperature goes down from 100°C to 20°C.



(a)

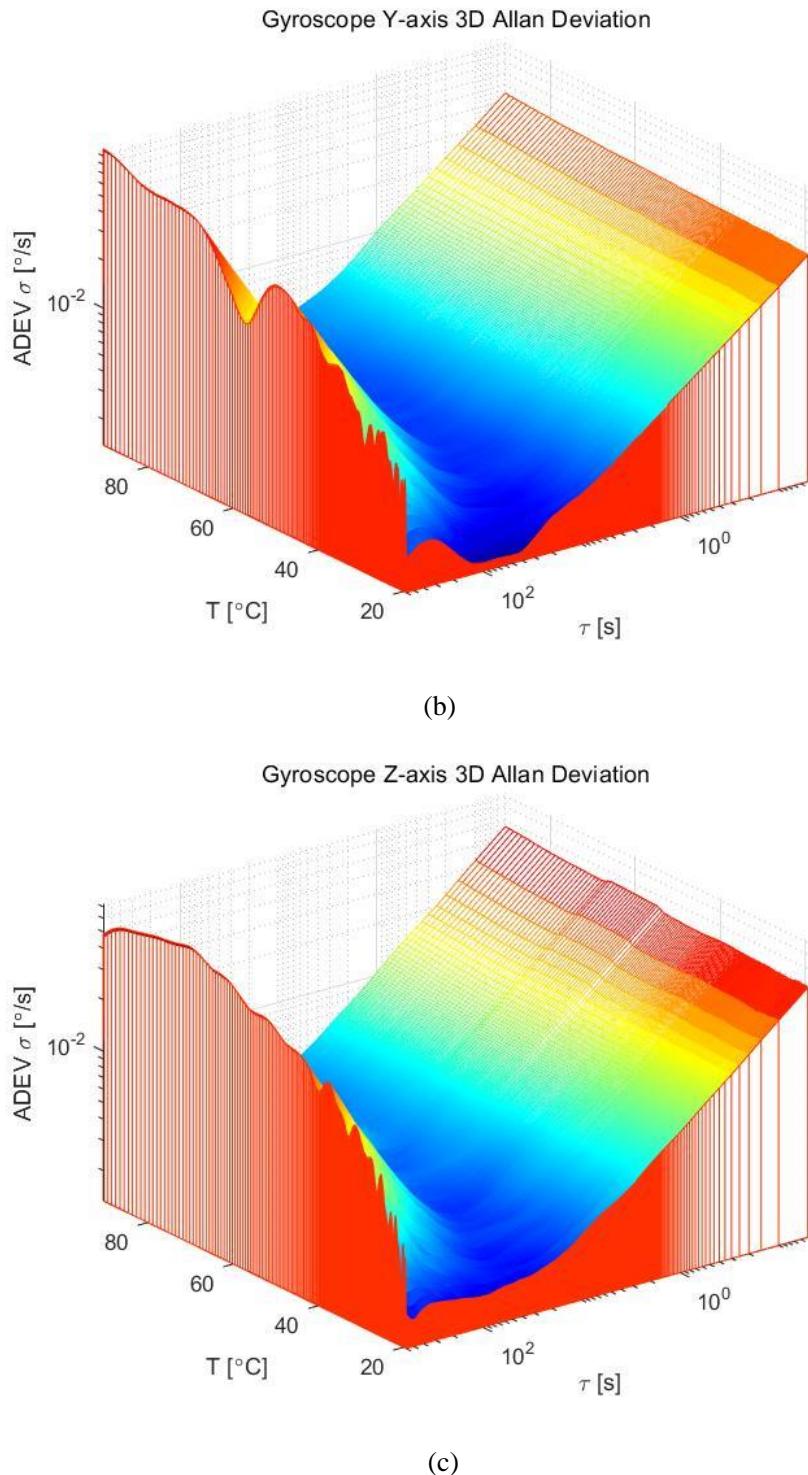


Figure 5.23: Gyroscope ADEV against temperature (a) X-axis (b) Y-axis (c) Z-axis

5.5 Temperature Compensation Test

Based on the previous analysis, the change of temperature influences the performance of MPU-9250 is significant. Hence, applying temperature compensation to the raw data is necessary. Generally, the temperature compensation process can be achieved by hardware design or by data postprocessing. Some of hardware manufacturers integrate a temperature compensation register on the PCB board, however, the MPU-9250 doesn't provide this option.

