University of Toronto

Masters Thesis

MEMS Accelerometer Specifications and Their Impact in Inertial Applications

by

Kei-Ming Kwong

Supervisor

Prof. David A. Johns

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science, Graduate Department of Department of Electrical and Computer Engineering, in the University of Toronto

University of Toronto

Abstract

Faculty of Applied Science and Engineering

Department of Electrical and Computer Engineering

Master of Applied Science

MEMS Accelerometer Specifications and Their Impact in Inertial Applications

by Kei-Ming Kwong

Recent development of microelectromechanical systems (MEMS) accelerometers improved their performance. Coupled with their benefits of lower cost and smaller size, enabled their increased utilization in navigation, automotive and consumer devices. However, specification and testing methodologies of these devices are not robustly defined. This work investigates and defines a set of testing methodology for MEMS accelerometers, making use of a 3D printer based testing platform and a scalable inertial sensor testing board. Specification results show that Kionix KXRB5 and Invensense MPU6000 perform the best of the devices tested. Furthermore, commonly used inertial algorithms were applied to study the impact of accelerometer choice in an inertial navigation system (INS). Across a attitude estimation and dead reckoning tests, results indicate that noise density has little impact on performance after inertial algorithms are applied. Cross-axis, bias variability and step motion specification results are better indicators of performance after inertial algorithms are applied.

Acknowledgements

I would like to thank my parents and sister for their continuous support throughout my years in undergraduate and graduate school. Your support and encouragement gave me the motivation to learn new things and pursue my interests.

I would like to thank my supervisor, Prof. David A. Johns, for providing guidance and advice throughout the thesis and for all his insights during our weekly discussions throughout the project. I would like to thank you for your approach to the project, giving me a high degree of freedom and expression in this project. Finally, I would also like to thank you for giving me this opportunity to pursue this project, it was a humbling and extremely rewarding experience.

I would also like to thank Peter Timmermans and Alon Green for your insight during the early stages of the project which helped shaped different aspects of the project. I learned a lot in regards to the considerations needed for transportation purposes.

Finally, I would also like to thank Wahid Rahman, for giving me an outlet to discuss my ideas with.

Contents

List of Figures

List of Tables

Abbreviations

ADC	Analog Digital Converter
API	Application Programming Interface
AVAR	Allan variance
DR	Dynamic Range
DUT	Device under test
FFT	Fast Fourier Transform
GPS	Global Positioning System
IC	Integrated chip
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
NIST	National Institute of Standards and Technology
PCB	Printer Circuit Board
SNDR	Signal to Noise and Distortion Ratio
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface

Chapter 1 Introduction and Study Organization

1.1 Introduction

Accelerometers are inertial sensors used to measure forces, and subsequently acceleration, across three orthogonal axes; the measurements are output as a digital or analog signal and further utilized in other applications. Another type of inertial sensor is a gyroscope, which measures angular velocity around three orthogonal axes. Traditionally, these inertial sensors were implemented using bulky mechanical fixtures which were costly but accurate. In the recent years, microelectromechanical systems (MEMS) were used to implement inertial sensing methods in the micro scale on an integrated chip (IC). These MEMS accelerometers are used in a variety of applications, such as airbag deployment, earthquake detection and navigation purposes [1]. These MEMS sensors are lower cost and smaller, but limited in performance; however, recent development sensors and methods are increasing the performance of these MEMS sensors enabling them for navigation purposes [2]. Due to the reduction in price and smaller size of these MEMS sensors, they have also been increasingly utilized in consumer devices for step tracking and new forms of human and computer interactions such as the Oculus Rift [3] [4]. In smartphone consumer devices alone, this accounts for more than 1.4 billion units and sales of 400 billion dollars in the global market.

These MEMS inertial sensors suffer from a variety of errors, limiting their implementation in navigational purposes. These errors arise from uncertainty in measuring the acceleration, such as nonlinearities, bias variabilities, noise and other sources which are further magnified when integrated to get velocity and displacement. In addition, common applications of a MEMS accelerometer naturally introduce attitude errors that contaminate the results of the accelerometer with the effects of gravity, further causing issues when integrated. With increasing use in navigational and military uses, it is crucial to understand and determine important specifications that need to be considered when evaluating the performance of these inertial sensors.

Specifications are used to compare between devices, but these specifications are often not comparable because of methodology and testing platform differences. A study by the National Institute of Standards and Technology (NIST) found a lack of standardized testing protocol for evaluating accelerometer performance, which resulted in differences between their specifications and the actual measurements [5]. Currently, to specify and test inertial sensors for a specific application, evaluation boards from the respective manufacturers of each device are utilized. These differences between the evaluation boards were an issue, resulting in difficulty when comparing the performance of the different accelerometers [6]. Another method currently used involves custom evaluation boards that are used with a rate table or shaker to test specifications of a single inertial sensor, which limits the comparison [7] [8]. The importance of a consistent testing methodology and platform is necessary for a comparison between devices, especially for users of accelerometers, as the specifications are commonly used to identify the performance of these inertial sensors.

In addition to the specification of an accelerometer, it is also important to look at key application areas of accelerometers. MEMS accelerometers are an important part of an Inertial Navigation System (INS), which are used to calculate location and attitude of a system. Inertial navigation systems utilize accelerometers and gyroscopes in a variety of different applications, ranging from military, robotics, and transportation. Most practical applications of INS use a multitude of different sensors, such as accelerometers, compasses and gyroscopes, alongside external references such as Global Positioning System (GPS) or Wi-Fi. Extensive research has been done on external reference enhanced-INS, however these external references are not often available in transportation or indoor scenarios [4] [9] [10] [11]. In these scenarios, the ability of the INS to determine the attitude and heading is crucial. One major area of research in inertial navigation systems involve location and attitude estimation without the use of external references. These systems combine accelerometers with gyroscopes and compasses due to a significant error when using the accelerometer alone. There has been research into inertial algorithms to improve the ability for INS to estimate the location and attitude [12] [13]. There has also been research into improving the capabilities individual MEMS sensors alone through the design of the readout circuit in the IC [14]. Both these areas consider the evaluation of the different algorithms for one device, or the evaluation of a single device and the performance of their specifications. However, in common applications of MEMS accelerometers, they are often used in conjunction with other inertial sensors through the use of inertial algorithms. What our study will look at is the impact of these specifications once common inertial algorithms are applied, and test this across two common usage scenarios.

To summarize, there are two main goals that this study is hoping to achieve:

- Develop a low-cost testing platform and define a consistent testing methodology that will be able to measure key specifications and compare between the capabilities of different MEMS accelerometers.
- Determine the impact of these specifications after common inertial algorithms are applied, and evaluate the importance of different specifications on usage scenarios.

1.2 Study Organization

To effectively achieve these goals, the thesis will begin by covering the background needed. Chapter 2 will do this by covering the basic dynamics of MEMS accelerometers and INS. It will also explore inertial algorithms which are commonly used for sensor fusion and navigation purposes.

Chapter 3 will show the platform and testing board that were developed to support the different device under test (DUT) that were evaluated. A testing methodology for 5 specifications were outlined and utilized to evaluate a variety of devices. Some of these are more complex movements which are helpful to evaluate the performance of the devices in more realistic motion. These specification results are compared to data sheet when possible, but otherwise, the feasibility of these testing methodologies is evaluated by comparing between the different devices.

Chapter 4 will explore two common inertial algorithms used in an INS – Kalman and Complementary filter. The specific algorithms were tested in two common applications of an INS – attitude estimation and dead reckoning. The impact of the different device specifications in these inertial applications are explored in this chapter.

Finally, a summary of the results is drawn and other aspects which can be further explored is discussed in the last chapter.

Chapter 2 Background

2.1 MEMS Inertial Sensor

There are two main inertial sensors that will be utilized in this thesis. The first one will be an accelerometer, the focus of this study, and the second will be a gyroscope, which is used in the inertial algorithms. This section will cover some of the basic principles of operations and how each of them will be used.

2.1.1 Accelerometer Principles of Operation

An accelerometer is based upon the principles of a spring-dampener system. A frame is connected to a known mass through a spring and dampener system. When an acceleration or force is applied to the frame, it is measured by looking at the displacement between the mass and the frame.

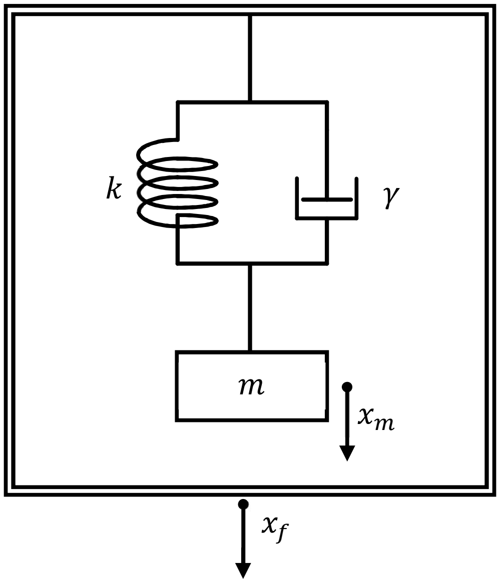


Figure 2-1 – Mechanical basis of a MEMS accelerometer.

The following equation is derived from looking at the forces acting on the proof mass,

$$F_{net} = ma_m = m\frac{\delta^2 x_m}{\delta t^2} = F_e - k\left(x_m - x_f\right) - \gamma\frac{\delta\left(x_m - x_f\right)}{\delta t} \tag{2.1}$$

F_e is the external force acting on the proof mass that causes the displacement. The second term of the equation results from Hooke's law, where k is the spring constant. The third term results from the damping that is often done to prevent the sensor from ringing, γ is the damping coefficient of the gas that is used in the system.

Subtracting $m\frac{\delta^2 x_f}{\delta t^2}$ from both sides and rearranging,

$$m\frac{\delta^2(x_m - x_f)}{\delta t^2} + \gamma\frac{\delta\left(x_m - x_f\right)}{\delta t} + k\left(x_m - x_f\right) = F_e - m\frac{\delta^2 x_f}{\delta t^2} \tag{2.2}$$

Substituting $x = x_f - x_m$, and $F = m\frac{\delta^2 x_f}{\delta t^2} - F_e$, it becomes a second order differential equation of the following form.

$$m\frac{\delta^2 x}{\delta t^2} + \gamma\frac{\delta x}{\delta t} + kx = F \tag{2.3}$$

Solving this differential equation for x, which is the distance between the frame and the proof mass,

$$x = \left[\frac{\frac{\delta^2 x_f}{\delta t^2}}{s^2 + s\frac{\gamma}{m} + \frac{k}{m}}\right] = \left[\frac{\frac{\delta^2 x_f}{\delta t^2}}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}\right] \tag{2.4}$$

Accelerometers which are used for inertial navigation are working at frequencies much lower than the resonant frequencies, resulting in the following approximation,

$$x \approx \frac{a_f}{\omega_0^2} \tag{2.5}$$

Measuring the distance between the frame and the proof mass will allow us to calculate the acceleration of the frame. An important consideration is the effect of gravity on this mass spring dampener system. A spring dampener system measures force, and not simply the acceleration. This results in a $9.81\frac{m}{s^2}$ constant acceleration in the direction of the sensor aligned with gravity. One of the most popular methods of measuring displacement in a MEMS accelerometer is capacitive sensing.

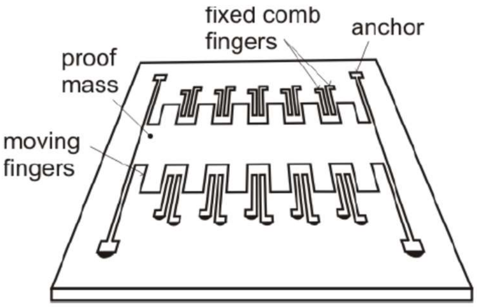


Fig 2-2 – Typical MEMS Capacitive sensing implementation.

The proof mass acts as one of the plates in a capacitor, while the anchored comb fingers act as the other capacitor plate, and the frame. The change in capacitance is measured and used to determine the acceleration.

2.1.2 Gyroscope Principles of Operation

Gyroscopes use the properties of the Coriolis acceleration along with vibrations to measure the angular velocity of the system. Coriolis acceleration is observed in a rotating frame of reference and is proportional to the angular velocity.

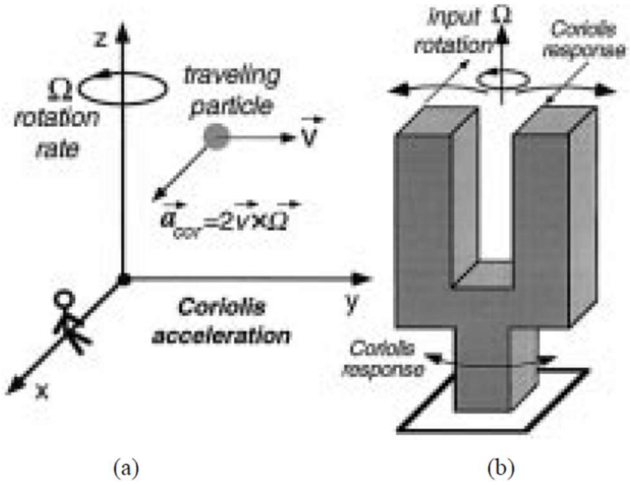


Fig 2-3 – Mechanical Basis of MEMS Gyroscope [15]

In fig. 2-3a, a particle traveling in the direction of the y-axis with velocity $\vec{v}$ is also rotated around the x-axis with angular rotation Ω; an acceleration is seen along the x-axis that is proportional to the angular rotation due to the Coriolis effect, hence named Coriolis Acceleration. In a MEMS gyroscope, this is commonly achieved using a tuning fork system, where vibrations are electrically driven along an axis and then sensed in the orthogonal axis by measuring the amplitude of the vibrations. In the absence of rotations, the sensing axis will not measure any acceleration, but when the device is rotating, it would appear as if the axis are coupled, and the Coriolis acceleration is seen.

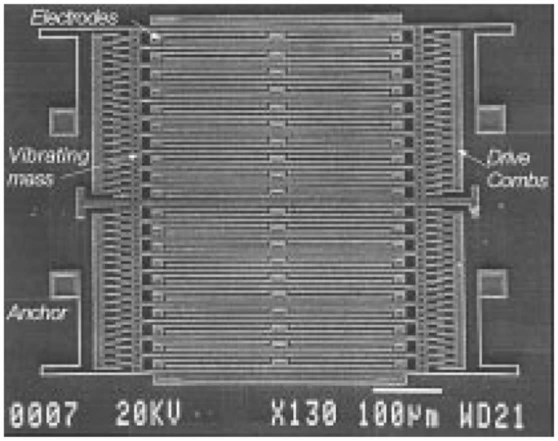


Figure 2-4 SEM view of a comb-driven polysilicon surface micromachined [16]

Capacitive sensing is also one of the most common techniques used to sense the vibration amplitude. Fig. 2-4 shows a vibrational mass anchored using MEMS springs, where the combs and vibrating mass sensed using the change in capacitance.

2.1.3 Typical INS Principle of Operation

In a typical Inertial Navigation System, the angular velocity and acceleration are each measured along 3 axes. Figure 2-5 shows typical system and their measurement axis.

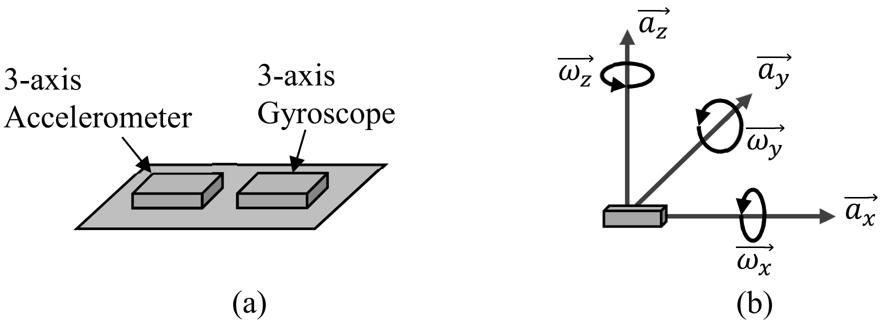


Figure 2-5 – (a) shows a basic INS setup involving two 3-axis MEMS accelerometer and Gyroscope, and (b) shows their measurement axis.

An INS provides more than the raw acceleration and angular velocity. It is often used in conjunction with inertial algorithms to determine a myriad of different measurements. Common measurements used from an INS are listed as follows

- Acceleration
- Angular Velocity
- Attitude
- Velocity
- Position

2.2 Inertial Algorithms

Two common filtering methods used in an INS are the Complementary and Kalman Filter. This section will cover the principles behind each of these filters, the specific implementation for our testing will be covered in Chapter 4, where the implementation details of these filters are covered.

2.2.1 Complementary Filter

Sensor fusion is a method of combining different inertial sensors to estimate the attitude. A complementary filter is one such example which combines data from accelerometers and gyroscopes. This filter is commonly used for attitude estimation due to its simplicity and ease of implementation.

The Complementary filter can combine two different datasets by utilizing their different characteristics. The complementary filter structure low pass filters the dataset with high frequency noise and vice versa with a dataset with low frequency noise and a high pass filter.

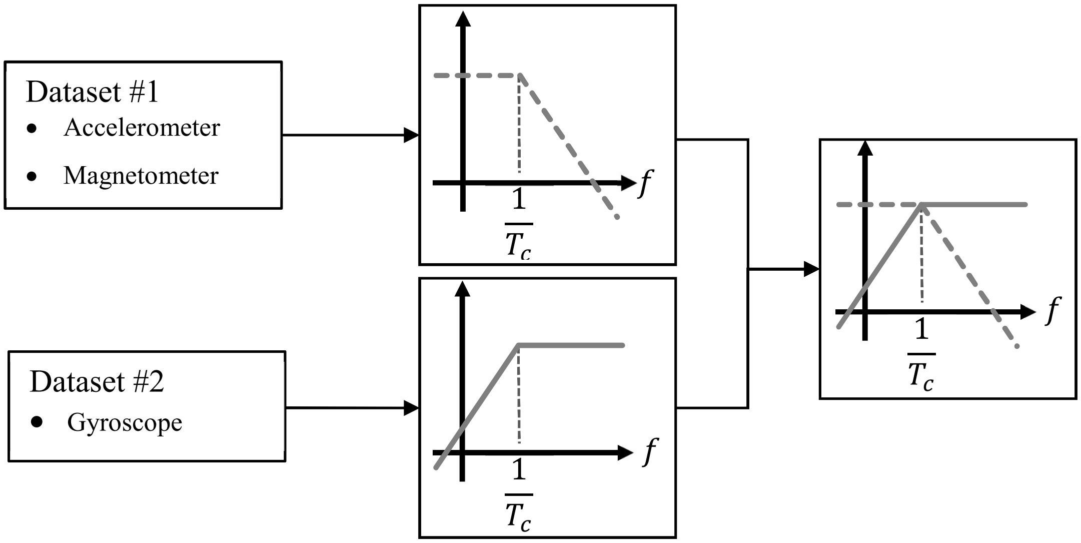


Figure 2-6 – Principles of operation for a complementary filter, with typical sensors listed.

The basics of a Complementary filter utilizes the differences of different data sets, and as such, it is often used to combine accelerometers and gyroscopes to estimate attitude due to their distinct advantages and disadvantages. There are different filter structures targeted towards inertial applications providing better gyro bias estimation, such as the Explicit Complementary filter and Passive Complementary filter. These all make improvements on the classical Complementary filter by avoiding coupling of different axis and incorporation of magnetometer results [17].

2.2.2 Kalman Filter

Kalman filtering is an estimation method that is used in systems where the effect of statistical noise affects the measurements. The filter uses multiple measurements with a dynamic model to estimate the state of the system recursively.

There are two main stages in Kalman filtering:

- Prediction (a priori) Stage
- Update (a posteriori) Stage

The prediction stage is used for the estimation of the state based on inputs and the dynamic model of the system. It doesn't utilize the measurement from the current time step. The update stage uses a current measurement and the a priori result to update and refine the result.

In the prediction stage, the state of the system ($\hat{x}_k$) is estimated based upon the previous state and the effects of the control (u_k). Likewise, the covariance matrix of the system (P_k) is also estimated. The control is an input that specifies how the dynamics of the system may change. This is analogous to knowing how much the gas pedal is pressed when the system is tracking speed.

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \tag{2.6}$$

$$P_{k|k-1} = F_k P_{k|k-1} F_k^T + Q_k \tag{2.7}$$

In the update stage, the innovation for the state (y_k) and the covariance matrix (S_k) is calculated using the measurement (z_k) and the a priori estimate. The innovation is then used to determine the kalman gain (K_k) which is used to adjust and update the state and covariance of the system.

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{2.8}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{2.9}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{2.10}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{2.11}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{2.12}$$

Through this recursive process, the state ($\hat{x}_k$) will be more accurate than utilizing the measurements (z_k) alone. To utilize the Kalman filter, it is crucial to determine the dynamics of the system in the form of Eqn. 2.6 and the observation z_k. This is fundamental, as it will determine the covariance matrices used in the filter. Utilizing the Kalman filter will help reduce the effects of stochastic noise on the measurements. Figure 2-7 shows a general view of how these equations are used to across the time steps.