As a result, in our case, a simple data postprocessing method is applied for temperature compensation.

The relation between temperature and raw data is modelled through polynomial fitting. MATLAB provides a function $\text{polyfit}(x, y, n)$ that can model the relation. n is the order of polynomial. This function implements temperature as x and raw data as y and returns the coefficients p for a polynomial fitting. Fit values can be calculated by another MATLAB function $\text{polyval}(p, x)$, where p is the fitted coefficients and x is the temperature point.

The order of polynomial fitting can influence the performance of model and computational time as well. In this case, it is necessary to find a proper order to model the relation at first. The differences between raw data and fitted values are important to evaluate the performance of different orders. The mean and standard deviation of these differences are calculated as two factors to evaluate, which are shown in Figure 5.24 and Figure 5.25. In these plots, the 2nd order, the 3rd order, the 4th order and the 5th order polynomials have approximately same mean error values, but the 4th order has smaller standard deviation than the 2nd order and the 3rd order polynomials and also less computational effort than the 5th order polynomial fitting. Finally, the 4th order polynomial fitting is chosen for our temperature compensation.

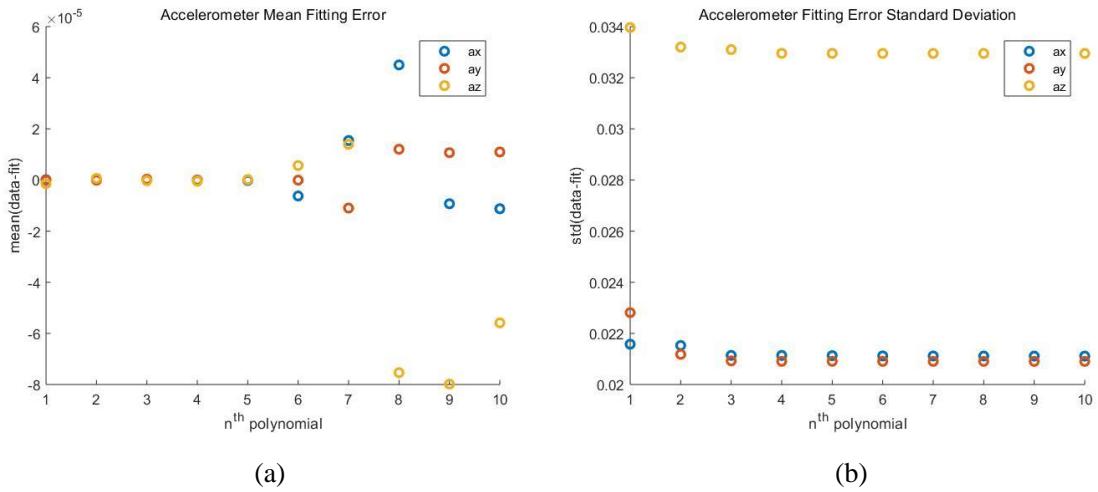


Figure 5.24: Accelerometer polynomial fitting evaluation (a) mean error (b) error standard deviation

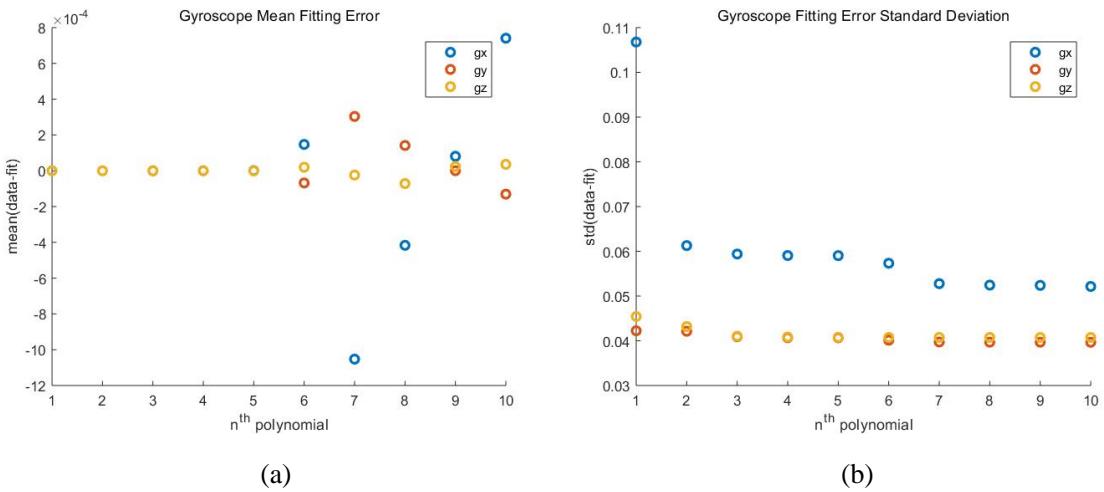


Figure 5.25: Gyroscope polynomial fitting evaluation (a) mean error (b) error standard deviation

The relation between temperature and output data is modelled with 4th polynomial in our case. Next step is to assume a temperature point that has no bias and get the fit value as “true

value". Based on the previous analysis, 20°C is assumed as a temperature point without any bias. Then calculate the differences between fit values of the other temperature points and fit value at 20°C. Finally, subtracting these differences from raw data and get the compensated data. Figure 5.26 shows the temperature compensated data and uncompensated data in the same plot. When comparing with raw data, it is obvious that temperature compensated data show a relatively horizontal line as expected.