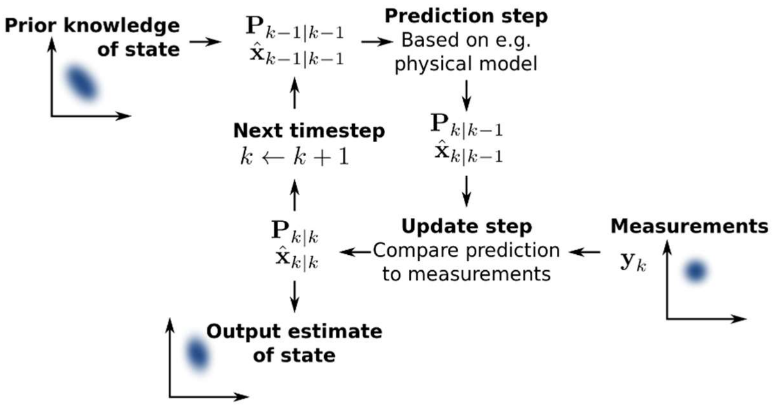


Figure 2-7 - The kalman filter is separated into the prediction and update step and is used to iteratively improve the output estimate in comparison to solely using the measurements or dynamics alone [18].

Chapter 3 Inertial Measurement Unit Specification

Specifying the performance of an accelerometer is crucial to comparing different MEMs devices. This chapter covers the testing methodology and platform that were developed to compare the performance of MEMS accelerometers. The chapter will first cover the testing platform that was created. Secondly, it will cover in detail the testing methodology for accelerometers and the specification results collected from a variety of devices under tests.

3.1 IMU Testing Platform

This first section covers the Inertial Measurement Unit (IMU) testing platform created. There are two main objectives for the construction of the platform.

1. Provide an easy data logging/processing solution for IMU sensors.
2. Provide a consistent testing platform that supports a variety of IMU sensors.

The construction of this testing platform is crucial to comparing accelerometers with a similar environment, however, the implementation details of the testing platform will be covered within Appendix A. The following sections will cover brief implementation details and views of the overall system only. The first section will cover the hardware platform that was created and the various design decisions made. The second section will cover the software designed to allow for a multi-sensor support.

3.1.1 Data Logging Hardware

The general system is composed of two boards – the motherboard and the sensor board. The motherboard encompasses the logging functions, processing capabilities and the power distribution, while the sensor board ensures connectivity of the sensor. Connecting the two is a common protocol that supports a variety of sensors. By building a custom testing platform, this ensures the testing platform has a consistency between different device under tests (DUTs). Figure 3-1 shows the basic block diagram of the system and Figure 3-2 shows the final data logger implemented in a Printer Circuit Board (PCB). Specific implementation details are available in Appendix A.

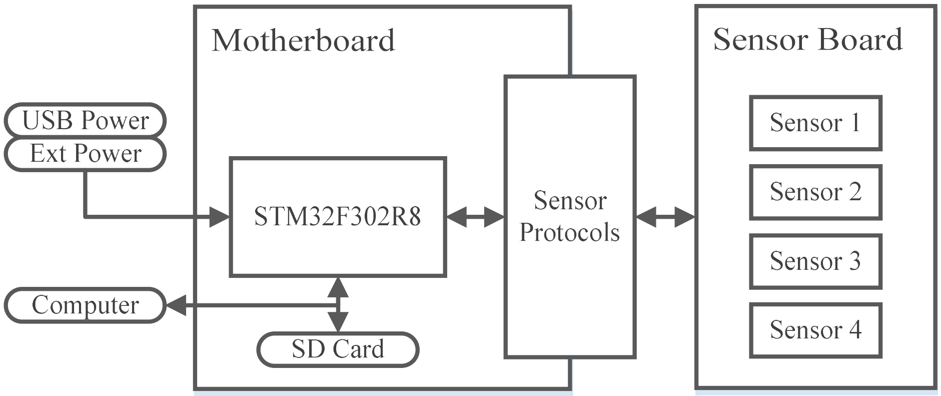


Figure 3-1 – This figure highlights the inputs/outputs of the system and the connections of the major blocks.

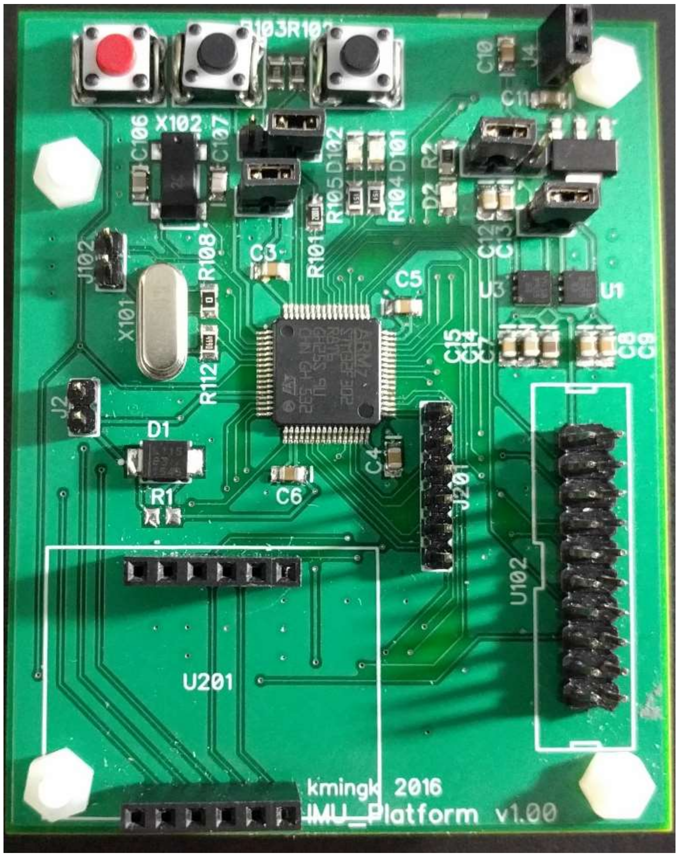


Figure 3-2 Picture of data logger platform motherboard

3.1.2 Sensor Protocols and Example implementations

For the platform to support a variety of sensors, several different protocols are supported by the system. Instead of supporting all the protocols on separate pins, and using them inefficiently, the pinout was chosen such that various combination of board configurations are supported. The following configurations are valid sensor board configurations, without external multiplexers.

- 1 Analog device (3-Axis)
- 1 Analog device (3-Axis) + up to 3 Serial Peripheral Interface (SPI) devices
- 1 Analog device (3-Axis) + up to 3 I2C devices
- Up to 4 SPI or I2C devices

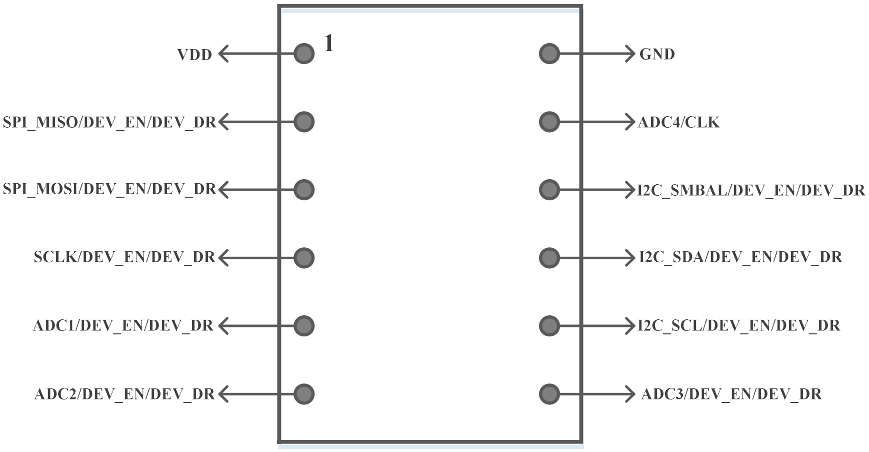


Figure 3-3 - A pinout of the sensor board to motherboard connection. These are the most important pins, each of the pins can be repurposed as an enable to the sensor or data ready signal from the sensor.

There are a variety of ways the sensor boards can be designed. This is an example of the sensor board that utilized 2 SPI devices. This was created for a specific pair of sensors which both utilized the SPI protocol. Shown here is the pinout and resulting PCB:

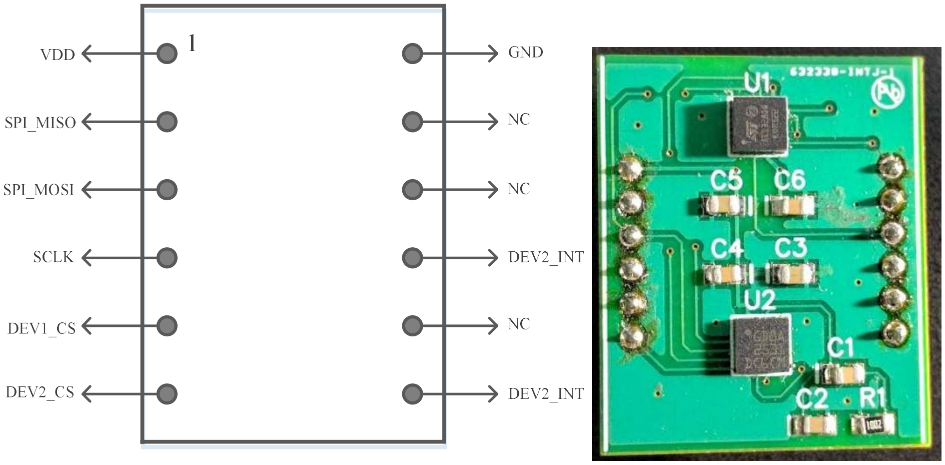


Figure 3-4 Example of pinout for two SPI devices(Left) Picture of implemented PCB (Right)

3.1.3 Software

This section mainly covers the software implementation of the Data Logger. This includes the API and system level organization of the data logger with an execution description of the logger program developed.

3.1.3.1 *Architecture*

The software architecture was developed for the future adoption of new sensor boards. To do this, the construct of a sensor board and device level API was created, which allows for flexibility in the sensor board and sensor codebase. This abstraction allows duplication of devices between boards without redundant code, and allows for easy configuration of the board such that different pinouts can be laid out easily. The following figure shows the functional blocks which were created within the architecture to support the sensor board system.

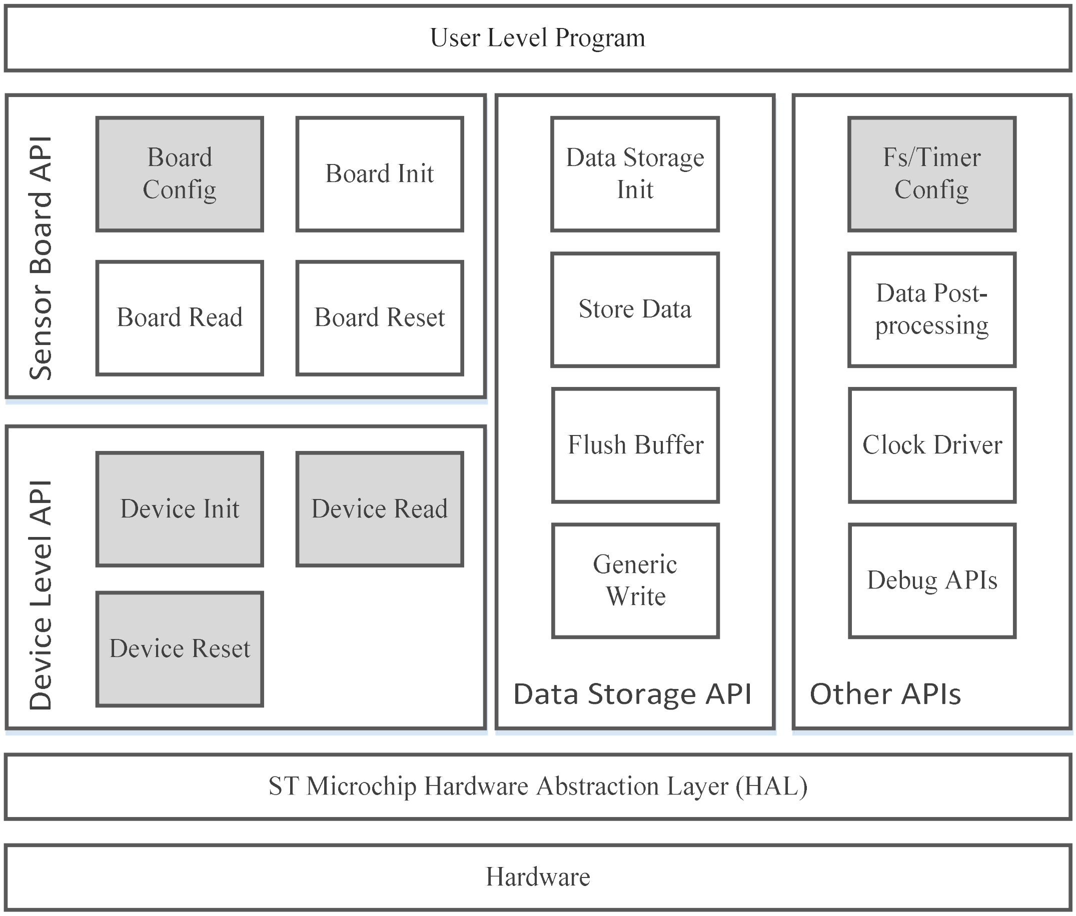


Figure 3-5 Software API Architecture. Blocks in grey indicate blocks that will need to be configured specifically for each new device.

There were four main categories of Application Programming Interface (API) created for the platform:

Device level API – This is a user implemented API with a defined prototype of functions. The main purpose is to have a common method of initializing the sensor, configuring it and reading values from it.

Board level API – This was mainly implemented in the imu_wrapper.c file. As long as the device level APIs are readily available, a board can be integrated into the system using configuration settings, after which the system can utilize the same functions to initialize/reset/read to the IMUs on the board.

Data Storage API – This supports the use of SD card or Serial port to transfer data. In the case of the SD card, the data is stored in a text file using the FatFs filesystem implemented by ChuaN.

Other APIs – This api mainly includes a variety of different useful functions, such as debug messages on the serial port, generating clock signals, internal timer initializations and data processing methods.

3.2 Mechanical Testing Platform

This section covers the setup which was used to produce the mechanical movements that was utilized within the metric testing stage. It includes a description of the setup and highlight specifics of the testing platforms. There are two main purposes to the mechanical testing platforms:

1. Provide an accurate measurement of metrics relevant to an accelerometer
2. Be a low-cost and low-effort setup.

3.2.1 3-D Printer

The main mechanical testing platform is a 3D printer. A 3D printer is a low-cost setup that provides a good control of the movement while being accurate to the millimeter. The printer used in this case is the Rostock MAX 3D printer which can achieve a resolution of 1mm at speeds of 800mm/s. The dispensing printer head was removed and used as a platform for the accelerometers to be attached.

The 3D printer is controlled using the Repetier Host application which accepts G-code as a way of controlling the movement. G-code allows the user to provide coordinates within the working space along with a speed.

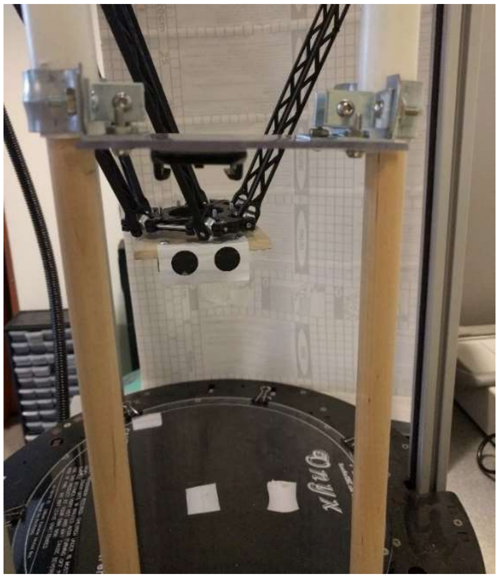


Figure 3-6 - Picture of the 3-D printer platform with the horizontal accelerometer testing rig attached

The printer has two main benefits:

- Control – g-code allows precise control over the positioning of the platform.
- Repeatable – movements are highly repeatable, allowing for consistent results.

Although this makes the printer a good candidate for mechanical testing, there are two main drawbacks to the 3D printer which makes the design of the movement patterns extremely important.

- Mechanical noise – noise introduced is highly inseparable from data.
- Peak Acceleration – peak acceleration within a ± 0.3g range.

Due to the noisy nature of the 3D printer, and low signal amplitude, it makes the 3D printer unsuitable for linearity measurements. Thus, another mechanical setup is used for linearity.

3.2.2 Circular Motion Platform

This platform is mainly used for the linearity measurements. Linearity is difficult to measure without the ability to produce a good quality signal at a high signal amplitude. This mechanical platform utilises the concept of circular motion and gravity to produce a ±1g tone which is a clean input needed to test linearity. Details of the implementations will be discussed in the outlining of the testing methodology.

3.3 Accelerometer and Gyroscope Selection

The following accelerometers are the list of accelerometers used in testing. They were selected from three manufacturers that have shown good results in a previous study. [6] As noise density are often used to evaluate the relative performance of difference devices, they were selected to cover a range of noise densities. The following is the list of devices that were tested across the different suite of tests.

Table 3-1 Table of Accelerometers tested

Accelerometer Model	**Manufacturer**	**Noise Density** $[\frac{\mu g}{\sqrt{Hz}}]$	**Type**	**Cost** [$]
LSM303D	ST Microchip	150	Digital	4.40
LIS3DSH	ST Microchip	150	Digital	3.64
AIS328DQ	ST Microchip	218	Digital	13.47
MPU6000	Invensense	400	Digital	5.28
KXTC9	Kionix	125	Analog	5.25
ICM20689	Invensense	150	Digital	5.86
KXRB5	Kionix	40	Analog	10.25

3.4 Accelerometer Metrics Test

Using the testing platform outlined before the devices were tested in several specifications. A consistent testing methodology was used to test each specification, ensuring a consistent comparison.

1. Noise Density
2. Cross-Axis Motion
3. Linearity
4. Bias Variability
5. Step Motion

These five were chosen because the first three are commonly specified on data sheets, and the last two are tests that will give a different look at accelerometers. The testing methodology will discuss the test dynamics, metric calculated for each specification and analysis of the results. Temperature and vibrational effects were initially considered as well, however, due to limitations of the testing platform, they are hard to measure consistently and as such not explored in this study.

The purpose of a standard specification methodology is for an accurate comparison of Accelerometers. Accelerometers are used in a variety of settings which allow for different metrics to be prioritized. To serve as an example application, the specification results were weighted for a dead-reckoning application. The weighting was derived from the importance of velocity and displacements in a dead-reckoning application, where the bias variability and step motion results will take a higher precedence.

Table 3-2 - Example application weighting of metrics

Metric	Weight (Out of 25)
Noise Density	5
Cross-Axis	3
Linearity	3
Bias Variability	7
Step Motion	7

The data logger platform is powerful enough to process the data, however, MATLAB was used to process the data instead so different processing methods can be tested. The following is the standard configuration.

- Data was sampled at 400Hz.
- Bandwidth set at 44-50Hz range.
- Hanning Window Applied (before frequency analysis only)
- Per device calibration done to remove offset and gain errors.

AIS328DQ's bandwidth is restricted to half of the sample rate at 200Hz from the device, so it cannot be reconfigured.

3.4.1 Noise Density

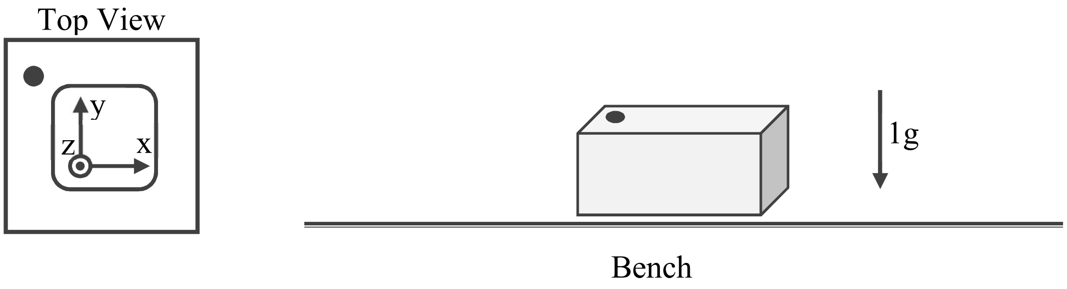


Figure 3-7 Dynamic Model of the Noise Density Test

To measure noise density, the device was mounted on the bench with the z axis aligned to gravity. The test dynamics are relatively straightforward; the device was held stationary for 10 minutes on the bench. The resulting data was analyzed to calculate the noise density. Noise density is calculated for all three axes and averaged for the final metric.

3.4.1.1 *Time Domain Method*

There are two calculation methods for noise density which are evaluated. The first method uses the assumption that in a stationary state, the noise model of the accelerometer is white noise. If that is the case, noise can be calculated using the following equation:

$$ND = \frac{\sigma}{\sqrt{f_{3dB}}} = \sqrt{\frac{1}{f_{3dB}}\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3.1}$$

The time domain method is affected by the device's inherent filter and 1/f noise, as such, the preferred method is the frequency domain method it provides a more accurate derivation of the noise density.

3.4.1.2 *Frequency Domain Method*

The frequency domain analysis is more accurate, as it takes into account the noise model of the accelerometer. However, it is more computationally intensive and requires more understanding of the accelerometer.