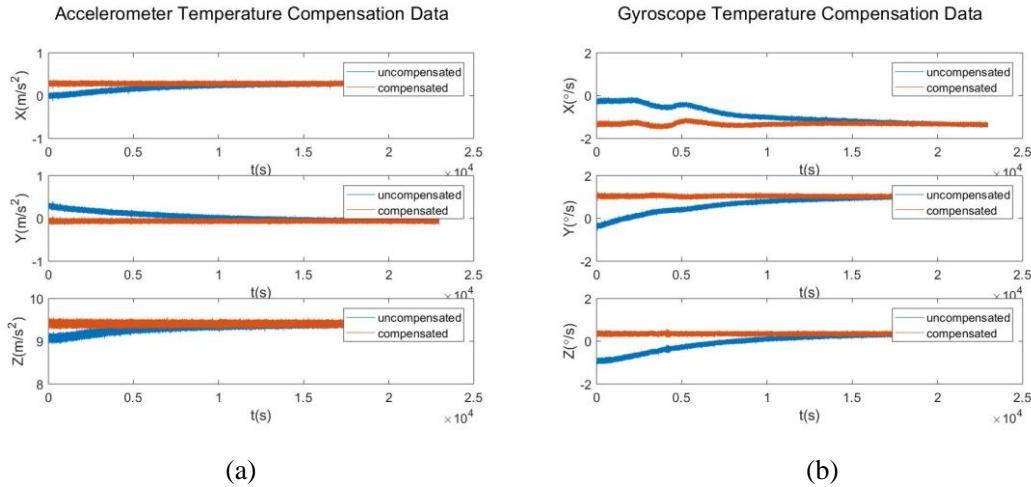
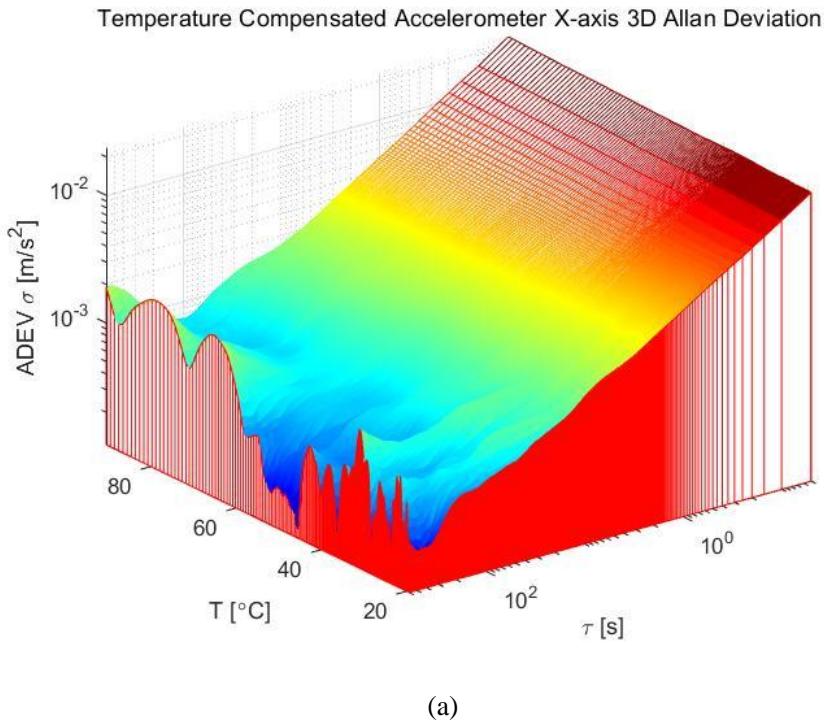
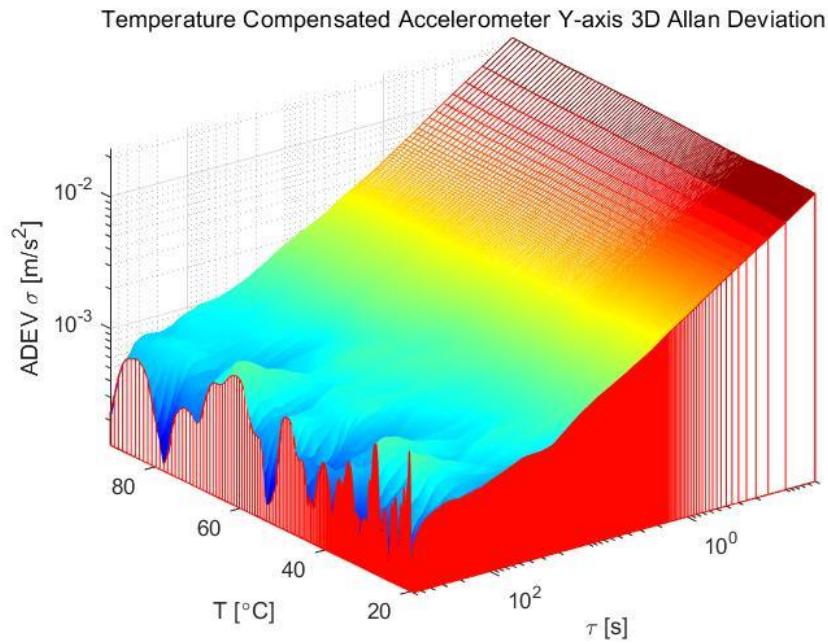


Figure 5.26: Compensated and uncompensated data (a) accelerometer (b) gyroscope

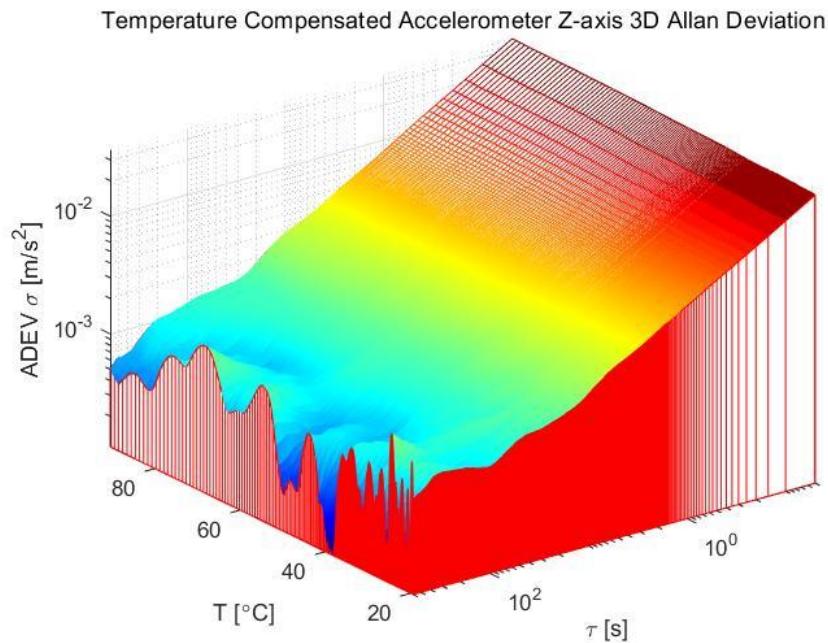
Figure 5.27 and Figure 5.28 shows the dynamic Allan Deviation over temperature for the temperature compensated data. The effect of temperature compensation for the high temperature is very obvious. The Allan Deviation values for high temperature decreases significantly when comparing with Figure 5.22 and Figure 5.23. Moreover, the effect of temperature compensation on accelerometer is more significant than on gyroscope when comparing Figure 5.27 and Figure 5.28.



(a)

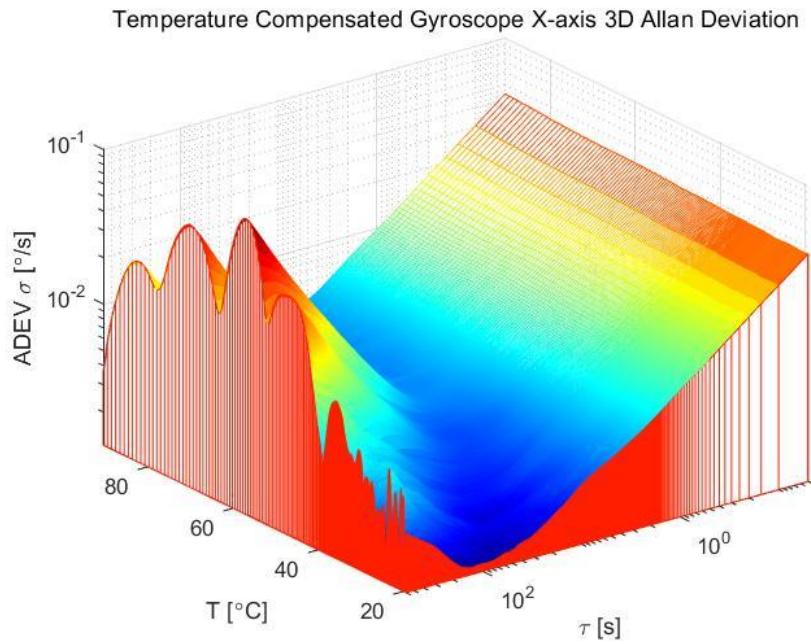


(b)