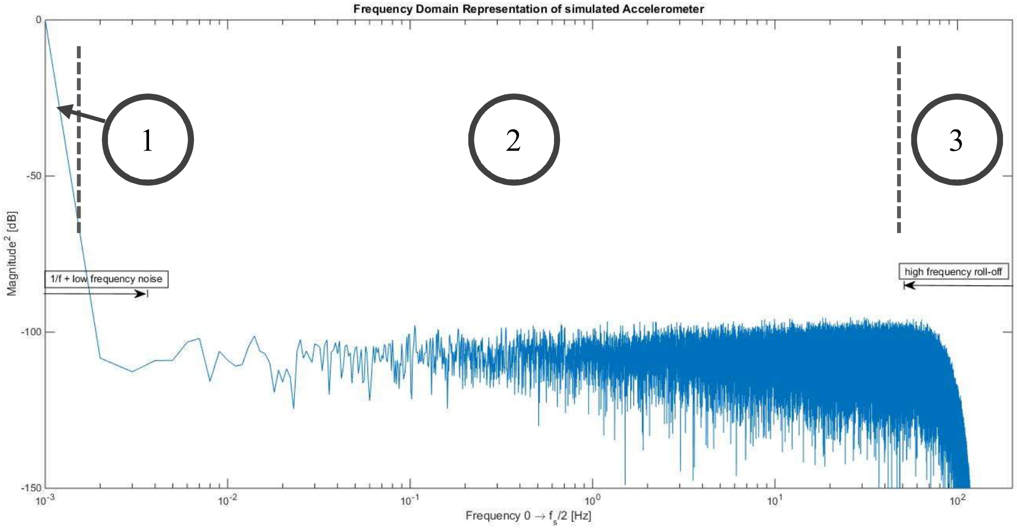


Figure 3-8 Frequency representation of simulated noise on z-axis.

The figure above shows the FFT of a typical accelerometer noise model; the model is separated into 3 distinct regions. Region 1 involves the bias and 1/f noise, which is unwanted and not part of the noise density calculation. Region 3 is the roll-off caused by the low pass filtering done on the device. As region 3 doesn't affect where the signal of the system is, it is not included when calculating the noise density. Therefore, the region of interest, is region 2, where the noise density is calculated using the following formula:

$$ND = \frac{1}{f_{3dB}} \sqrt{\frac{1}{f_2 - f_1} \int_{f1}^{f2} \left|\overrightarrow{A(f)}\right|^2 \, df} \tag{3.2}$$

f_{3dB} is the bandwidth of the device, and is equal to the end frequency, f_2. The start frequency f_1 was empirically determined to be 1/100th of the bandwidth to ensure the 1/f noise doesn't affect the noise density calculation.

$$f_1 = \frac{1}{100} f_{3d} \tag{3.3}$$

3.4.1.3 *Noise density Results*

Figure 3-9 and 3-10 show an example of how the time domain and frequency domain of a noise density plot will appear like. The spectrum shows the different sections of the noise model as well.

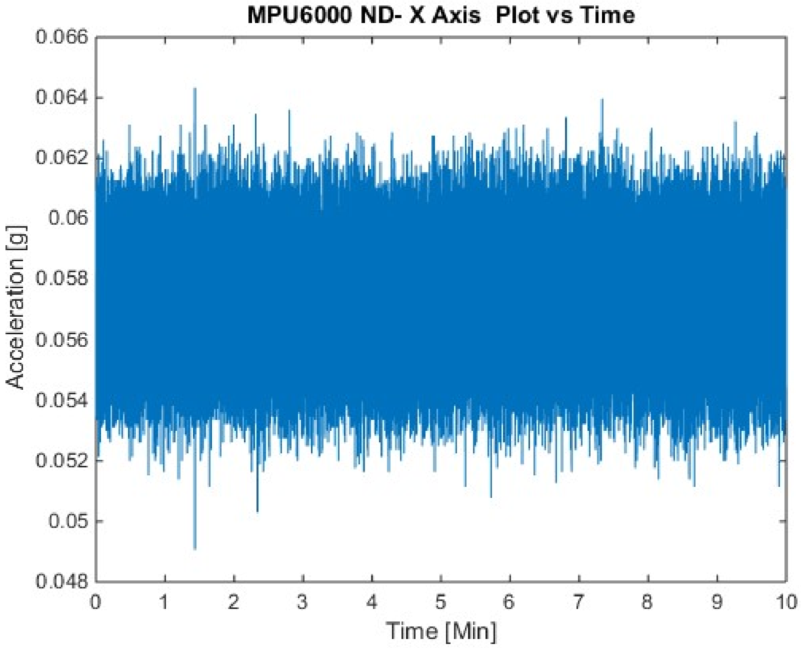


Figure 3-9 Time domain plot of noise density test

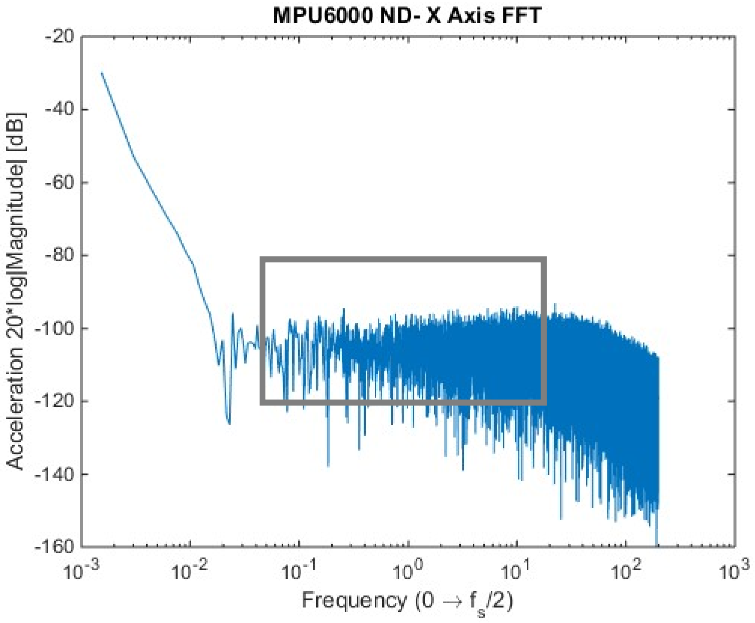


Figure 3-10 Frequency domain plot of the noise density test – box indicates the region used for noise density calculation

Using both the frequency and time domain methods, the noise density was determined to be the following for the different accelerometers.

Table 3-3 Noise Density test results

Device	**Spec Sheet** $[\frac{\mu g}{\sqrt{Hz}}]$	**Noise Density** Time Domain method $[\frac{\mu g}{\sqrt{Hz}}]$	**Noise Density** Frequency domain method $[\frac{\mu g}{\sqrt{Hz}}]$	**Total Noise** $[\mu g]$	**Bandwidth [Hz]**
LSM303D	150	619	826	5840	50
LIS3DSH	150	355	604	4270	50
AIS328DQ	218	361	360	5091	200
MPU6000	400	252	413	2739	44
KXTC9	125	126	194	1372	50
ICM20689	150	175	280	1857	44
KXRB5	40	138	227	1605	50

There are two main points of discussion from the summary which will be explored in detail:

1. Time domain vs Frequency domain
2. Discussion of outliers LSM303D/LIS3DSH

The time domain analysis consistently underestimates the noise density except for AIS328DQ, where it is comparable since the bandwidth is at half of the sample rate. In general, the time domain analysis will take into consideration the effect of the filtering, resulting in lower noise density. If the frequency method limits were changed to include the high frequency ranges, the noise density will match the time domain results. Most of the devices have a spec sheet noise density which does not specify how they are calculated. The time domain method and frequency based method can yield vastly different results particularly when different low pass filters are employed by different devices. By quoting based of the frequency domain, this eliminates the ambiguities that different spec sheets have. In addition to calculation methodology, data sheets do not differentiate between different axes. The following table shows that there is a considerable difference between them.

Table 3-4 Frequency Domain based Noise Density result per axis

Device	Spec Sheet $[\frac{\mu g}{\sqrt{Hz}}]$	Measured $[\frac{\mu g}{\sqrt{Hz}}]$		
		x	y	z
LSM303D	150	602	348	1530
LIS3DSH	150	583	421	809
AIS328DQ	218	319	299	464
MPU6000	400	369	337	532
KXTC9	125	210	37	336
ICM20689	150	282	283	273
KXRB5	40	176	223	282

Secondly, looking at LSM303D and LIS3DSH, the noise density is significantly higher than the specification. The discrepancy results from the output data rate selection; at higher sampling rates, even when the device is kept at the same bandwidth, the noise is much higher than specified. Noise density should not be affected by the output data rate, but rather the device bandwidth. LSM303D and LIS3DSH share a similar chip architecture, with the difference being LIS3DSH includes an on-chip magnetometer, and both show this issue. This was recreated across 3 different copies of each chip. When the device was set at the same data rate and bandwidth specified by the device, it gives a very comparable noise density. This highlights one of the issues with testing accelerometers - most specifications are not listed, and even when they are, they are inconsistent across vendors and devices.

By keeping a consistent methodology, it is possible to compare the relative noise performance. In general, we can see that the specifications do follow a similar trend compared to the spec sheet dataFor noise density, the KXTC9 is the best performing in relation to all the devices tested. To ensure a relative performance is tracked across all the metrics, the following formula is used to score and compare the relative performance of the devices. The same formula is applied to all subsequent metrics as well.

$$Score = \frac{Metric_{best}}{Metric_{dut}} * 100 \tag{3.4}$$

Where the metrics of the respective devices are compared to the best scoring device in that category.

Table 3-5 - Scoring Matrix for Noise Density

Device	**Noise Density** $[\frac{\mu g}{\sqrt{Hz}}]$	**Score /100**
LSM303D	826	23
LIS3DSH	604	32
AIS328DQ	360	54
MPU6000	413	47
KXTC9	194	100
ICM20689	280	69
KXRB5	227	86

3.4.2 Cross Axis

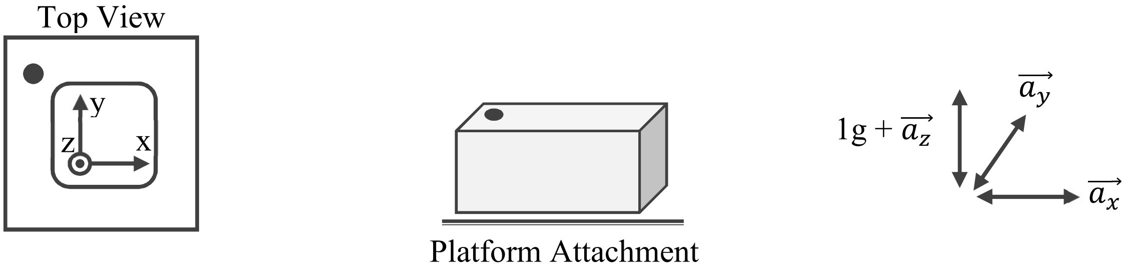


Figure 3-11 Dynamic Model of the Cross-Axis test

To measure cross-axis, a signal is introduced on one axis (signal axis), and the cross axis is the ratio of the signal introduced seen on the other axes (non-signal axis). In the figure above, this is equivalent to putting an input signal ($\overrightarrow{a_{signal\ axis}}$) and measuring its effect on the other axes. Then the cross-axis ratio is identified as:

$$Cross\ Axis\ Ratio = \frac{\left|\overrightarrow{a_{non-signal\ axis}}\right|}{\left|\overrightarrow{a_{signal\ axis}}\right|} \tag{3.5}$$

Any signal can be utilized for a cross-axis analysis, if the amplitude of the signal is identifiable. However, when using the 3D printer platform for motion generation, it will involve mechanical noise from the stepper motors, drive assemblies and other mechanical factors which are hard to differentiate from the signal, making the cross-axis ratio hard to calculate. To work around this, the input signal can be specified to allow for differentiation between the signal and mechanical noises. On the low-cost 3D printer, it is difficult to generate a smooth sinusoid; thus, a small duty cycle square wave was determined empirically to be easily differentiable from the mechanical noise.

1. Stop the device at the starting position for calibration period.
2. Move a set distance (D) at a constant speed (V) along the signal axis.
3. Return to the starting point at the same speed
4. Repeat steps 2 and 3 at a fixed frequency.

To verify this approach, the test movement was simulated assuming a 1g spike in acceleration for a short time duration, and the period of the whole motion is 1 second. Figure 3-12 shows the time domain view of this signal, and 3-13 shows the effect in the frequency domain.

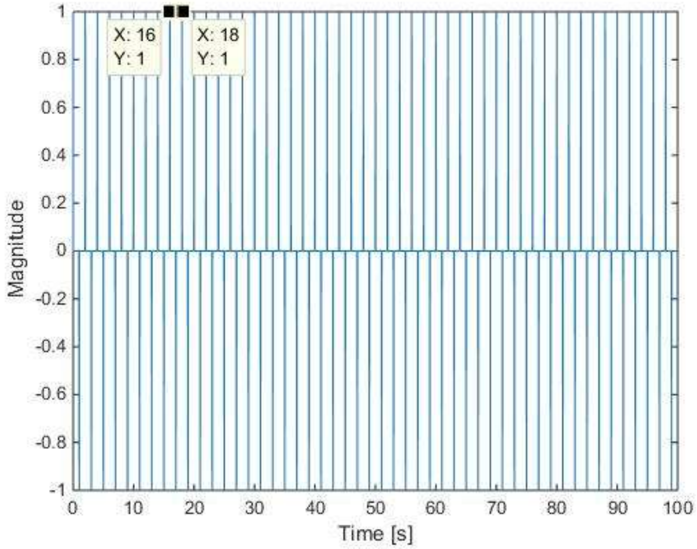


Figure 3-12 Simulated motion pattern for cross-axis calculation with no accelerometer noise model, $f_s = 400H$ $, T = 1s$

FFT simulation of delta train

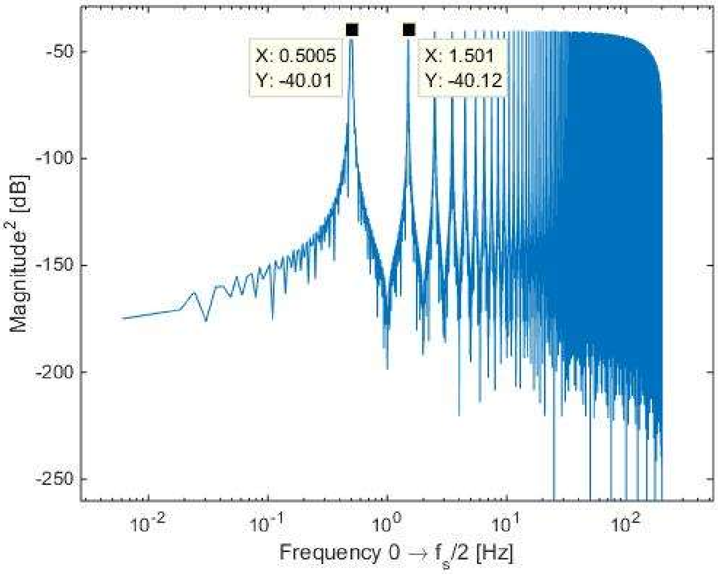


Figure 3-13 – Frequency domain of the simulated cross-axis test motion. $\Delta f \propto \frac{1}{T}$

This motion was chosen because it is easily identifiable in the frequency domain; in the frequency domain, a similar delta train is seen, where the spacing is dependant on the frequency.

In simulation results, similar delta trains can be identified on the signal and non-signal axes allows us to differentiate the signal from the mechanical noise. The power of this signal can be determined by summing up the power of the pulses seen in the frequency domain. By looking at the frequency bins where this mechanical motion exists, the cross-axis ratio can be calculated

$$p_{axis} = \sum_{\mathrm{N}} \left|\overrightarrow{A_{ax\imath s}(f)}\right|^2 \tag{3.6}$$

$\overrightarrow{A_{ax\imath s}(f)}$ is the frequency representation of the signal

N is the number of deltas to sum up the power – across the bandwidth of the device.

$$Cross\ Axis\ Ratio = \sqrt{\frac{p_{non\ signal}}{p_{signal}}} \tag{3.7}$$

$p_{non\ signal}$ is the signal power that is seen on the non-signal axis.

p_{signal} is the signal power that is seen on the signal axis.

The cross-axis metric is often specified on spec sheets as a ratio of the input signal, as such, we normalized it to a ration of amplitudes to allow for easy comparison with the spec sheet as well. The cross-axis ratio was calculated by introducing this signal individually on each axis and results were averaged to get the final cross-axis ratio for the device.

Figure 3-14 and 3-15 show the runs measured on KXRB5. In both the signal and non-signal axis, there is a significant amount of mechanical noise apparent in the $10^0 - 10^2$Hz range, which was shown to cause trouble when only comparing amplitude of the time domain data. In the signal axis, there are spikes occurring at the signal frequencies. Therefore, to determine the cross-axis ratio, the power of the signal was calculated by summing across all the signal frequencies within the bandwidth of the device.

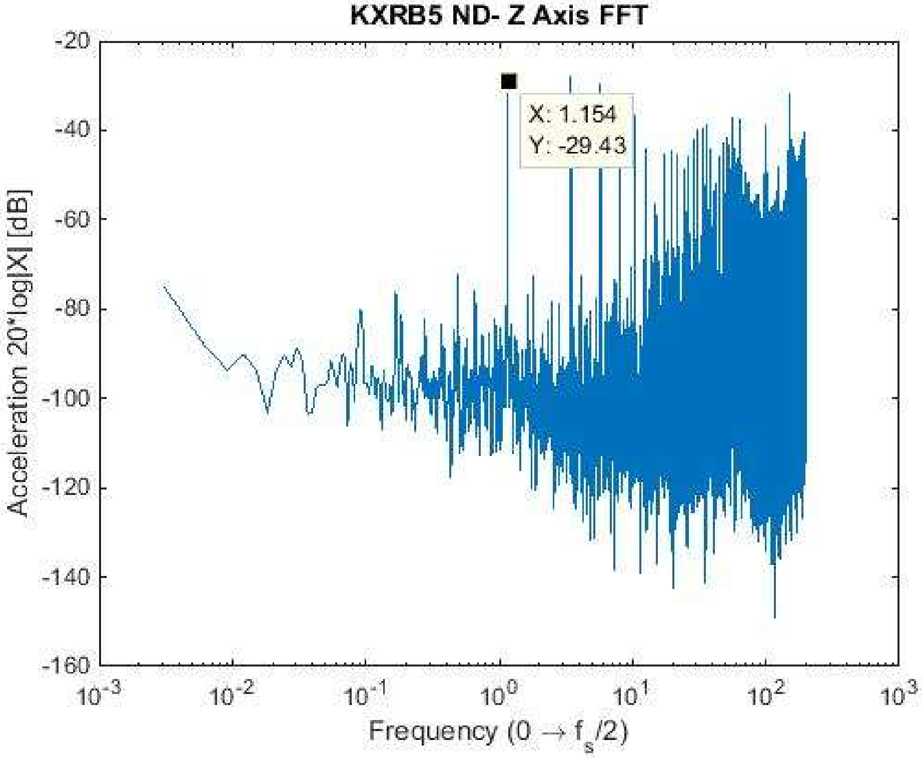


Figure 3-14 Fast Fourier Transform (FFT) of the data seen on the Signal Axis of Cross-Axis test.

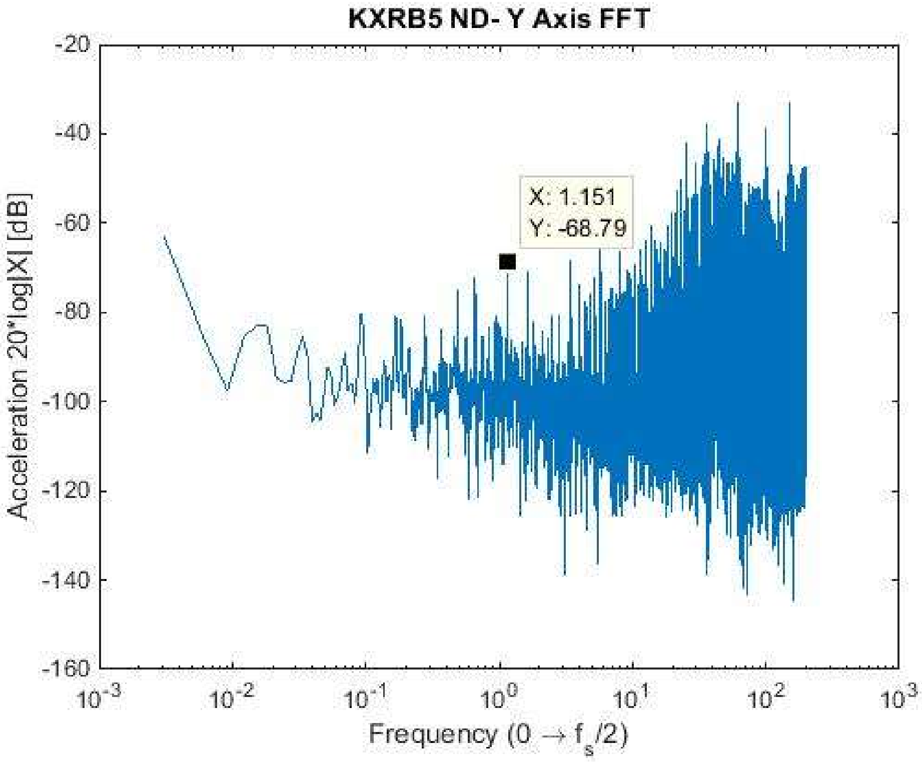


Figure 3-15 FFT of the non-signal axis in the cross-axis test

By analysing frequencies where the signal is in, it mitigates most of the effect from the mechanical noise which affects all axis in a similar fashion. From the testing results, KXRB5 has the best cross axis performance. Other devices are comparable in terms of cross axis performance, with most of them around the 1% cross axis ratio range. Most spec sheets do not specify the methodology of testing this, but our consistent methodology does show a similar result to the spec sheet comparison. This method is a way of utilizing a noisy platform to determine the cross-axis performance without the use of a highly controllable shaker table.