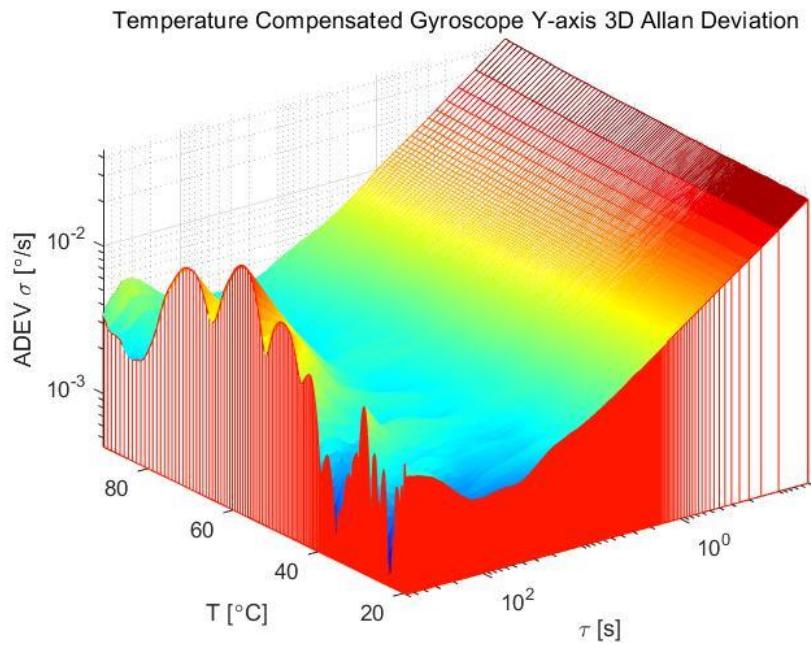


(c)

Figure 5.27: Compensated Accelerometer ADEV against temperature (a) X-axis (b) Y-axis (c) Z-axis



(a)



(b)

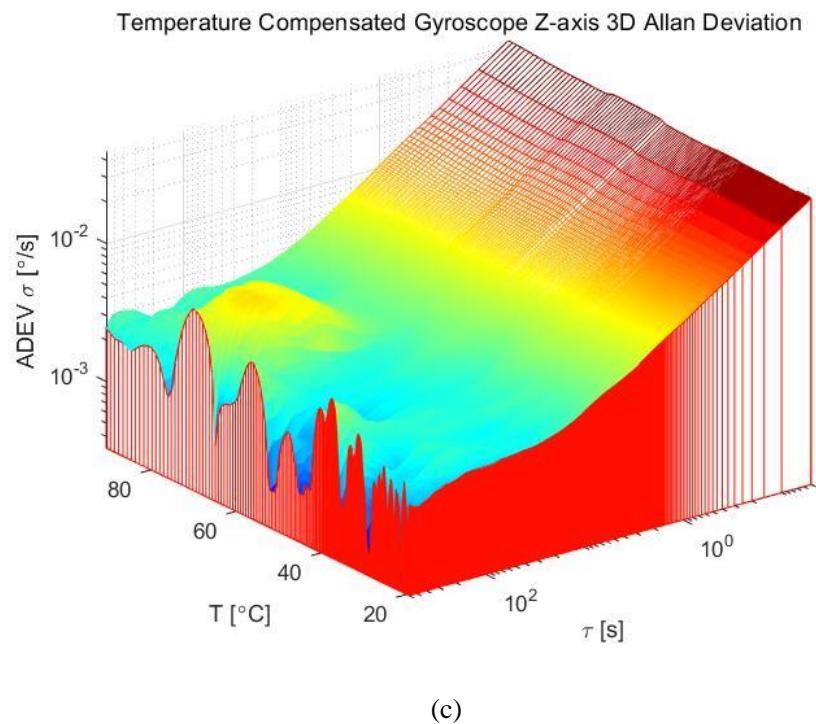


Figure 5.28: Compensated Gyroscope ADEV against temperature (a) X-axis (b) Y-axis (c) Z-axis

6. CONCLUSION

This master thesis works with MPU-9250, which is a very popular low-cost MEMS chip board IMU. Thanks to the MEMS technology, it is able to be designed and manufactured with advantages of cheap price, small size, light weight and so on. It can be integrated in many moving platforms, which require position estimation. However, MEMS IMUs are still not as accurate as the traditional IMUs, there are many factors influencing the performance of MEMS IMUs. Since temperature is one of the most important factors, this master thesis is dedicated to studying the effect of temperature on a low-cost MEMS IMU.