Table 3-6 - Cross Axis results and scoring summary

Device	Spec Sheet	Cross Axis Ratio	Score /100
LSM303D	--	0.015	53
LIS3DSH	--	0.012	67
AIS328DQ	0.05	0.025	32
MPU6000	0.02	0.017	47
KXTC9	0.02	0.012	67
ICM20689	0.02	0.010	80
KXRB5	0.02	0.008	100

3.4.3 Linearity

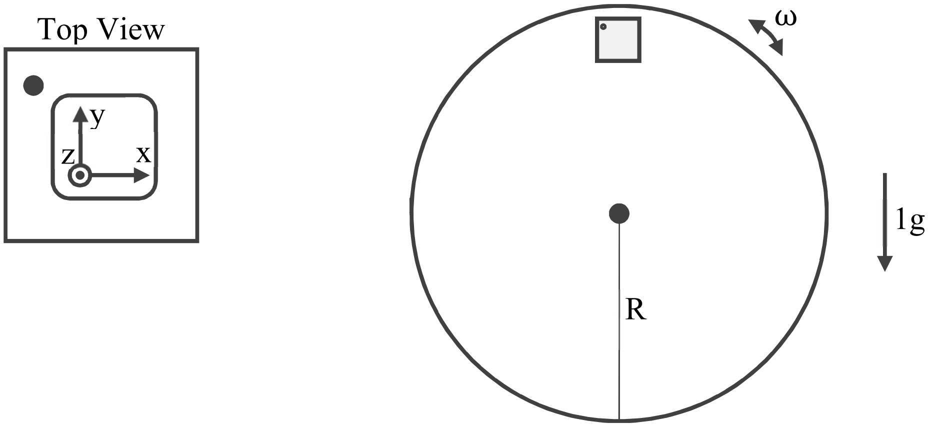


Figure 3-16 Dynamic Model of Linearity Test

Linearity is the third specification that is explored. It is important to produce a known signal with a known amplitude and frequency to measure linearity. However, due to the characteristics of the stepping motor, the 3D printing platform introduces mechanical noise and discretization which inhibits the ability to evaluate the linearity of a device. Linearity is especially a concern at higher accelerations near the limits of the accelerometer, as this is the region where the linearity is the highest. For this test, a vertical rotational platform was devised such that a tone with a $1g$ amplitude can be created. In this test, the vertical rotation platform will have a circular motion which induces the following forces on the tangential and radial axis:

$$\overrightarrow{a_t} = gsin(\omega t) \tag{3.8}$$

$$\overrightarrow{a_r} = -[\overrightarrow{a_c} + gcos(\omega t)] \tag{3.9}$$

$\overrightarrow{a_c}$ is the centripetal force caused by the rotational motion of the platform.

g is the force of gravity.

ω is the rotational speed in cycles per second.

To test linearity, the tangential acceleration is aligned with the axis to be tested. The resulting motion would provide a tone with an amplitude of $\pm 1g$ on the measurement axis of the device. This is not the limits of the accelerometer, as they can be rated to $\pm\ 8g$, but for navigational purposes, typical acceleration values seen on subways are much lower than this. Looking at the frequency domain, and the tones and harmonics generated, the metric to determine the linearity of the device for that input can be calculated. Figure 3-17 and 3-18 show the frequency domain of the test, where the input tone is seen at around $1.6Hz$ and the second and third harmonics are seen when it is zoomed in. Signal to Noise and Distortion Ratio (SNDR) is calculated by taking the power of the fundamental over the power of the noise and distortion terms. Signal to Noise Ratio (SNR) is the ratio of the fundamental and noise, while the Dynamic Range (DR) is the ratio of the fundamental and the harmonics.

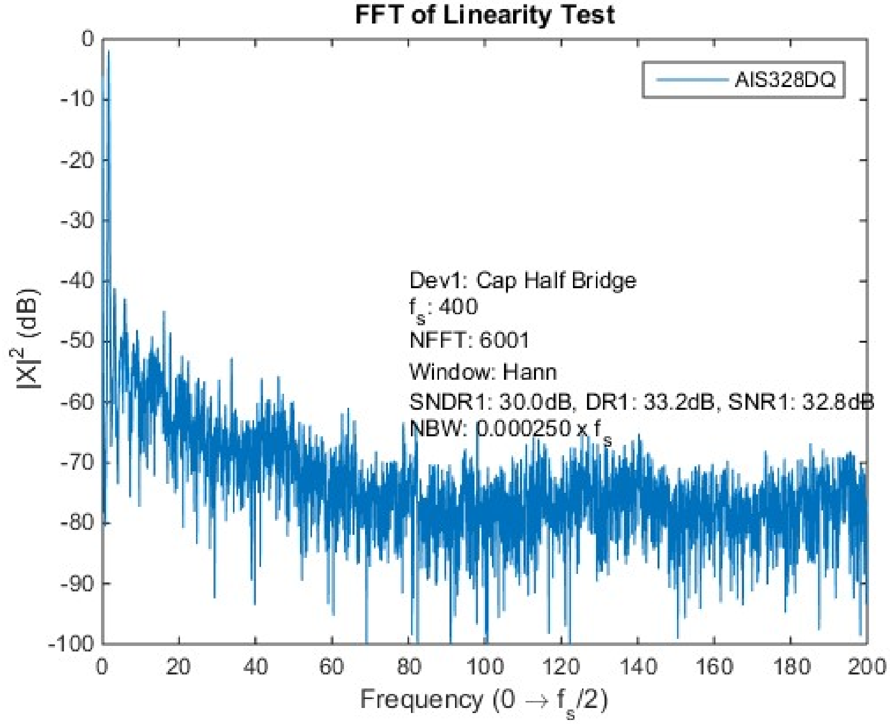


Figure 3-17 FFT of linearity test.

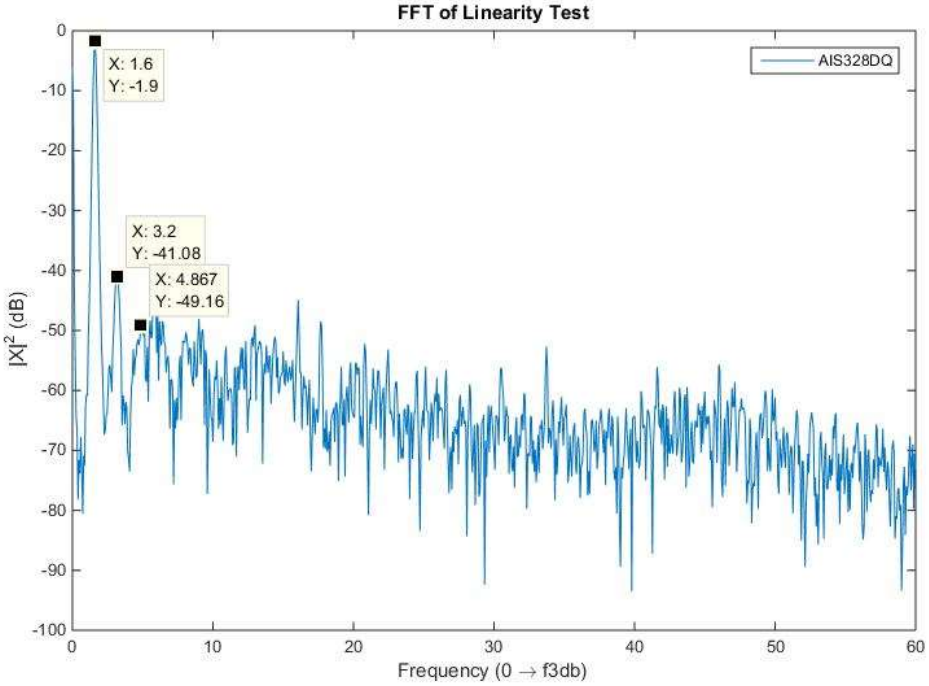


Figure 3-18 FFT of linearity test, showing second and third order harmonics.

Using the same methodology, the metrics for linearity are calculated for each device and the results are below. Since we are focusing on the effects of the harmonics, the dynamic range was used to determine the relative scoring.

Table 3-7 Linearity Summary and Scoring Results

Device	**Spec Sheet**	**SNDR** [dB]	**SNR** [dB]	**DR [dB]**	**Score /100**
LSM303D	--	32.5	35.6	35.5	92
LIS3DSH	--	32.7	36.6	35.0	90
AIS328DQ	--	30.0	32.8	33.2	86
MPU6000	--	34.4	38.4	36.6	95
KXTC9	30	34.6	38.8	36.6	95
ICM20689	--	34.9	37.5	38.3	99
KXRB5	30	35.1	37.6	38.7	100

Linearity is one of the less documented specification compared to cross-axis and noise density. However, it is important metric if high g applications are intended for the device. The following devices have comparably good performance: MPU6000, KXTC9, ICM20689, KXRB5. The harmonics are fairly apparent in this setup, however, the difference between the performances of the devices tested were relatively small - a spread of 3dB. This is due to the test setup introducing a fixed ± 1g sinusoidal input, which doesn't hit the boundaries of the typical $\pm 2g$ or $\pm 8g$ device range. With the current setup, there isn't a consistent method of varying the sinusoidal amplitude, making it difficult to test the linearity of the device across the input motion range. Despite this drawback, the linearity results still show a comparison between the devices since it does the relative performance of the devices in the $\pm 1g$ input, which is the region more important for inertial navigation applications.

3.4.4 Bias Variability

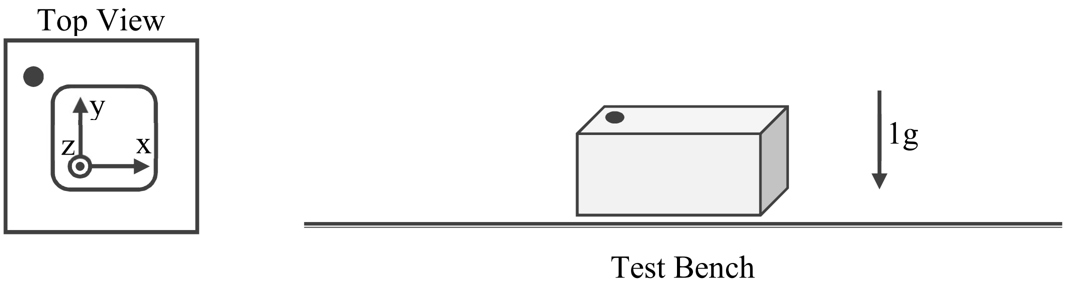


Figure 3-19 Bias Variability Dynamic Model

Noise modelling is an important aspect of research in MEMS inertial sensors. For inertial navigations, the effect of integration amplifies the effects of specific types of noise, thus making it important to understand the varieties of noise in the sensor. A typical noise model of a MEMS inertial sensor will break down the noise model into three categories.

1. White Noise
2. Zero Mean Flicker Noise (1/f)
3. Random Walk Noise (Bias Changes)

Through modelling, the noise is broken down into three different sources to illustrate the effects of each type. To illustrate the different effects of noise, a typical noise model of an accelerometer was constructed. These values were determined empirically by looking at typical values seen across the different DUTs.

- Noise Density: $200 \frac{\mu g}{\sqrt{Hz}}$
- Flicker Noise Power: 1/5th the power of the white noise.
- Random Walk Noise: $\int_0^t 10 \frac{\mu g}{s * \sqrt{Hz}}$, Random walk noise is an integral of white noise.

This noise model was simulated and shown in Figure 3-20, where white noise has the most impact in acceleration. However, when the results are integrated to determine velocity, as shown in Figure 3-21, Random Walk and Flicker noise have a much more significant impact, resulting in more than $1 \frac{m}{s}$ and $-0.5 \frac{m}{s}$ for each type of noise respectively.

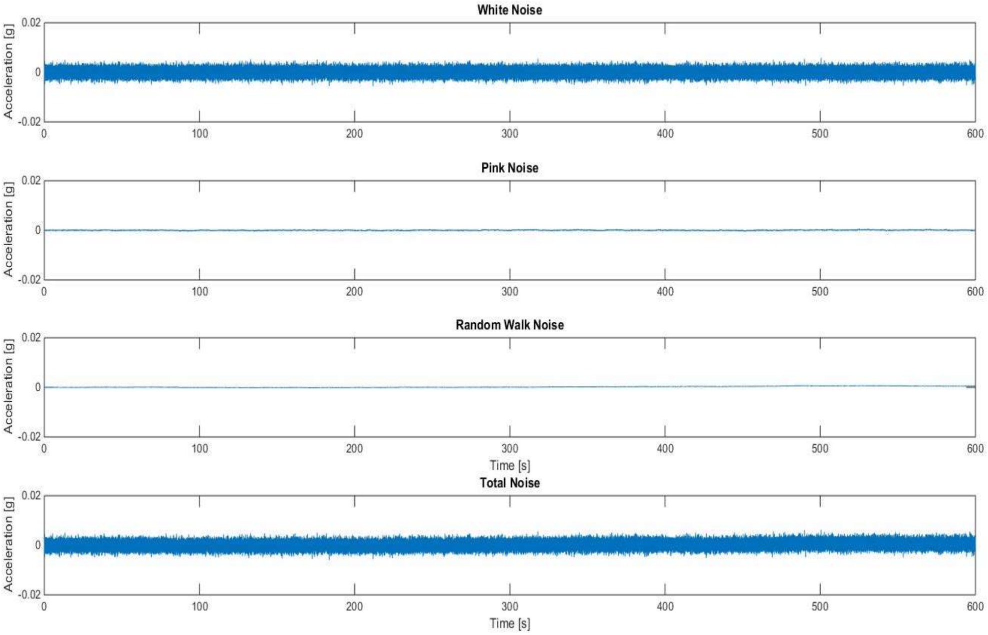


Figure 3-20 – Different types of noise seen on a typical accelerometer model.

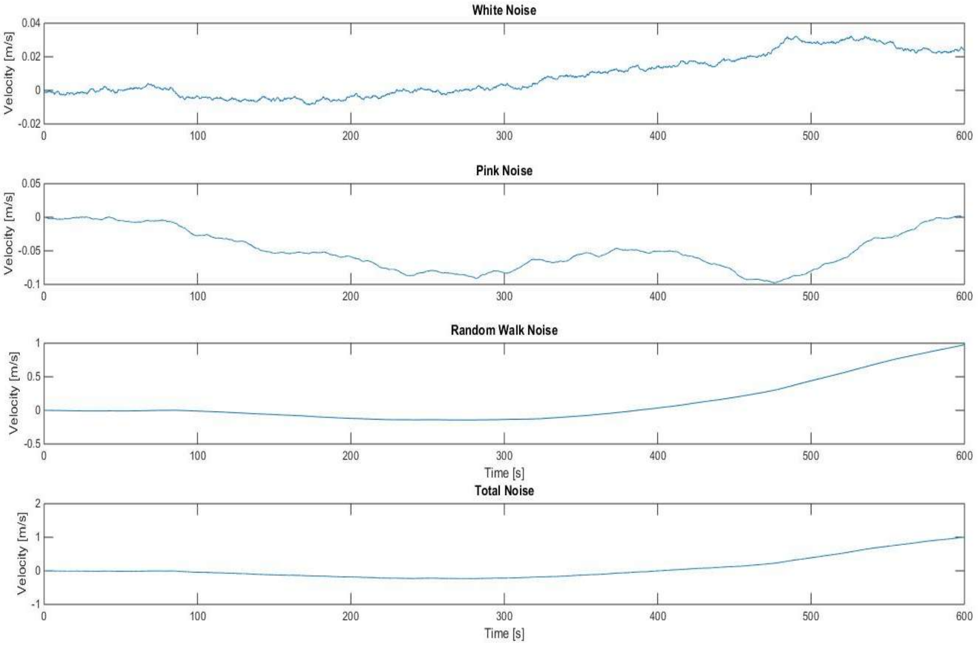


Figure 3-21 – Effects of integration on different type of noise seen in a typical accelerometer model

Therefore, the purpose of this test is to characterize and compare the performance of different devices across these other noise processes. The dynamic testing for this metric is similar to noise density; the device is strapped on to the test platform and held stationary. However, to ensure an accurate Allan variance plot can be constructed, the stationary data is recorded for 60 minutes, with the beginning 20 minutes truncated to minimize startup fluctuations.

3.4.4.1 *Allan Variance*

Allan variance *($AVAR$)* is commonly used to determine and distinguish the effect of noise in oscillators. There are two benefits to using this over a normal variance calculation method:

1. Not affected by non-stationary mean present in complicated noisy processes.
2. Allows differentiation of different regions of noise through a time based method.

There are two common methods of calculating $AVAR$, the first method is a non-overlapping method. It is calculated by averaging m number of consecutive data points, where m is chosen so that it can segment the dataset equally. For a dataset, x, with N data points, the averages, y, are calculated as follows:

$$y_i = \frac{1}{m} \sum_{j=(i+1)m}^{im} x_j \tag{3.10}$$

$AVAR$ is denoted with relation to a time period, which is related to the number of points averaged.

$$\tau = m\tau_0 = \frac{m}{f_S} \tag{3.11}$$

$AVAR$ is defined as the following:

$$AVAR = \sigma_y^2(\tau) = \frac{1}{2\left(\frac{N}{m} - 1\right)} \sum_{i=1}^{\frac{N}{m}-1} (y_{i+1} - y_i)^2 \tag{3.12}$$

Allan Deviation, which is more commonly used in the Allan Variance plots, is calculated simply by taking the square root:

$$ADEV = \sqrt{AVAR} = \sigma_y(\tau) \tag{3.13}$$

The non-overlapping method provides a quick way to determine the $AVAR$. At longer time periods, this calculation is less accurate due to a lower number of segments to calculate the first difference from. An overlapping Allan Variance will alleviate this issue.

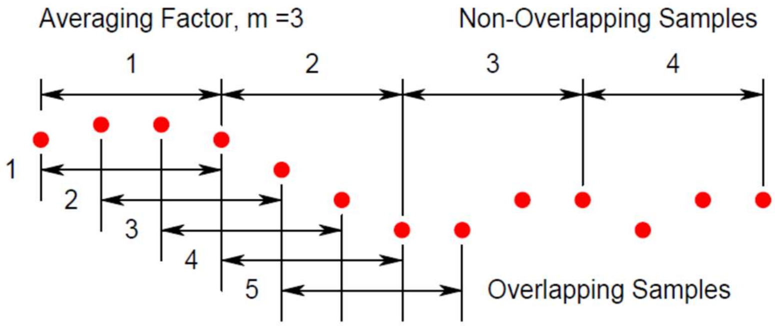


Figure 3-22 - Overlapping vs Non Overlapping Samples for Allan Variance. Image taken from NIST Handbook for Frequency Stability Analysis [19]

An overlapping Allan Variance is calculated using the following set of equations:

$$y_i = \frac{1}{m}\sum_{j=i}^{i+m} x_j \tag{3.14}$$

The Overlapping Allan Variance is calculated using the following:

$$\sigma_y^2(\tau) = \frac{1}{2m^2(N-2m)}\sum_{j=1}^{N-2m}\left\{\sum_{i=j}^{j+m-1} y_{i+m} - y_i\right\}^2 \tag{3.15}$$

Due to the computationally intensiveness of the inner loop, the Overlapping Allan Variance can also be estimated by first integrating x:

$$z_i = z_{i-1} + x_i, \qquad z_1 = 0; \tag{3.16}$$

Then the Overlapping Allan Variance can be estimated as

$$\sigma_y^2(\tau) = \frac{1}{2(N-2M)\tau^2}\sum_{i=1}^{N-2m}[z_{i+2m} - 2z_{i+m} + z_i]^2 \tag{3.17}$$

The Allan Deviation plot is obtained by sweeping across time periods and plotting the Allan Deviation vs τ in a loglog plot.

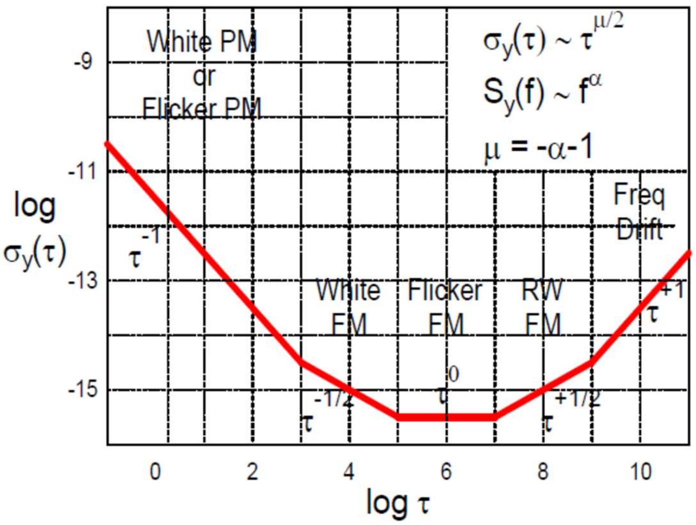


Figure 3-23 Slopes of difference types of noise in an Allan Deviation Plot. Image taken from NIST Handbook of frequency stability analysis [19]

To determine the metric, Allan Variance ($AVAR$) is used to compare between devices. In the case of accelerometers, there are three main processes of noise which are all captured by the Allan variance – White, Flicker and Random Walk. All three of these noise are seen with the following slope in the plot.

Table 3-8 - Types of noise seen on accelerometers and their slopes on Allan Deviation Plot

Noise Type	Slope (log Allan deviation)
White Noise	-0.5
Flicker Noise	0
Random Walk	0.5

To illustrate this, Figure 3-24 and 3-25 show the frequency domain plot and Allan Variance plot of the noise model that was illustrated in Figure 3-20.

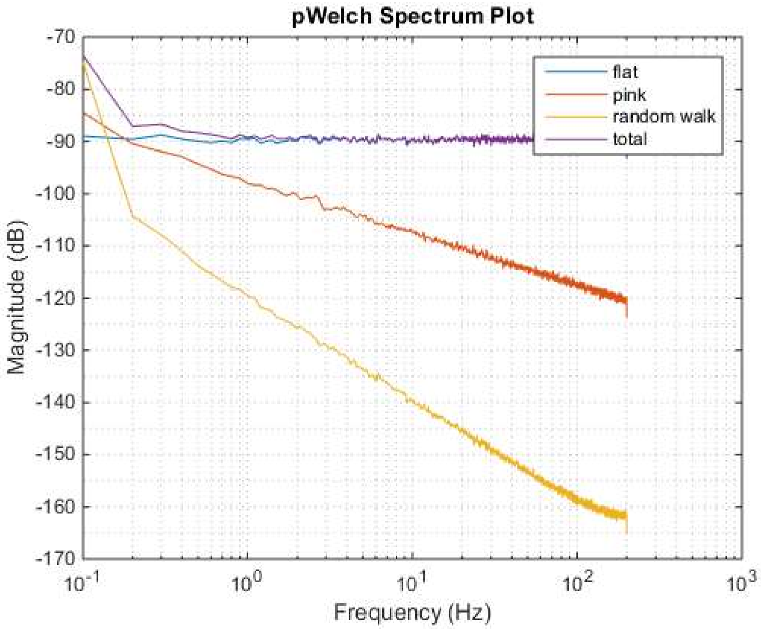


Figure 3-24 - Spectrum of the different type of noises. White noise is flat throughout. Pink noise exhibits a -10dB/decade. Random walk exhibits -20dB/dec and has significant impact at lower frequencies.

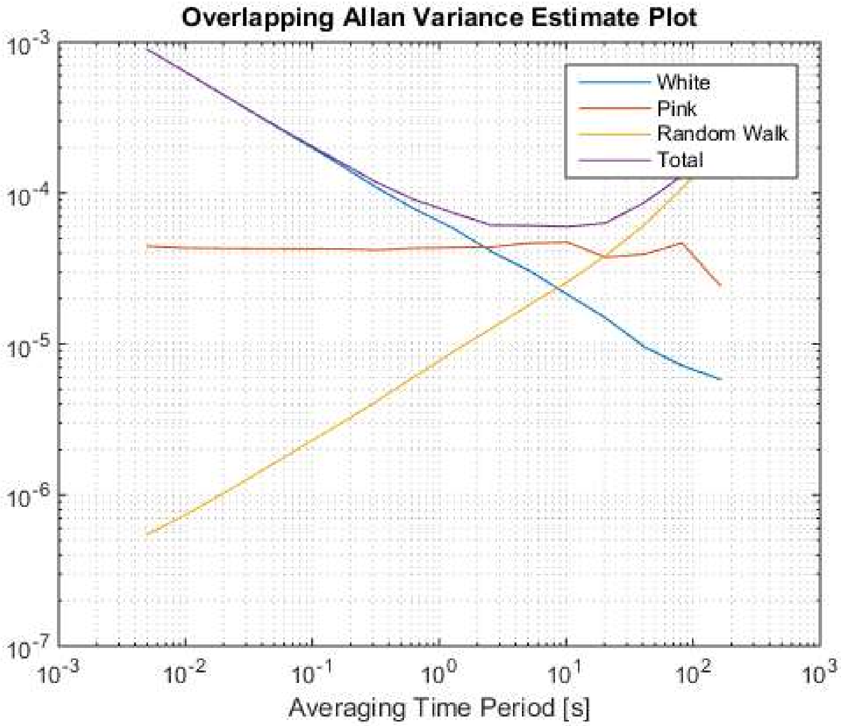


Figure 3-25- Allan Deviation Plot shows the noise in a different representation. On a loglog plot, white noise has a slope of
-0.5. Pink noise has a 0 slope. And random walk with a slope of 0.5.

3.4.4.2 *Bias Variability results and discussion*

Using the overlapping $AVAR$, the Allan deviation plot for each device was determined. From there, two different metrics are measured to determine the bias variability. The smallest time, where the slope is within 0 ± 0.05, was recorded along with the Allan deviation at that time. The Allan deviation measured at that time is best performance the device can theoretically achieve when the output is averaged for that time. The Allan deviation measured at the minimum can also be interpreted as random variable bias change.

For example, when Bias variability is specified at $30\mu g\ given\ \tau = 40s$

$@t = 0$: $Bias = B_t$

$@t = \tau$: $Bias = B_t \pm 1\sigma_y = B_t \pm 30\mu g$

Or it is interpreted as the bias becoming a random variable with a standard deviation of $30ug$. The following graphs are examples of how the Allan deviation plot appears for the devices measured.

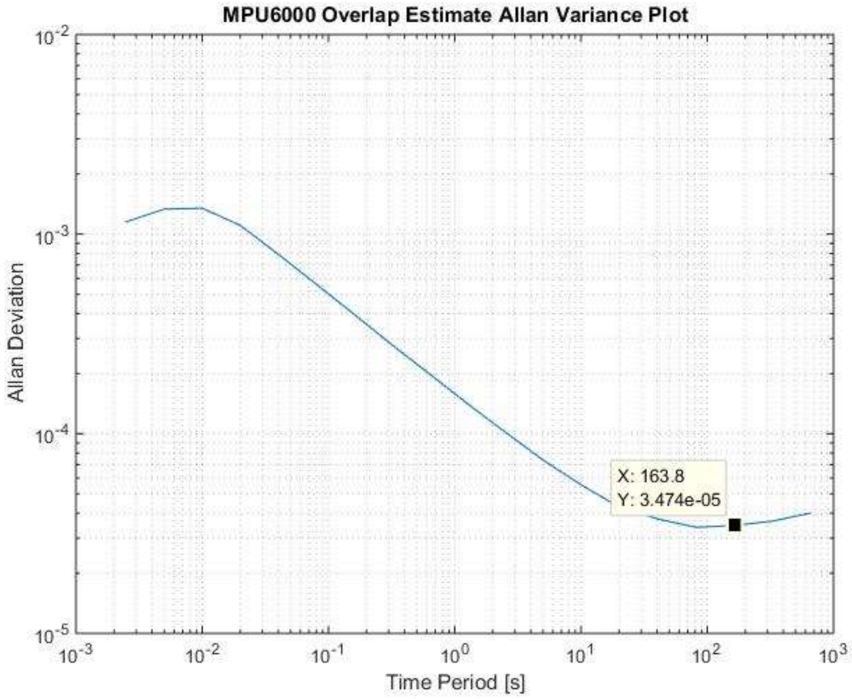


Figure 3-26 – Sample of MPU6000 Allan Deviation Plot

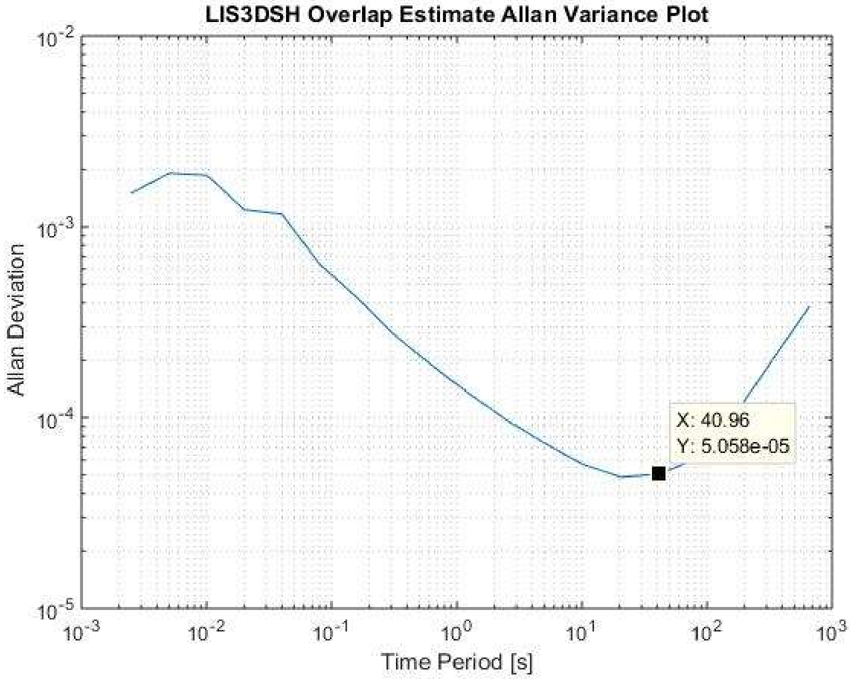


Figure 3-27 - LIS3DSH Allan Deviation Plot

Table 3-9 Bias Variability Results and Score

Device	Time Period $[s]$	$\sigma_y(\tau)$ $[\mu g]$	Bias Variability $[\frac{mg}{hr}]$	Score /100
LSM303D	41	76.6	6.72	11
LIS3DSH	41	50.6	4.44	17
AIS328DQ	41	64.2	5.64	13
MPU6000	164	34.7	0.76	100
KXTC9	82	101.8	4.47	17
ICM20689	41	61.3	5.38	14
KXRB5	82	66.9	2.94	26

Allan deviation isn't specified amongst all the device tested, and typically not performed. However, it gives the benefit of being able to account for other sources of noises other than white noise. Ideally, the Allan deviation is as low as possible, where the time is as large as possible. This indicates that the flicker and random walk noises are as small as possible. However, as it is difficult to compare 2 separate metrics, and to promote comparison between the devices, the Allan deviation was normalized to the time measured, and then normalized to mg/hr. This allows for an easy comparison as it accounts for the effect of the time and the Allan deviation at that time.

From the Allan deviation plot, there are two specific devices which stand out. MPU6000 has very good performance, showing a bias variability of $0.76\ \frac{mg}{hr}$, followed by KXRB5, showing a bias variability of $2.94\frac{mg}{hr}$. In the case of the MPU6000, it results from both a low Allan deviation and large time period, indicating lower effects of these types of noises. This method explores other type of noise processes not usually covered by manufacturers, but these types of noises are especially important for inertial navigation purposes where integration is often necessary.

3.4.5 Step Motion

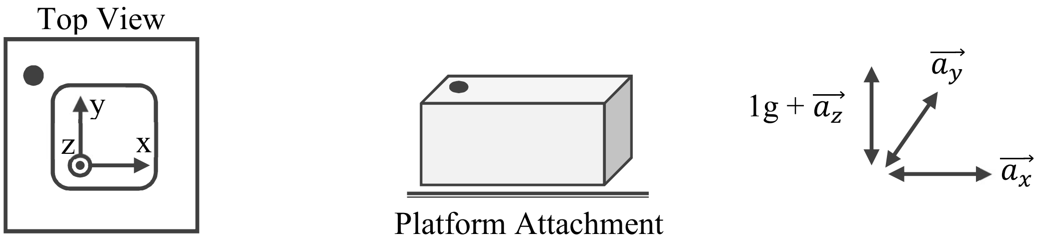


Figure 3-28 Step Motion Test Dynamic Model

The step motion test is a controlled representation of motion, and its purpose is to measure how the device performs at calculating displacement in a short time. This is a more complicated specification which tests more than the basic parameters of an accelerometer, but it utilizes the device in a similar way that it will be commonly used. The test moves the platform a fixed distance repeatedly and seeing how close the double integration of the results will be to the displacement moved.

This test will perform the following motion:

1. Move 1cm along one axis
2. Pause for 2 seconds
3. Repeat steps 1-2 10 times.

4. Repeat steps 1-3 5 times.
5. Repeat for other axes
6. Repeat test for 5 times.

The calculation of the metric will use the integration of the accelerometer. The displacement is calculated by double integrating the acceleration using the following formula:

$$\vec{d} = \iint_0^T \overrightarrow{a_{signal}} - k_{cal}\, dt \tag{3.18}$$

k_{cal} is the calibration constant from the 2 second pause which is used to remove the effects of the offset issues.

$$Step\ Motion\ Metric = \epsilon_{distance} = \left|\vec{d} - d_{ref}\right| \tag{3.19}$$

d_{ref} is the reference displacement of that axis.

The following figures are example plots for ICM20689 which show the acceleration and velocity plots of the motion.

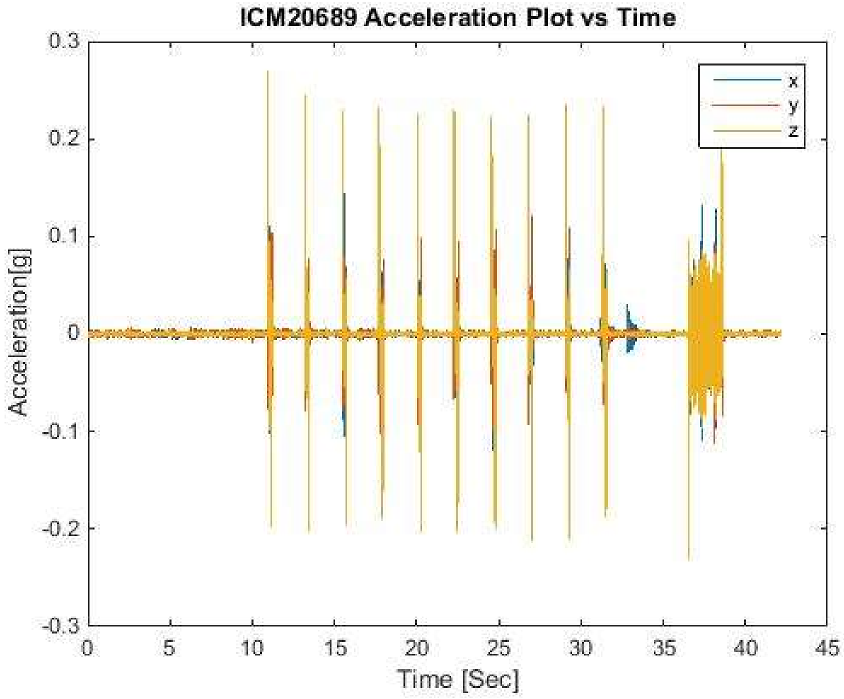


Figure 3-29 - Raw Acceleration for a z-azis step motion run

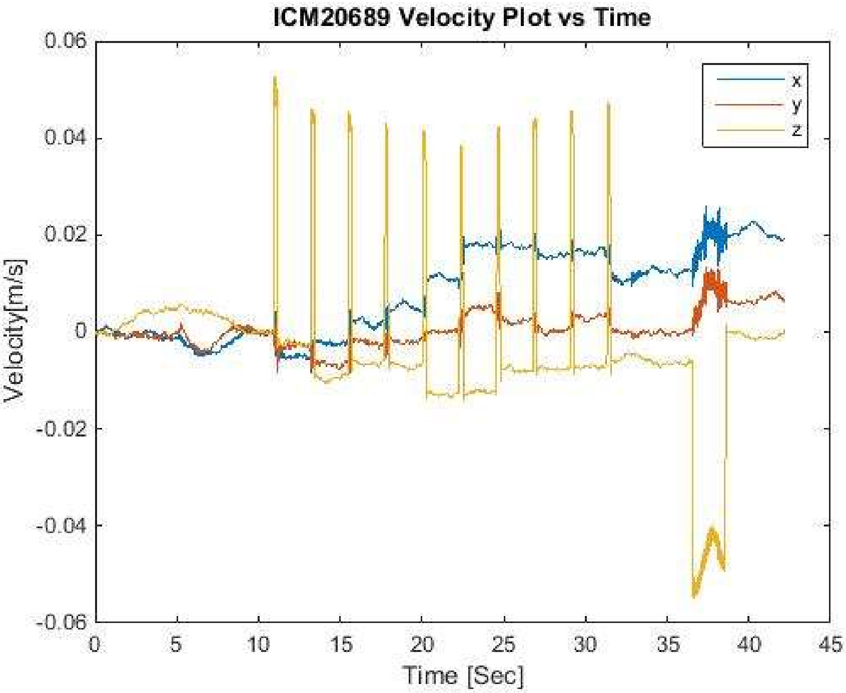


Figure 3-30 - Velocity of z-axis step motion from integration of acceleration. During periods of no motion, a zero velocity is ensured through the removal of the offset in the acceleration.

Due to removal of offset during periods with no motion, there is no change in velocity between the movement. However, the velocity still changes due to integration of a variety of errors during the motion, such as cross axis contamination or stochastic errors. Looking at the shorter motions in Figure 3-30, the velocity and duration matches closely with the expectation – the duration and amplitude results in ~1cm of displacement, and this is quite consistent across all the movement. As a result, the offset error of the velocity comes mainly from any errors during the movement, which results in a change in velocity before and after the spike. This error is more apparent when looking at longer movement durations, specifically at the 35-40 seconds. The velocity before and after the movement differs a lot. Figure 3-31 shows the velocity profile of AIS328DQ, where the movement spikes aren't always consistent, and this test will capture that. This test was meant to be help quantify some of the errors that were not covered in previous specifications and be a more realistic test as it is based on the movement and displacement of the device. If a device is prone to short term processes which get exacerbated by integration, such as cross axis or vibrations, this test will highlight it.

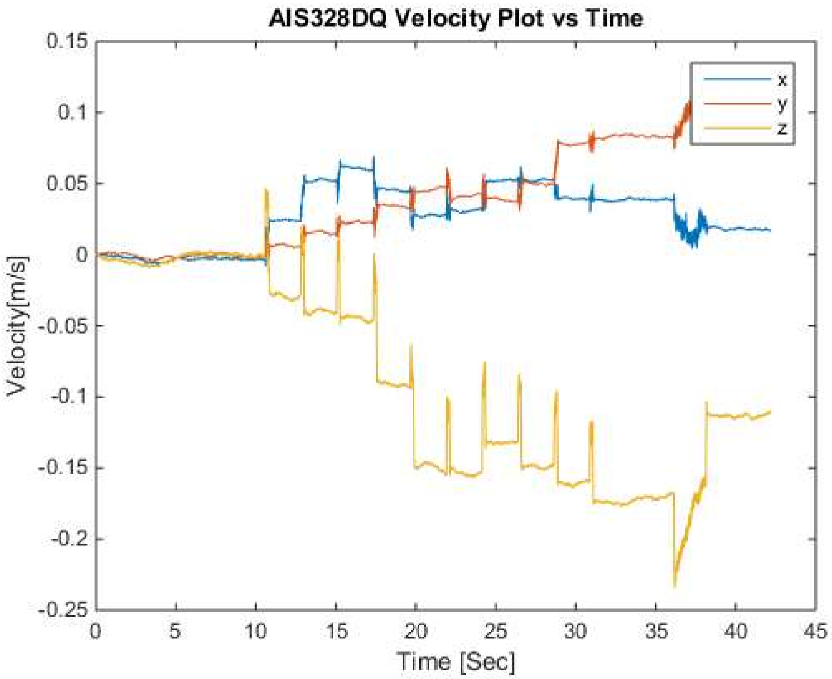


Figure 3-31 - AIS328DQ showing more significant errors during periods of motion, with the velocity changing significantly after motion.

By looking at the end displacement, it tracks how much the velocity has changed due to the errors accumulated by the motion. This test was repeated 5 times and the step motion error was averaged to get the following results.

Table 3-10 - Step Motion Results and Score

Device	**Step Motion Metric** [m]	**Score /100**
	Average	
LSM303D	0.47	23
LIS3DSH	0.25	44
AIS328DQ	1.74	6
MPU6000	0.22	50
KXTC9	0.11	100
ICM20689	0.39	28
KXRB5	0.20	55

There are two main contributions of this test:

- A specification that is similar to common use cases of accelerometers, and captures different aspects from the other specifications.
- Capture effects that are exacerbated by integration – calibration, noises, or in-motion attitude errors.

This test doesn’t narrow down the source of error, but it does show that some devices are more susceptible to errors that worsen with integration and provides a way of characterizing the performance of the different devices. In the step motion test, KXTC9 has the best performance, with a close performance from the KXRB5 and MPU6000. AIS328DQ and MPU6000 have similar noise, cross-axis and linearity performance, however, AIS328DQ has a much worse performance compared to the MPU6000. In another comparison, ICM20689 and KXRB5 both have a relatively good performance in the other specifications, but there is a larger difference when used in this test. The step motion test exercises the devices differently from the previous specifications and is necessary to consider when comparing accelerometers. It highlights difference between devices with similar noise performance when they are moved and integrated, which is extremely important as accelerometers are commonly integrated in inertial systems.

3.5 Device Specification Summary

There were two problems that this chapter seeks to address – inconsistencies at comparing accelerometers across different evaluation platforms and standard testing methodologies on a low-cost platform that will allow comparison of different accelerometers. The construction of a common testing platform provides the ability to effectively compare between different devices tested and any future devices. When investigating noise density, which is a very common metric, devices did not match specification and appeared to use an improper method of calculation. For other metrics, there were many situations where not all the devices were specified. In the process of defining the testing methodology, we have shown a set of testing methodologies which provide a consistent and stable way to analyse and compare accelerometers. This section has clearly defined 5 distinct metrics which are measurable using a low-cost 3-D printer to effectively compare between accelerometers and their performance.