To achieve this purpose, a time domain stability analysis method called Allan Variance (AVAR) has been used in this thesis. AVAR is a method that can characterize and identify the noises of sensors. It can not only describe the noises of the sensor graphically, but also estimate several important noise parameters of the sensor. The most frequently used noise terms are ARW/VRW and BI, which are evaluated in this thesis. In order to evaluate the noise terms of MPU-9250 changing over temperature, two ways of data collection were implemented.

Firstly, four datasets are collected at temperature of 34°C, 43°C, 62°C, 82°C, each for 1h. The data analysis is performed by using AVAR method. It is obvious that 34°C has smaller ARW/VRW and BI values than the other. For ARW/VRW values, it appears that the value increases as temperature goes up. For the BI values of accelerometer and gyroscope, there exist some fluctuations, however, the tendencies are the same.

Secondly, the temperature was heated up to 100°C, which is a little bit beyond the official working temperature, then a cooling down process was performed after the oven was turned off. The cooling down process lasted for approximately 6h. Finally, the MPU-9250 experienced a non-linear temperature decreasing until 20°C. Here, not only 2D ADEV were used for noises evaluation, but also a 3D ADEV were plotted to show the noises change dynamically. Finally, it can also come to a conclusion that the noise terms are tend to decrease as temperature goes down from 100 °C to 20°C.

Finally, according to the previous analysis, 20°C is chosen as the temperature point that MPU-9250 has no bias. A simple temperature compensation method, which is based on polynomial curve fitting, is applied on the 6h dataset. The effect of temperature compensation for high temperature is significant.

Last but not least, the official working temperature of MPU-9250 could be -40°C, this master thesis only tested relatively high temperature, due to the fact that the oven provided is only able to heat up. It would be interesting to test in the cold environment in the future work. Moreover, this thesis only implements a simple and widely used temperature compensation model, more different types of temperature compensation models can be tested in the future work to find better temperature compensation model.

BIBLIOGRAPHY

- [1] ANDERSON, E.W. (1966): The principles of navigation, Hollis and Carter.
- [2] Sobel, Dava. (2004): Longitude: The true story of a lone genius who solved the greatest scientific problem of his time, Academy of Management Learning and Education 3, pp 220-220.
- [3] Morse, Walter Priest. (1953): An Analysis of the Development of Celestial Navigation, Diss. University of Florida.
- [4] Wikipedia contributors. (2020): Inertial Navigation System. Wikipedia, The Free Encyclopedia. URL: <https://en.wikipedia.org/wiki/Inertial_navigation_system> [accessed 13 January 2020]
- [5] Titterton, David H., John L. Weston. (2004): Strapdown Inertial Navigation Technology. 2nd, The Institution of Electrical Engineers.
- [6] ETHW contributors. (2020): MEMS. Engineering and Technology History Wiki, 4 April 2017. URL: <<https://ethw.org/MEMS>> [accessed 13 January 2020]
- [7] O. J.Woodman. (2007): An introduction to inertial navigation. University of Cambridge, UCAM-CL-TR-696, ISSN 1476-2986, Tech. Rep.
- [8] Beeby, S., Ensell, G., Kraft, M., White, N. (2004): MEMS Mechanical Sensors, Artech House, Norwood.
- [9] Acar, C., Shkel, A.S. (2008): MEMS Vibratory Gyroscopes: Structural Approaches to Improve Robustness, Springer, New York.
- [10] Antonello, Riccardo, Roberto Oboe. (2011): MEMS gyroscopes for consumer and industrial applications, InTech.
- [11] Greiff, P., et al. (1991): Silicon monolithic micromechanical gyroscope, TRANSDUCERS'91: 1991 International Conference on Solid-State Sensors and Actuators. Digest of Technical Papers. IEEE.
- [12] Bernstein, Jon, et al. (1993): A micromachined comb-drive tuning fork rate gyroscope, Proceedings IEEE Micro Electro Mechanical Systems. IEEE.
- [13] Armenise, Mario N., et al. (2010): Advances in gyroscope technologies, Springer Science & Business Media.
- [14] Lopez-Hignera, J-M., Miguel A. Morante, and Adolfo Cobo. (1997): Simple low-frequency optical fiber accelerometer with large rotating machine monitoring applications, Journal of lightwave technology 15.7, pp 1120-1130.
- [15] Pupo, Leslie Barreda. (2016): Characterization of errors and noises in MEMS inertial sensors using Allan variance method, MS thesis.
- [16] Mohammed, Zakriya, Ibrahim Elfadel, and Mahmoud Rasras. (2018): Monolithic multi degree of freedom (MDoF) capacitive MEMS accelerometers, Micromachines 9.11, pp 602.
- [17] Stockwell, Walter. (2003): Angle random walk, Application Note. Crossbow Technologies Inc, pp 1-4. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.210.1133&rep=rep1&type=pdf>
- [18] Stockwell, Walter. (2004): Bias stability measurement: Allan variance, Crossbow Technology, Inc., Tech. Rep. URL: <http://file.elecfans.com/web1/M00/57/5F/pIYBAFtGgDeADLPaAAy7GZFlc0M782.pdf>
- [19] E. Foxlin. (2002): Handbook of Virtual Environment Technologies, chapter Motion Tracking Technologies and Requirements, Lawrence Erlbaum Publishers, pages 163–210.
- [20] Allan, David W. (1966): Statistics of atomic frequency standards, Proceedings of the IEEE 54.2, pp 221-230.
- [21] IEEE Std. 1293 (1998): IEEE Standard Specification Format Guide and Test Procedure for Linear, Single-Axis, Non-gyroscopic Accelerometers, IEEE.
- [22] Hou, Haiying, and Naser El-Sheimy. (2001): Inertial sensors errors modeling using Allan variance, Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003).
- [23] Riley, William J. (2008): Handbook of frequency stability analysis, NIST, pp 81.
- [24] Variance, Variance A. Allan (2015): Noise Analysis for Gyroscopes, Freescale Semiconductor Application Note AN5087.
- [25] IEEE Std. 952 (1997): IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros, IEEE.