1. Noise Density
2. Cross Axis
3. Linearity
4. Bias Variability
5. Step Motion

The first three specifications are strictly defined to ensure comparability between the different devices, and bias variability and step motion are static and dynamic tests that investigate areas that are not covered by the other common specifications. Throughout this definition, the development of a 3D printer based accelerometer testing and measurement platform enabled a consistent comparison. Along with a strict methodology to test and measure the devices, it becomes possible to also interpolate performance in key application areas. An earlier example mentioned is when an accelerometer is used for dead-reckoning purposes. In that scenario, the accuracy of the displacement calculated is the priority. In that case, specifications such as the bias variability and step motion will be more significant than noise density, cross-axis and linearity. Which results in a weighting system to compare between different devices. However, this is only an example weighting and the metric weightings will change depending on the application.

Table 3-11 Scoring Matrix for dead-reckoning scenario

Device	Noise Density	Cross Axis	Linearity	Bias Variability	Step Motion	Final Score
Weighting	5	3	3	7	7	25
LSM303D	23	53	92	11	23	7.9
LIS3DSH	32	67	90	17	44	10.6
AIS328DQ	54	32	86	13	6	7.6
MPU6000	47	47	95	100	50	17.1
KXTC9	100	67	95	17	100	18.1
ICM20689	69	80	99	14	28	11.8
KXRB5	86	100	100	26	55	16.0

Using the example weighting, KXTC9, MPU6000, and KXRB5 will be the better choice for applications which need better dead reckoning performance. The decision of the weighting is highly dependant on the application, and its selection is outside of the scope of this study. However, using this example, it highlights a method of using the result of a standardized testing methodology and how it can aid in selecting accelerometers for different application.

Chapter 4 Specification Impact on Inertial Algorithms

There are extensive studies that explore different inertial algorithms and their implementations. There are also studies that explore a variety of applications for a single algorithm. This section explores the impact that specification of a MEMS accelerometer would have on the performance of an inertial navigation system (INS) in estimating location and attitude.

This section will approach this by exploring and evaluating common methods used in improving accuracy of inertial navigation systems. Afterwards, we will focus on specific scenarios and applications that commonly utilize an INS and evaluate the performance of a variety of accelerometers. Finally, by analyzing the performance of the different accelerometers in these scenarios, and discussing this in context of the previous chapter, it will provide insight on the impact of accelerometer specification on the performance of an INS.

4.1 Inertial Algorithms

4.1.1 Attitude errors on INS performance

As established in Chapter 1, there are a variety of studies considering inertial algorithms involving external references or a combination of multiple sensors. This section will cover some of the common inertial algorithms used in attitude and location estimation. One of the significant error in an INS is due to attitude inaccuracies which results from contamination of the acceleration measured by gravity [13]. Figure 4-1 illustrate a scenario where there is a fixed attitude error due to a rotation about the y axis. This results in Eqn 4.1, where the rotation will cause gravity to be seen on the measurement axis – $\overrightarrow{a_x}$.

$$\begin{bmatrix} \overrightarrow{a_x} \\ \overrightarrow{a_z} \end{bmatrix} = \begin{bmatrix} \cos(\theta_y) & -\sin(\theta_y) \\ \sin(\theta_y) & \cos(\theta_y) \end{bmatrix} \begin{bmatrix} 0 \\ \overrightarrow{f_g} \end{bmatrix} \tag{4.1}$$

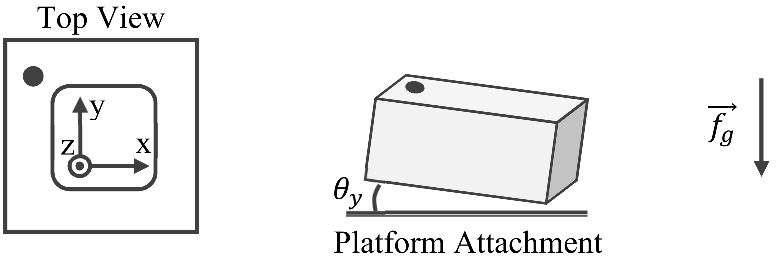


Figure 4-1 - Dynamics of a 1 dimensional attitude error of a stationary accelerometer

Thus, a portion of gravity will be seen in the measurement frame of the sensor. If the device is stationary, calibration is done to measure this stationary error and account for it by rotating the measurement axes. The scenario is more complicated when there is an external movement, and the attitude is no longer stationary.

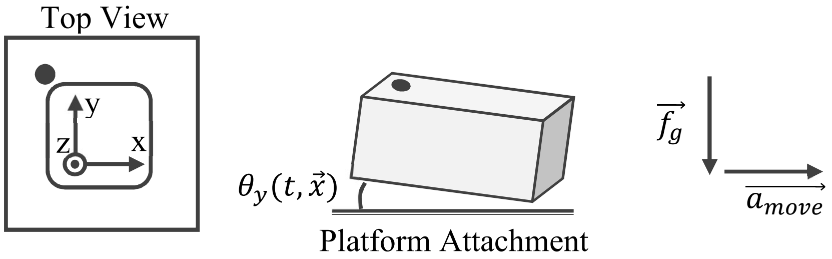


Figure 4-2 – Dynamics of an attitude error on an accelerating accelerometer

$$\begin{bmatrix} \overrightarrow{a_x} \\ \overrightarrow{a_z} \end{bmatrix} = \begin{bmatrix} \cos(\theta_y(t,\vec{x})) & -\sin(\theta_y(t,\vec{x})) \\ \sin(\theta_y(t,\vec{x})) & \cos(\theta_y(t,\vec{x})) \end{bmatrix} \begin{bmatrix} \overrightarrow{a_{move}} \\ \overrightarrow{f_g} \end{bmatrix} \tag{4.2}$$

Although the governing equations and dynamics are still the same, this problem is much more challenging due to the difficulty of getting an accurate measurement of the attitude change. The acceleration from the movement of the device $\overrightarrow{a_{move}}$ will now cross-contaminate with gravity, affecting the measurements seen at the sensor. In practice, the challenge lies in determining the attitude of the system accurately and in real time.

This attitude error can result from device offset, alignment issues, or attitude changes resulting from the movement. Alignment errors manifests itself as a fixed attitude error and arises from a variety of sources, such as the ones listed below [5]:

- sensing element and package alignment

- chip and board alignment
- board and system alignment

These sources are unavoidable in practice, as there are always alignment errors during fabrication or setup. Since these sources have a stationary nature, they consistently affect the measurement axis in the same way, they can be calibrated and accounted for.

On the other hand, attitude errors resulting from movement and vibrations cause time varying attitude changes which are difficulty to accurately measure. An example of this is the effects of suspension system on a subway train car. A recent study showed that the suspension leads to attitude changes resulting from people within the train car and from the movement of the train [6]. This is the main challenge of inertial algorithms, as it becomes a necessity to estimate the attitude change accurate to compensate for the effects of gravity on the measured acceleration. The error is even more apparent when looking at applications which rely on calculating velocity or displacement, where this error gets further magnified by integration.

To illustrate the importance of attitude estimation, a $0.5°$ error in was introduced in Eqn. 4.1.

$$\begin{bmatrix} \overrightarrow{a_x} \\ \overrightarrow{a_z} \end{bmatrix} = \begin{bmatrix} \sin\left(\frac{2\pi}{360} 0.5°\right) \\ \cos\left(\frac{2\pi}{360} 0.5°\right) \end{bmatrix} = \begin{bmatrix} 0.0087 \\ 0.9999 \end{bmatrix}$$

That results in a contamination of $\sim 0.009g$ on the measurement axis due to gravity. For perspective, a typical subway will hit around a short term sustained acceleration of $0.1g$ over when it is accelerating [6]. This means that a small angle change causes a $\sim 10\%$ difference in the measurement. This attitude error is exacerbated by the fact that the accelerometers measure acceleration. When it is integrated to look at velocity or displacement, this small attitude error has a significant impact.

Using $\Delta t = 60s$,

$$\Delta v = |a| \Delta \mathrm{t} * 9.81 \frac{\mathrm{m}}{\mathrm{s}^2} = 5.1 \frac{m}{s}$$

$$\Delta d = |a| \frac{1}{2} \Delta t^2 * 9.81 \frac{m}{s^2} = 153.6m$$

To simulate this change, the accelerometer model used in Chapter 3 was rotated by 0.5° and simulated for a runtime of 60 seconds, showing comparable errors to the calculations.

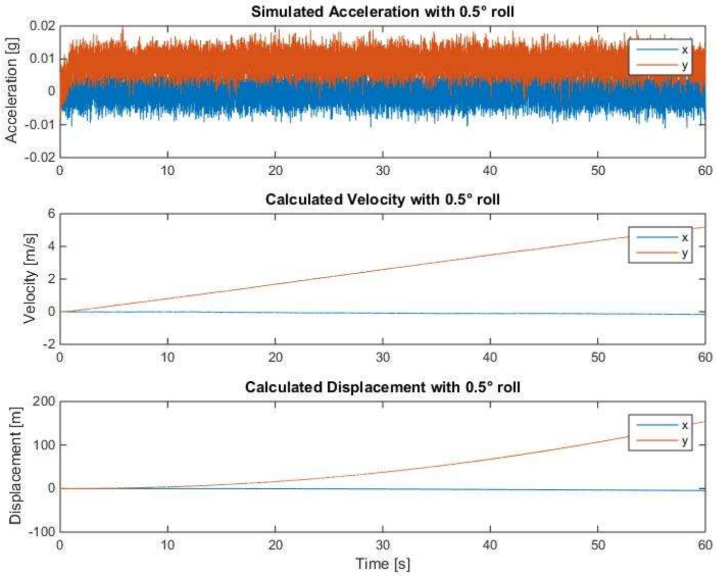


Figure 4-3 – Acceleration, Velocity and Displacement of a $0.5°$ *attitude error.*

The results of these attitude errors will manifest itself as an offset/bias error in the measurement axis of the accelerometer, as shown in Figure 4-11. To address this, a previous study showed a post-processing bias estimation method which resulted in final displacements deviating by 10% over 80 seconds of motion. This method uses a single accelerometer, and zero velocity detection techniques in combination with a gradient descent bias estimation algorithm. [6] This study showed a significant improvement over raw integration of the accelerometer, however there is a significant drawback of not being able to run in real time and relies upon knowledge of motion dynamics. For most practical purposes, this post-processing technique will not be useable, however, it highlights the potential of accelerometer usage in an INS without external references.

To start, we explore two basic methods of calculating attitude changes – accelerometer calculation and gyroscope integration. This is followed by a discussion of well-known sensor fusion algorithms, and their implementation.

4.1.2 Basic Attitude Estimation

In a stationary accelerometer setup, the magnitude of the acceleration is constant. Since the magnitude is constant, the accelerometer can be used to determine the attitude of the system. Using only an accelerometer, and the constant magnitude of gravity, it is possible to determine two of the three angles needed for a complete estimation of the attitude, as shown by Eqn 4.4 and 4.5.

$$|\vec{a}| = \sqrt{\overrightarrow{a_x}^2 + \overrightarrow{a_y}^2 + \overrightarrow{a_z}^2} = 1g \tag{4.3}$$

$$roll = \beta = \theta_x = sign\left(\frac{\vec{a}_y}{\vec{a}_z}\right) * \arctan\left(\sqrt{\frac{|\vec{a}_y|^2}{|\vec{a}_x|^2 + |\vec{a}_z|^2}}\right) \tag{4.4}$$

$$pitch = \gamma = \theta_y = sign\left(\frac{\vec{a}_x}{\vec{a}_z}\right) * \arctan\left(\sqrt{\frac{|\vec{a}_x|^2}{|\vec{a}_y|^2 + |\vec{a}_z|^2}}\right) \tag{4.5}$$

In contrast to an accelerometer, a gyroscope is a MEMs device which measures the angular velocity of a system. It is typical for a gyroscope to be a 3-axis gyroscope, which gives the ability to detect angular change across the three axis – roll, pitch, yaw. In a single gyroscope setup, it is possible to determine the angle of the system by integrating the angular change over the duration of the rotation.

$$roll = \beta = \theta_x = \int_0^t \omega_x dt \tag{4.6}$$

$$pitch = \gamma = \theta_y = \int_0^t \omega_y dt \tag{4.7}$$

These basic methods are not very useful in attitude estimation due to their individual drawbacks when used in a practical environment. In the case of using the accelerometer only to determine the attitude, the attitude calculation is susceptible to movements and vibrations and the noisy nature of the accelerometer. In addition, calculating the angle using an accelerometer will not give an indication of the yaw.

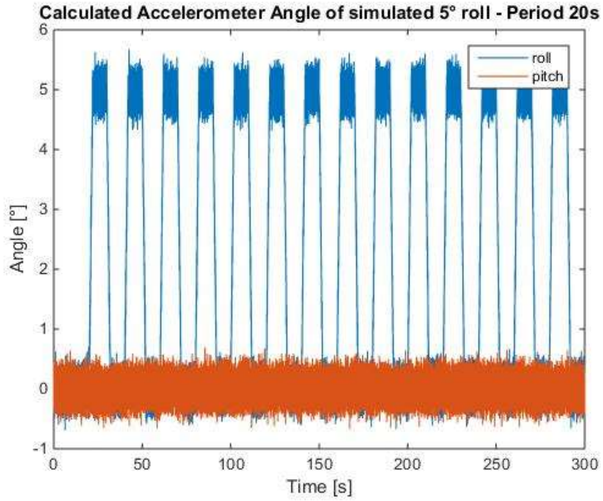


Figure 4-4 – Attitude estimation using accelerometer only. Angle fluctuations of ± $\mathbf{0.5}$° *are seen.*

Figure 4-3 highlights the noisy nature of the attitude estimation through an accelerometer. In contrast, attitude estimation using a gyroscope isn't affected by noise or vibrations, but rather the integration of noise – drift.

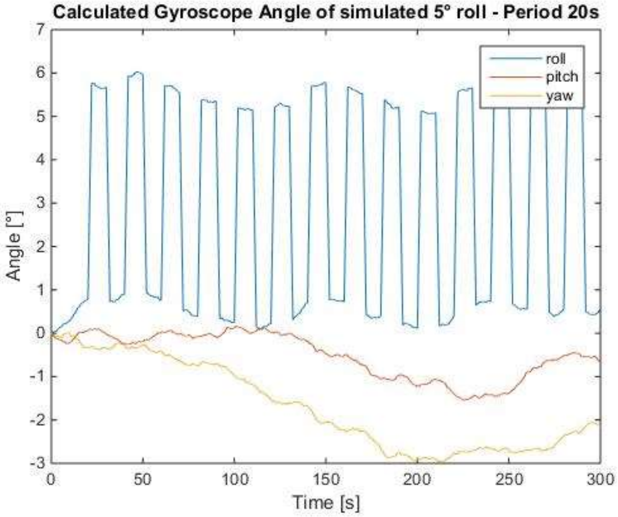


Figure 4-5 -- Attitude estimation using gyroscope only. Angle drifts by 0.4-2° over 5 minutes.

Basic attitude estimation methods are inadequate due to their respective drawbacks of noise and drift. As such, it is common to utilize both of these sensors in conjunction to improve the attitude estimation. Other inertial algorithms use a magnetometer as well to provide an even more accurate attitude estimation. However due to measurement platform limitations which prevent a magnetometer from analog testboards, it was excluded from all test boards. Thus, only inertial algorithms which use an accelerometer with a gyroscope is included in the scope of this study.

Regardless, there are a number of situations where a gyroscope and accelerometer combo is used to determine attitude. This study focuses on determining the impact of accelerometer specification on their performance in common inertial algorithms. As such, the gyroscope is the same across these inertial tests. MPU6000 from Invensense will be utilized throughout the tests as a way of providing consistent gyroscope measurements. Two of the accelerometers, LIS3DSH and LSM303D are also excluded from this testing due to their noise issues during the higher sampling rates. The next two sections will go over the implementation of more complicated attitude estimation algorithms used in this study to combine the accelerometer results with the gyroscope.

4.1.3 Complementary Filtering

The complementary filter is a common sensor fusion method used for the attitude estimation for gyroscopes and accelerometers. This method is popular due to its simplicity; the filter has a single parameter which can be tuned to change the cross-over frequency of the filter.

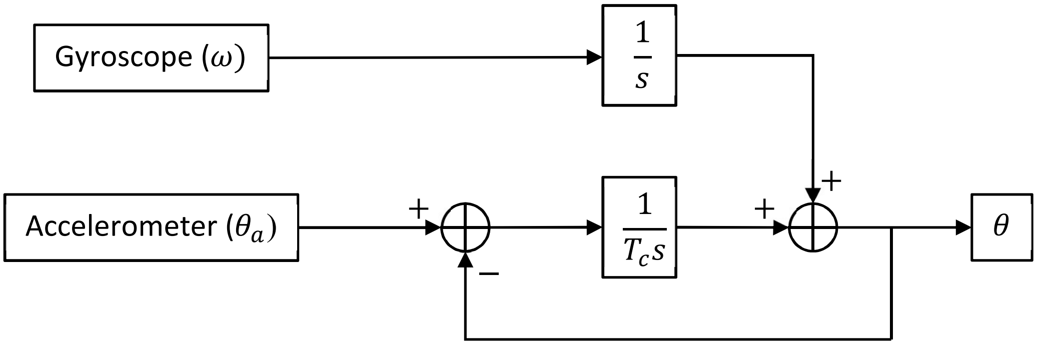


Figure 4-6 - Classic Complementary Filter Block Diagram

This filter structure will result in low-passing the accelerometer and high-passing the gyroscope. This is particularly useful because gyroscope estimates of attitude introduce low frequency drift, where as the accelerometer is susceptible to high frequency movements.

Solving Figure 4-5 for the output,

$$\theta(s) = \frac{1}{1 + T_c s}\,\theta_a + \frac{T_c s}{1 + T_c s}\,\omega\left(\frac{1}{s}\right) = \frac{\theta_a + T_c\omega}{1 + T_c s} \tag{4.8}$$

Using the backwards difference equation,

$$s = \frac{1}{\Delta t}(1 - z^{-1}) \tag{4.9}$$

And a change of variables as defined below,

$$\frac{\Delta t}{T_c} = \frac{1 - \alpha}{\alpha} \tag{4.10}$$

The discrete time equation for the complementary filter is expressed as follows:

$$\theta_k = \alpha(\theta_{k-1} + \omega_k \Delta t) + (1 - \alpha)\theta_{a_k} \tag{4.11}$$

Δt, is the sample time.

α, is the tunable parameter.

The tuning parameter is empirically decided as it depends on the inertial sensors used and the application of the system. For our testing, it was empirically chosen that an α of 0.97-0.98 worked best in estimating the attitude with minimal drift. Solving Eqn 4.10 gives a cross over frequency of around $\frac{f_s}{49} \rightarrow \frac{f_s}{32}$ which is around $8Hz - 12.5Hz$.

4.1.4 Kalman Filtering

The Kalman filter is another useful sensor fusion algorithm for attitude estimation. However, the filter needs to be customized to combine the gyroscope and acceleromter. The following equation defines the state dynamics of the system; it is shown in a similar form to equation 2.6 to highlight the state variable and the control variable.

$$\begin{bmatrix} \theta \\ b \end{bmatrix}_{k|k-1} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ b \end{bmatrix}_{k-1|k-1} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \omega_k \tag{4.12}$$

$\begin{bmatrix} \theta \\ b \end{bmatrix}_k$, is the state variable tracked by the Kalman filter. θ tracks one of the three attitude axes and b, tracks the bias or offset of the gyroscope

ω_k, is the current angular change measured by the gyroscope

The gyroscope measurement is used as the control variable to update the attitude dynamics. From equation 4.12, the initial values of the state variables are needed. In this case, the initial angle and bias of the gyroscopes are determined during the calibration stage. This process can introduce errors as the calibration stages aren't completely accurate, however, this only affects the transition time before the Kalman filter recursively adjusts the parameters. This was by holding the system motionless for s short time before each experiment movement was initiated.

In addition to the initial values of the states, the covariance matrices needed initial values as well. This used a large value and let the filter recursively determine them. Next, the measurement (z_k) uses the attitude calculated from the accelerometer measurement. This is then used to determine the residual in equation 2.8.

The noise covariance matrices (Q_k, R_k) used in the covariance prediction and update stages are hard to determine and model, and one of the difficulty of using Kalman filters. In this case, they are assumed to not be time-dependant to simplify the system, simplifying equations 2.7 and 2.9 to the following form.

$$P_{k|k-1} = F_k P_{k|k-1} F_k^T + Q \tag{4.13}$$

$$S_k = H_k P_{k|k-1} H_k^T + R \tag{4.14}$$

From equation 4.13 and 4.14, Q and R are the tunable matrices which are used to control the characteristics of the filter. R is used to control the amount of noise seen from the accelerometer. For noisier accelerometers, a higher R value can be used so that the measurement is trusted less and the gain will be geared more towards the dynamics. Q is the noise covariance of the dynamics, and tuning this adjusts for the noise from the gyroscope. For our testing, the Q matrix was kept consistent and the R matrix was tuned for each device and fixed for the experiments

4.2 Attitude Estimation Evaluation

Attitude estimation is a key application of an INS. It is important for removing attitude errors from location estimation, but it is equally as important in robotics and mechatronics, where it is used to determine the orientation of systems. This section will conduct tests which will test the two-sensor fusion based attitude estimation algorithms and evaluate their performance in measuring attitude.

4.2.1 Attitude Estimation – Static change Test

The first test focuses on the performance of in tracking attitude change where there is a short duration of motion. This test will also minimize vibration and movements except for the movement needed to change the angle of the platform. This setup requires human intervention to change angle, so the performance during the angle change isn't evaluated, but rather the accuracy in determining the angle change after it has stabilized. Attaching an adjustable mitre gauge to the stationary bench, the following procedure was used for testing.

1. System is mounted to a stationary platform
2. System is kept still at 0° for 10 seconds to use as calibration.
3. The gauge is move to the next angle position and locked in. This movement will take 10 seconds of time.
4. System is kept still at the angle for 10 seconds.
5. Steps 3-4 is done for the following angles
    - $0°, 15°, 30°, 45°, 30°, 15°, 0°$
6. Steps 1-4 are repeated for the different axis.

The performance is measured by keeping track of the accuracy of the angle changes – 15° per change. To calculate the angle change, the angle in the middle of the stationary time, where it isn't affected by the movements, were used to calculate the attitude changes. The test will measure performance by keeping track of $\Delta° \pm \sigma$, where a small Δ, σ indicate a better performance. Δ refers to the average attitude error seen, and the σ is the deviation of this error. Considering a 0.5° was shown to have a significant impact in a long duration, it is ideal to keep the average error and deviation low.

Table 4-1- Static change test results across method and device.

	Accelerometer Only [°]	Complementary Filter [°]	Kalman Filter [°]
AIS328DQ	0.468 ± 0.545	0.211 ± 0.239	0.210 ± 0.238
MPU6000	0.108 ± 0.183	0.101 ± 0.153	0.103 ± 0.153
KXTC9	0.114 ± 0.132	0.101 ± 0.122	0.102 ± 0.122
ICM20689	0.129 ± 0.139	0.122 ± 0.133	0.122 ± 0.132
KXRB5	0.122 ± 0.134	0.077 ± 0.098	0.078 ± 0.096

From this test, there were two main points of discussion:

1. Similar performance of the two inertial algorithms.
2. The ability of using noise density to compare device performance

Both sensor fusion based algorithms have similar performance in this test. The similarity between the two inertial algorithms are unexpected. In this case, this similarity is due to the setup of this test which reduces external movements and has a shorter test time. As shown in the next test, the difference gets exacerbated when more movements are involved.

The second area of discussion arises from analyzing the results of this test and considering specification performances. From the specification testing and the weighted comparison, AIS328DQ was shown to be one of the worst device tested. This test shows a similar result compared to the weighted analysis of the specification. However, it becomes interesting when comparing two relatively similar devices from a noise density stand point. AIS328DQ has a comparable noise density to MPU6000, but their performance in this test is very different. Another demonstration of this is in KXTC9, which has the best performance in noise density specification, but is not the best performing device after inertial algorithms are applied. This is an indication that noise density, one of the more documented and important specification listed across MEMS accelerometers, are not a good indication of performance once inertial algorithms are applied.

4.2.2 Vibration affected static angle test

In real applications, mechanical noise and sources of disturbances exists. This test will test attitude estimation again, but it will introduce random vibrations and movement from the 3D printer during the test to simulate a noisy environment.

1. System was connected to the 3-D printer.
2. System was kept stationary for 10 seconds at 0° for calibration purpose
3. System was changed to 15°
4. 3-D printer starts to perform random vibrations and movements for 40 seconds
5. Steps 1-4 was repeated for every device

The mean and deviation was evaluated during the period where the system was being moved by the 3-D printer. The metrics calculated are shown below:

Table 4-2 - Summary of Attitude estimation when affected by external perturbations

	Accelerometer Only [°]	**Complementary Filter [°]**	**Kalman Filter [°]**
AIS328DQ	14.365 ± 8.344	14.353 ± 0.976	14.369 ± 0.914
MPU6000	15.037 ± 2.521	15.036 ± 0.468	15.036 ± 0.426
KXTC9	14.891 ± 2.599	14.886 ± 0.450	14.894 ± 0.409
ICM20689	15.027 ± 2.569	15.024 ± 0.310	15.030 ± 0.284
KXRB5	14.862 ± 2.568	14.854 ± 0.251	14.862 ± 0.230

There are three main areas of discussion from this test.

1. Differences between Kalman and Complementary filter
2. Importance of utilizing an inertial algorithm.
3. Device specific results.

The Kalman had consistently lower deviation when compared to the complementary result. The difference is small, accounting for $\sim 0.05°$ difference between the deviation performance, however, considering the significance of a small attitude error, this can yield differences when looking at dead-reckoning scenario.

This test also highlight the importance of using an inertial algorithm. The overall performance of post filtering is much better than using only the accelerometer. Utilizing a 6-axis device which includes a gyroscope and an accelerometer, such as the MPU6000, it is much more beneficial from a performance and cost perspective when compared to utilizing KXRB5 directly. In addition, using the inertial algorithms improves the ability of the INS when there are random vibrations and movements, as illustrated by figure 4-6 and 4-7. There is a significant deviation of the angle calculated and cannot be used directly when no inertial algorithm is used.

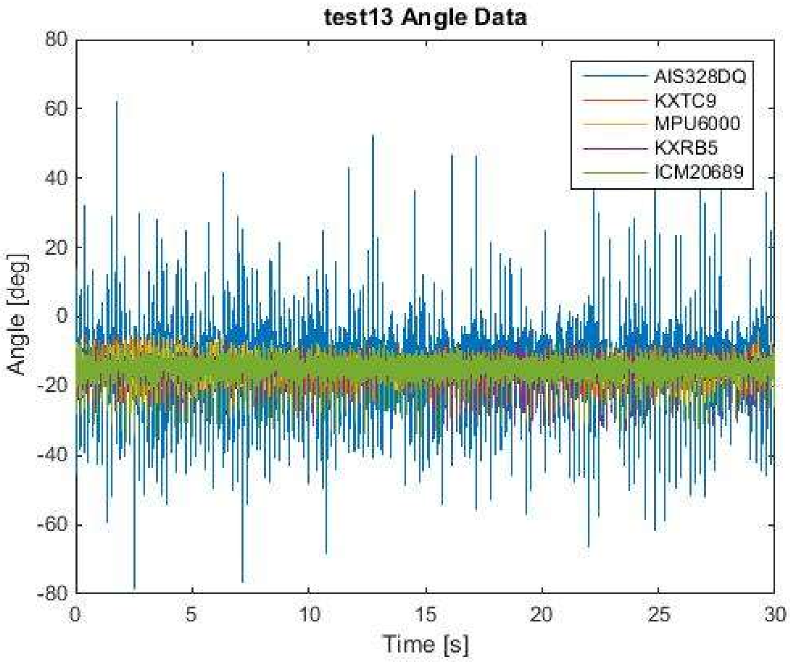


Figure 4-7 - Attitude calculation using accelerometer only when affected by vibrations.

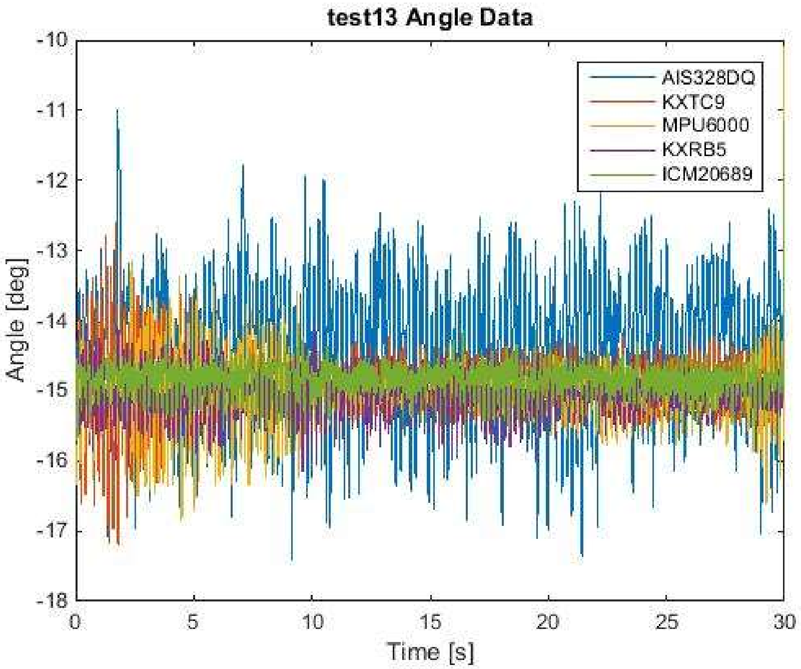


Figure 4-8 - Attitude Estimation using Kalman Filter when affected by vibrations

In regards to device specific performance, AIS328DQ is an interesting outlier. Despite a similar motion utilized for all the tests, AIS328DQ's results are highly affected by vibrations, as shown by the higher deviation compared to the other devices pre-processing. This was also noted with two different units of the same chip, and this susceptibility to vibrations and external movements is the reason why this device consistently underperforms the other devices throughout the inertial tests. Excluding the AIS328DQ, each device shows a similar level of deviation before the processing, but after inertial algorithms are applied, there was a difference in performance between the devices. Approximately 0.2 °rms difference was seen between MPU6000 and KXRB5. The KXTC9 has one of the better performances when it comes to noise density, but in this test, it has very minimal impact on the accuracy of attitude estimation. On the other hand, cross-axis is a much better specification to look at. It is illustrated by MPU6000/KXTC9 and ICM20689/KXRB5, where the both group of accelerometers have comparable cross-axis and performance in this test. The latter group of ICM20689/KXRB5 show a better cross-axis ratio and thus performance in this movement based attitude estimation test. Cross-axis performance is important in this scenario where the inertial sensors are moved and vibrated. When the sensors are vibrated, and moved, contamination of the movement to different axis will affect the attitude estimation by adding errors which are not directly caused by the attitude change, and not as easily removed by filtering. When a device is less affected by cross axis errors, it will have a reduce the impact of linear acceleration contamination on attitude estimation.

4.3 Dead Reckoning Tests

Dead reckoning is the ability of an INS to determine acceleration, velocity and displacement. The purpose of this section is to explore and test the ability to improve acceleration measurement and thus velocity calculations after attitude removal techniques are applied.

4.3.1 Processing Model

Attitude estimation is important to the removal of the effects of on velocity calculations in dead reckoning purposes. However, attitude estimation cannot be used directly due to limitations from only combining a gyroscope and an accelerometer. This leads to an inability of the inertial algorithms to differentiate between very slowly changing accelerations and attitude changes. This is typically mitigated by incorporating a magnetometer which can be used to determine an attitude that is not affected by slow movement. However, due to the limitations of the daughter boards created, this was not an option in this study. This limits the ability to separate slow moving accelerations and attitude changes, and thus it will affect gravity removal during motion. To mitigate this effect, when extended periods of acceleration are detected, the gyroscope is without the accelerometer to predict the attitude change during the transient time.

The second limitation arises from the inability of the attitude estimation algorithms to determine the final attitude change – Yaw. This can also be solved with the addition of magnetometer, which can provide the yaw angle. Due to this limitation, if a yaw attitude change is introduced during these tests, the system will be unable to estimate this without drift from the accelerometer. This issue was mitigated by introducing motion which would not introduce a yaw attitude change.

With these considerations, the algorithm outlined by Figure 4-8 is used in these Dead-Reckoning tests to determine the acceleration, and subsequently the velocity.

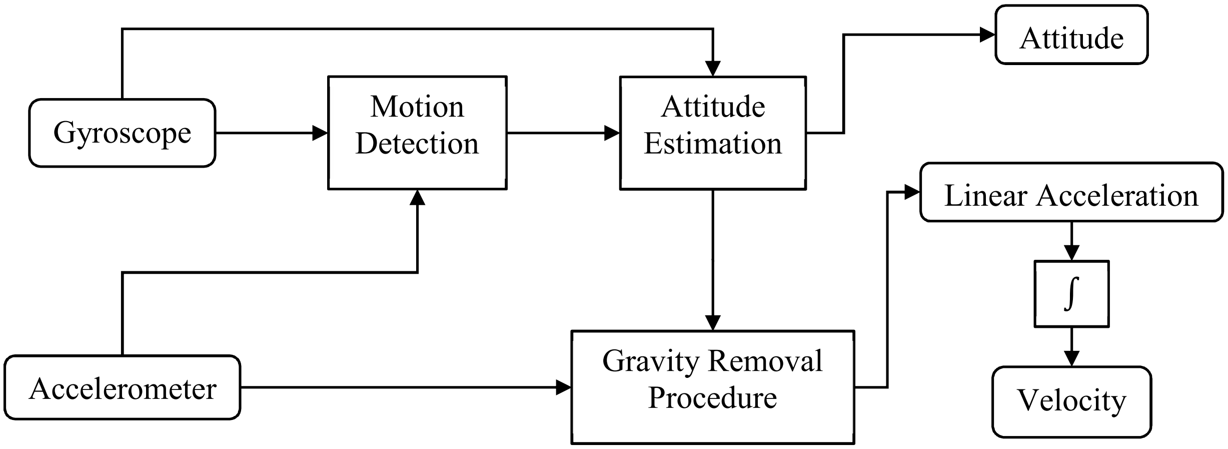


Figure 4-9 - Block Diagram of Linear Acceleration Estimation Algorithm

4.3.2 Reference Generation

To evaluate the performance of a motion based test, a reference signal was generated using an external monitor. The reference motion signal is constructed using a GoPro camera to determine the displacement of the system while it goes though a series of motions on the 3-D printer. The position of the system is tracked though the use of two circular markers which is then tracked through Hough circle detection used on the video clip. This is later used to determine the velocity and generate the reference that different devices are compared to.

4.3.3 Testing Method

There are two main testing runs done to compare the abilities of the devices in Dead Reckoning applications. Both tests operate by introducing a certain amount of attitude change along with motion.

4.3.3.1 *Single Attitude Change Test Dynamics*

This test repeats the same attitude change and movement back and forth 10 times. The final position angle is empirically determined from the final position. Due to the nature of the setup, the exact attitude change during the movement is unknown.

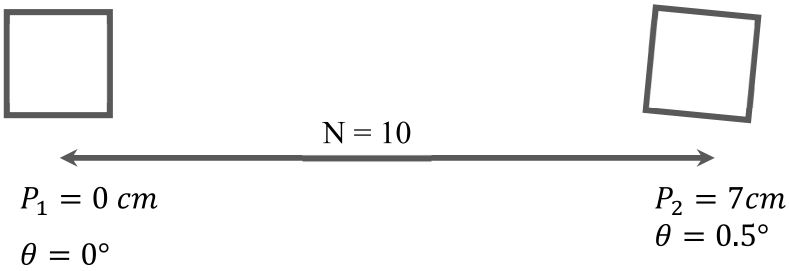


Figure 4-10 - Test dynamics of the single attitude change test

4.3.3.2 *Small-Step Attitude Change Test Dynamics*

This test does small steps which introduces a small attitude change over 7 steps, goes back to the initial position and repeats this in the other direction.

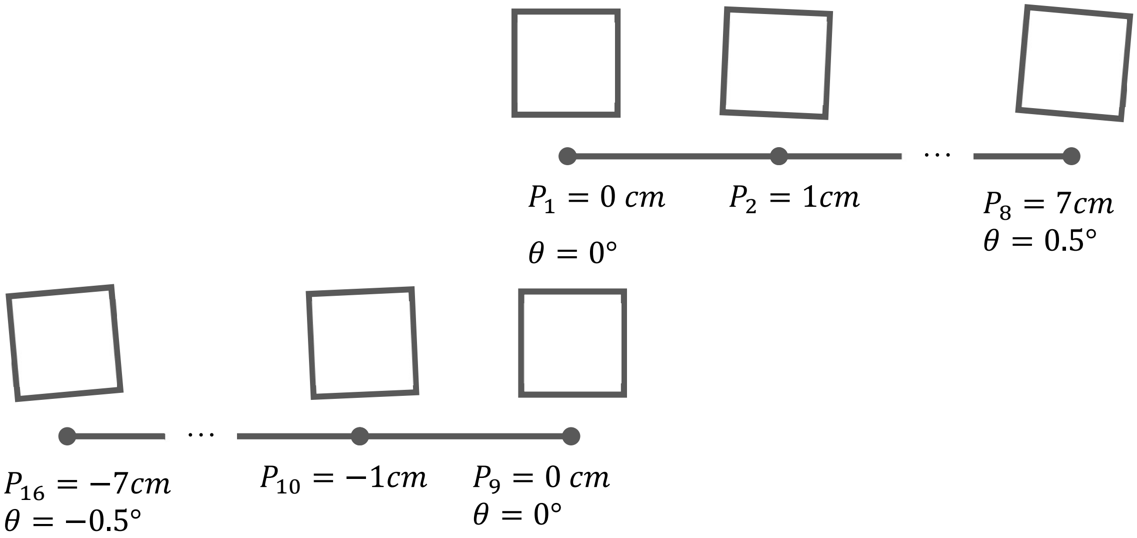


Figure 4-11 - Test dynamics of the small-step attitude change test.

4.3.4 Dead Reckoning Testing Results

For ease of visual comparison, velocity is used to compare between the reference and the device data. The single attitude change was applied across the different devices, and figure 4-12 and 4-13 are the resulting graphs of the velocity profile. The RMS of the difference between the reference and each devices' velocity estimation was also computed for a general comparison of their performance.

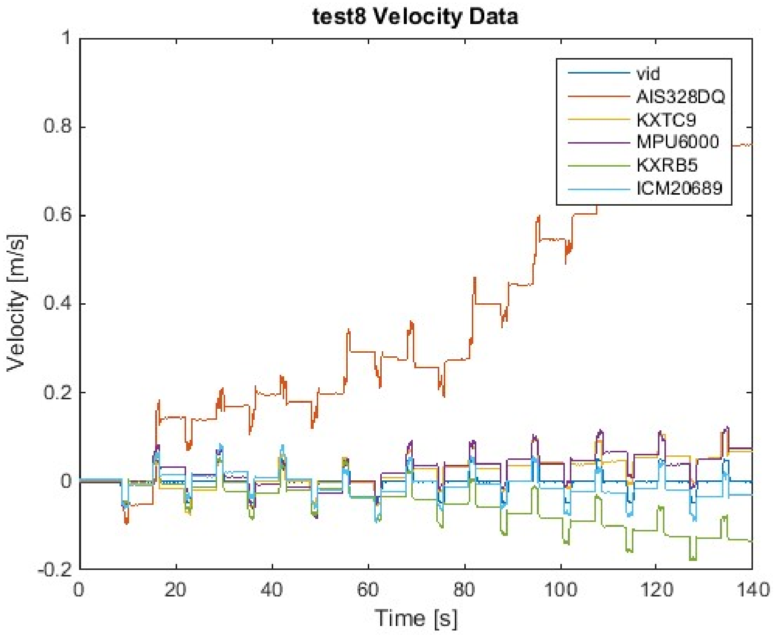


Figure 4-12 - Velocity profile of the single attitude change test with the Complementary filter applied

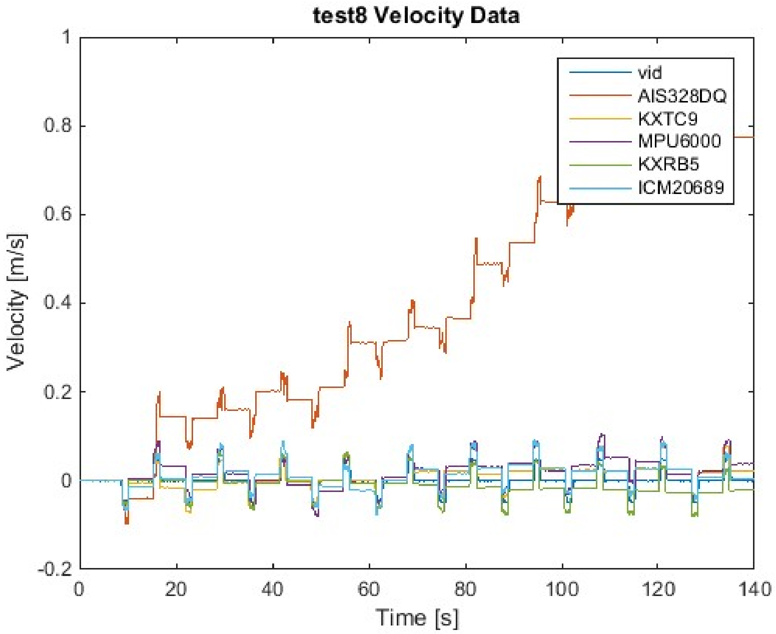


Figure 4-13 - Velocity profile of the single attitude change test with the Kalman filter applied

There are two main observations from looking at the graphical results. The Kalman filter has an observable performance improvement compared to the complementary filter – comparing figure 4-12 to 4-13. Secondly, AIS328DQ's poor performance in attitude estimation is translated to the results in the dead-reckoning tests. This was also shown in the step-motion specification, where even in the absence of any attitude change, the AIS328DQ performed poorly due to errors during movement which gets magnified by integration to velocity.

The following table shows the RMS velocity errors, allowing for a per device comparison.

Table 4-3 - Results of the single attitude change dead reckoning test

	Complementary Filter $[\frac{m}{s}]$	**Kalman Filter $[\frac{m}{s}]$**
AIS328DQ	0.278	0.296
MPU6000	0.027	0.017
KXTC9	0.027	0.013
ICM20689	0.045	0.020
KXRB5	0.018	0.017

The same process was applied to the small step attitude change test, and the results are shown in Figure 4-14 and 4-15, and the RMS errors are summarized in table 4-4.