MASTER THESIS: TEMPERATURE DEPENDENCY OF A LOW-COST IMU

- [26] InvenSense Inc. (2016): MPU-9250 Product Specification Revision 1.1.
URL: <https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>
- [27] InvenSense Inc. (2015): MPU-9250 Register Map and Descriptions Revision 1.6.
URL: <https://www.invensense.com/wp-content/uploads/2015/02/RM-MPU-9250A-00-v1.6.pdf>

APPENDIX A

Allan Deviation MATLAB Code

allan.m

```

function [tau, adev] = allan(omega, t0)

% This function is to compute for a single axis allan deviation and
% averaging time tau values.

% Inputs:
%     omega: rate output
%     t0:    time interval

% Outputs:
%     tau:    averaging time [s]
%     adev:   overlapping allan deviation value

theta = cumsum(omega, 1)*t0;
maxNumM = 100;
L = size(theta, 1);
maxM = 2.^floor(log2(L/2));
m = logspace(log10(1), log10(maxM), maxNumM).';
m = ceil(m); % m must be an integer.
m = unique(m); % Remove duplicates.
tau = m*t0;
avar = zeros(numel(m), 1);
for i = 1:numel(m)
    mi = m(i);
    avar(i, :) = sum((theta(1+2*mi:L) - 2*theta(1+mi:L-mi) + theta(1:L-2*mi)).^2, 1);
end
avar = avar ./ (2*tau.^2 .* (L - 2*m));
adev = sqrt(avar);
end

```

APPENDIX B

Polynomial Temperature Compensation MATLAB Code

temp_comp.m

```
function [x_compensated] = temp_comp(temp, x, n, T_NB)
% This function is designed for n-th order polynomial temperature
% compensation for MPU-9250
% Inputs:
%     temp:    temperature vector data
%     x:       raw data
%     n:       n-th polynomial fitting
%     T_NB:   temperature assumed to have no bias
% Outputs:
%     x_compensated: temperature compensated data
p = double(polyfit(temp, x, n));
x_compensated = ax - (polyval(p, temp)-polyval(p, T_NB));
end
```