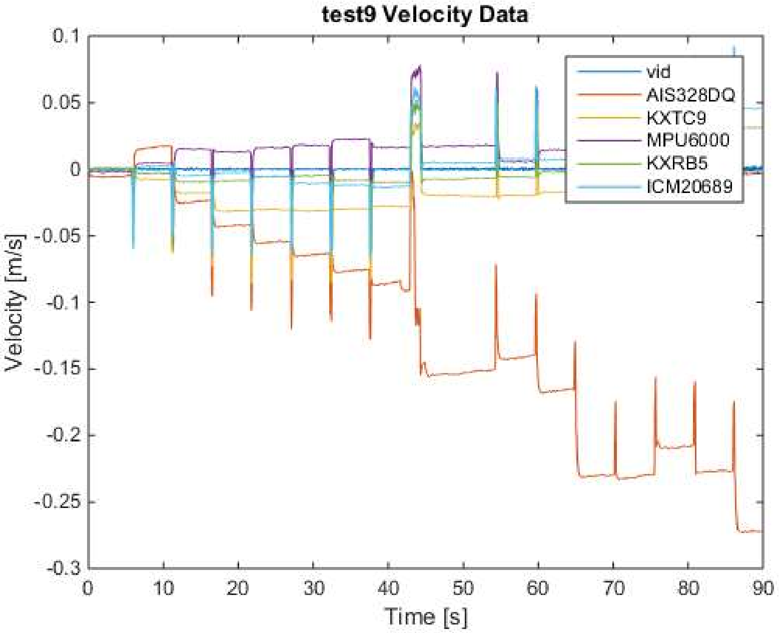


Figure 4-14 - Velocity profile of the small step attitude change test with the Complementary filter applied

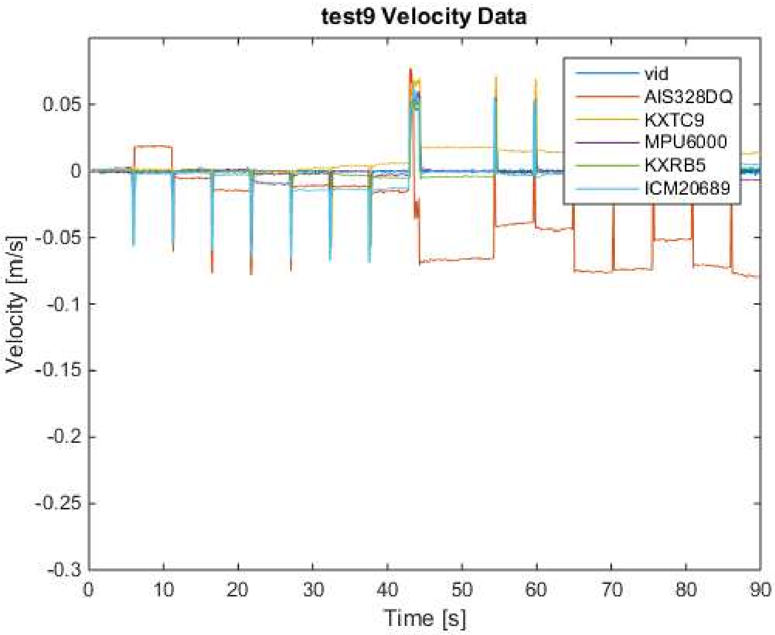


Figure 4-15- Velocity profile of the small step attitude change test with the Kalman filter applied

Table 4-4 - Results of the small step attitude change dead reckoning test

	Complementary Filter [$\frac{m}{s}$]	**Kalman Filter [$\frac{m}{s}$]**
AIS328DQ	0.0859	0.0324
MPU6000	0.0113	0.0092
KXTC9	0.0110	0.0063
ICM20689	0.0174	0.0101
KXRB5	0.0133	0.0049

The importance of the inertial algorithms are shown in this test. In the simulation with 0.5° error over a duration of 60 seconds, over 5 $\frac{m}{s}$ of velocity error resulted. The first test with a fixed angle applied consistently over 140s showed a significant reduction, showing significant improvement over calibration techniques alone. In addition, the two inertial algorithms show a consistent result across the different tests, with the Kalman filter performing better than the complementary filter. The Kalman filter is especially better with longer movement and higher amounts of vibrations, as shown in the second attitude estimation test and in the first dead-reckoning test which both introduce a larger amount of motion and movement. The better performance of the Kalman filter is considerable in dead-reckoning scenarios, but considering the larger amounts of tuning needed, it is much simpler to maintain and tune the complementary filter which is based off a single parameter.

Another purpose of the dead-reckoning based movement test is to see if there are different specification considerations when looking at different applications. To do this, device specific performance was considered across the two dead-reckoning based tests.

Looking at the error deviation, it reconfirms the observations from the graphs, where AIS328DQ has a much higher error. Similar to the attitude estimation tests, there is no relationship between the noise performance and the inertial test performances. This interesting notion is shown in AIS328DQ and MPU6000, which have very similar noise density performance, but the two very different inertial test performance. On the contrary, looking at the step motion and bias variation specifications should provide a better performance indicator as both these tests capture the effect of errors that get exacerbated by integration during motion and static periods. In this case, MPU6000 has a much better performance in the step motion test, while having a similar noise density to AIS328DQ. Another device which highlight this is ICM20689, which have poorer performance in step motion and bias variability compared to MPU6000 but has a much better noise performance. This shows that noise density does not impact performance as much as cross axis and bias variability when inertial algorithms, such as the Kalman and complementary filter, are applied. Cross axis specification was important in attitude estimation, however, in dead-reckoning purposes, where acceleration is integrated, the impact of the bias variations and step motion scores are much more significant. This is seen with the cross-axis performance of the ICM20689, which is better than the MPU6000 and KXTC9. However, the ICM20689 performs worse than both those devices in dead-reckoning tests.

Both tests introduced attitude changes in the movement, but the second test broke the motion into shorter durations, which decrease the timescale of the movement and reduces the influence of using a gyroscope. This subsequently reduces the amount of error which is seen across the two tests, where the smaller changes caused a much smaller error.

Chapter 5 Conclusions and Future Works

5.1 Conclusions

The use of inertial sensors and accelerometers is widespread for military and transportation purposes and is motivation for continued research into better the understanding of how to improve their performance for their navigational purposes. Despite the increased use of MEMS accelerometers in these fields, their testing methodology isn't clearly defined, and often not specified between manufacturers. This difference makes spec sheets hard to be utilized to compare between devices and make judgement in the utilization of these devices. Although it is possible to test these devices, it is often costly and measure specific specifications which may not impact the specific application.

To address the inconsistencies, a scalable IMU testing board was created to support a variety of communication protocols. This system allows new MEMS devices to be implemented on a daughter board and plugged into the testing board. This removes any inconsistencies resulting from the different testing platforms and gives a consistent comparison base between different devices. To address the inconsistent testing methodologies, the dynamics and movement of the accelerometer were controlled with a retrofitted Rostock 3D printer. The printer served as a low-cost movement platform for the IMU testing platform, and the flexibility allowed for movement based specifications which highlight different aspects which are not testable using traditional shaker or rate tables. Utilizing this system, five specifications were evaluated across seven different DUTs. Through this evaluation process, it was shown that several of the devices do not conform to their specifications, with some devices having an incorrect reliance on output data rate. Also, we show that two different methods of calculating a similar specification can yield vastly different results, and by not clearly indication the methodology it can hinder the comparison of devices. In addition, we show the comparative performance of the different MEMS accelerometers. Finally, a proposed scoring system and weighting system was introduced to provide an example of how these testing results can be utilized to differentiate between devices Specifically, the MPU6000, KXTC9 and KXRB5 are 3 devices which were identified to be better accelerometers when looking at specifications in a dead-reckoning scenario.

The accuracy and consistency of these testing methodologies are crucial to comparing different MEMS accelerometers. However, in practice, MEMS accelerometers are mostly used in conjunction with other devices due to their susceptibility to vibrations and external effects. This leads to a disconnect between the performance evaluated from looking at specifications alone and the actual performance when utilized There are multiple studies which explore the application of inertial algorithms to improve the inertial navigation system's accuracy in determining the attitude and location of a system. However, most of these algorithms use other sensors, such as gyroscope, magnetometers, Wi-Fi or even GPS in conjunction with the accelerometer. To explore this, the complementary and Kalman filter were utilized to combine each accelerometer with a gyroscope, MPU6000. Magnetometers, Wi-Fi and GPS were not included in the algorithm due to the limitation of the daughter boards created. These accelerometer and gyroscope pairs were tested by looking at specific applications that inertial sensors are commonly used in – attitude estimation and dead reckoning. Tests involving attitude estimation and velocity estimation were used to benchmark the results of the different accelerometers.

This section showed that inertial algorithms are vital to accurately determine velocity and attitude. The Kalman and complementary filters are real-time filters that allow a much more accurate velocity and thus displacement to be calculated. The Kalman Filter consistently outperformed the complementary filter in tests which includes vibrations or movements. From the two application based tests, the impact of accelerometer specification on the inertial test results were evaluated. Noise density, which is one of the most common methods used in comparing performance of accelerometers, is shown to be a poor indicator of performance when inertial algorithms are applied. Bias variability and step motion, were better indicators of performance for dead reckoning purposes when inertial algorithm were applied. On the other hand, cross-axis is better for attitude estimation where motions are limited.

To summarize, this study achieved the following:

- An inertial sensor measurement platform was designed and implemented to provide a low-cost and consistent testing system for inertial sensors.
- A strict methodology was outlined to test five different specifications. These specifications exercise different aspects of a MEMS accelerometer, giving a more well-rounded comparison of accelerometers than just noise density.
- Real time inertial algorithms, real time Kalman and complementary filters, were shown to be necessary and much better than simply using a better accelerometer in most cases.
- A common specification of accelerometers, noise density, is a poor indicator of performance after inertial algorithms are applied. Specifications such as step motion, cross-axis and bias variabilities were more important depending on the specific application the system is used in.

5.2 Future Work

In this study, common specifications and testing methodologies were explored. To address this a testing platform and system was built and used to ensure consistency. However, this platform still has limitations that hinder the testing methodologies, such as the small testing platform that limited the range of movement. This subsequently limits the accelerations and the way the devices are moved. Secondly, analog inertial sensor inputs into the processor are limited to ensure a good accuracy and speed can be achieved. This limited INS implementation when more sensors were involved, thus making it difficult incorporate a magnetometer for some of the boards. By improving the testing platform, it will allow for a more realistic testing platform that will be closer to how these accelerometers are implemented in practice.

The first part of the study evaluated five specifications and testing methodologies associated with them. In this case, there were two other specification that are important to consider – temperature and vibrations. However, the platform was limited and unable to test these in a repeatable manner due to the inability to control the temperature precisely. These specifications are important for navigation purposes and should be evaluated in a future study. Another aspect which can be explored is comparing the current test results to existing testing equipment. The available testing platforms are very bulky and costly and weren’t an available means to compare with the 3D printer platform.

Finally, in regards to the applications and inertial algorithms test, one of the limitations was the exclusion of the magnetometer. As a magnetometer has a similar role as the accelerometer and gives a more stable attitude calculation, the inclusion of a magnetometer can have a big impact upon whether the accelerometer choice affects applications. From evaluating just an accelerometer and gyroscope pair, the choice of accelerometer is important, but there are other more significant factors such as utilization of inertial algorithms.

To summarize, there are several suggestions for future considerations in this study.

- Utilize a movement platform with a bigger range of movement
- Utilize a new processor to improve speed and accuracy limitations with analog devices.
- Develop a platform where temperature and frequency of motion can be controlled in a precise manner so temperature and vibration performance can be evaluated.
- Utilize a shaker table and rate table to compare with the results from the 3D printer based platform
- Investigate the impact of accelerometers with magnetometers in the INS

Appendix A – Testing Board Implementation

This appendix will cover the construction of the IMU testing platform and the specifications of the system. The first section will discuss in detail the hardware created while the second section will cover the software drivers implemented.

A.1 Motherboard

This section highlights the motherboard details – including the block diagram, PCB and specifications. The main processor chosen was a Cortex M4 from ST Microchip – STM32F302R8. The chip supports a variety of protocols needed for communication with the sensors and storage solutions. The following figure shows processor's functional blocks and how they are connected to data storage (Computer/SD Card), sensors, and Board IO. Details regarding major blocks will be discussed in the subsequent sections.

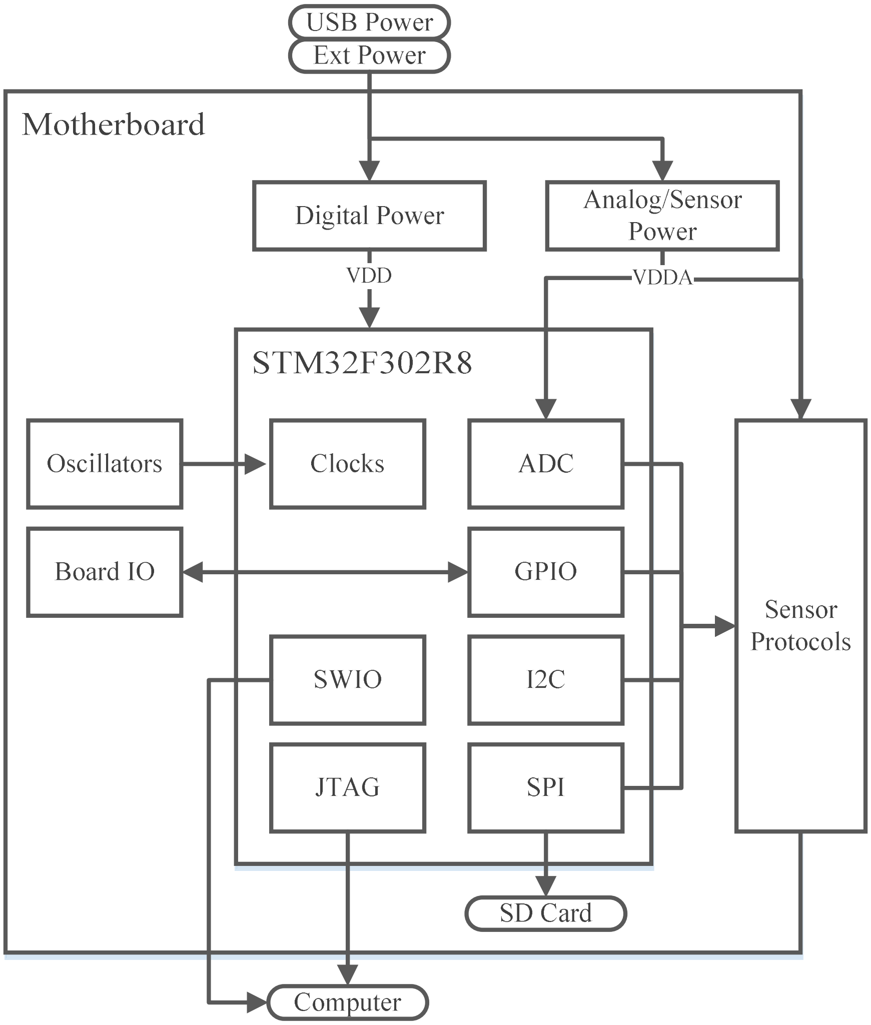


Figure A-0-1 This figure highlights the functional blocks within the processor and their use for different aspects of the data logging system.

A.1.1 Power Domain

The IMU platform can be supplied power through two different means – micro-USB input and external DC input. External DC inputs supports voltages ranging from 6V to 16V. The power inputs are used to drive two different power domains through the use of two 3.3V LDO regulators:

- Analog/Sensor Board Power (VDDA)
- Processor and Motherboard Power (VDD)

The two power domains are separated to isolate supply noise from the high speed digital processor and the sensors/analog components. The two domains share a common ground on the board to ensure continuity.

The board also has a “reset” button that will reset the processor, however, it will not bring down the 3.3V line for the sensors or analog blocks. Sensor reset is done through software during the software initialization stage.

A.1.2 Oscillators and Clocking

There are two components which require clocking – Sensor Board and Processor. For the processor, the logger has two crystal oscillators – 8 MHz and 32.768 kHz. The STM32F302R8 processor has two options for clocking – an internal RC resonator or an external oscillator supplied to an on-processor resonator circuit. For typical applications, the internal solution is enough, however, an external 8 Mhz crystal oscillator is used to improve speed and reduce clocking errors for clock signals generated by the processor PLLs. The clocks supplied to the sensor board is generated by processor timer.

Using the external oscillators, the Data Logger core runs at 72 MHz. The platform is capable of generating a clock up to 36 MHz for the sensor board.

A.1.3 Board Input Output Methods

The following input/output methods are available on the motherboard:

- SWIO/JTAG
- USART
- SD Card
- Micro-USB
- GPIO

SWIO/JTAG are the debugging and programming protocols supported on ST processors. It is mainly used for development purposes, but can be used for communication with the processor when needed.

USART is implemented for data collection purposes – allowing the motherboard to interface with computerized scripts or serial port loggers. However, due to the speed limitation of the protocol, it limits the data collection rate of the platform when using this protocol to a single 3-axis IMU at less than 500 Hz sampling rate.

SD cards are supported using the SPI protocol and is mainly used for data collection purposes. To support this, an open-source FatFs filesystem with an SPI interface was implemented. This interface enables higher speed data collections – supporting up to 2 devices (3-Axis) up to a sample rate of 1 kHz.

Micro-USB is currently used to power the whole system. However, the USB data lines are connected as well, allowing a future firmware/driver upgrade to enable a tethered high speed data collection method.

GPIOs are available for user configuration. There are a total of 4 user-programmable IOs - 2 active high push buttons and 2 LEDs. In addition to the user programmable IOs, there is a power indicator LED and a reset button for processor reset which are not configurable.

A.2 Sensor board

This section will cover the sensor board details, including details on the communication protocols available, example PCBs and pinouts for reference sensor boards.

A.2.1 Sensor Communication

To support a variety of IMUs, the motherboard supports the following protocols for interfacing with a variety of sensors:

- SPI communication
- I2C communication
- Analog input

SPI protocol implemented on the data logger platform supports SPI clock speeds of up to 18Mhz, making it sufficient for most commercially available IMUs on the market.

I2C protocol implemented on the data logger platform supports bit rates of up to 400kBits/s.

Analog input is supported by the use of the Analog Digital Converter (ADC) in the motherboard. It supports 12-bit conversions at speeds up to 5 Msps.

A.2.2 Example Analog PCB and Pinout Utilization

The next example shows a board which utilizes the analog inputs as well as the digital communication protocols. The board was designed such that it can be populated with several different combination of analog and SPI devices, only requiring a different population of the board.

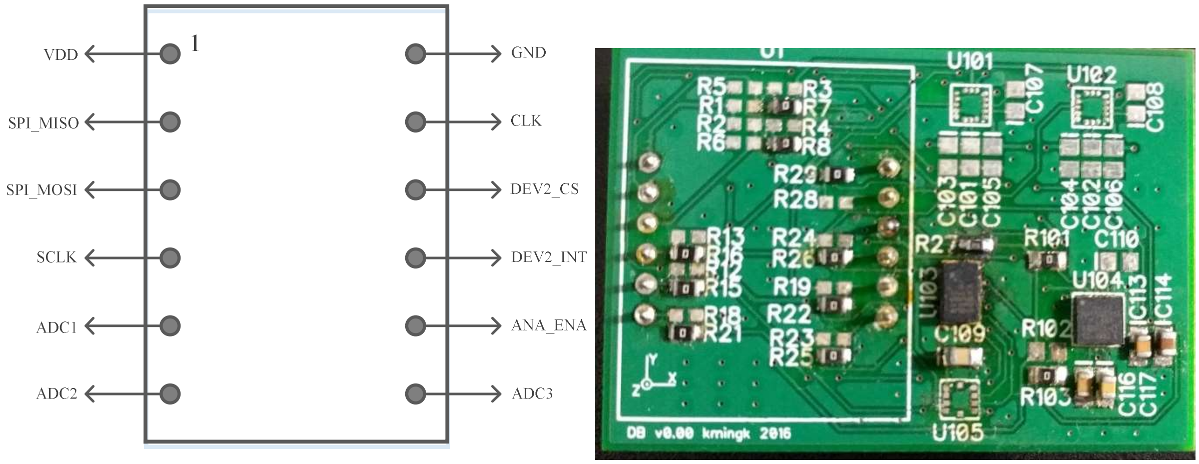
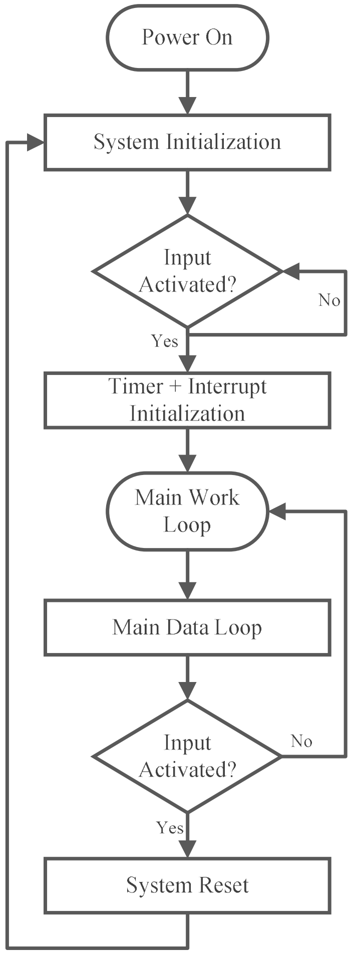


Figure A-0-2 Example of pinout for SPI/Analog devices (Left) Picture of implemented PCB (right)

Appendix B – Testing Board Software Flow

B.1 Data Logger Execution Flow



The software API previously outlined lets the user tailor the program for their specific application. It provides the tools needed to log the data or process it for the specific metrics needed. This section will cover the specific program that was implemented for our testing platform, where the main purpose is to log the data in an accurate and consistent manner.

The program for the testing is broken down into three main stages.

- System Initialization stage
- Data capture stage
- System Reset Stage

The flow chart to the left highlights the main process and decision making in the execution flow of the data logging software.

B.1.1 Initialization Stage

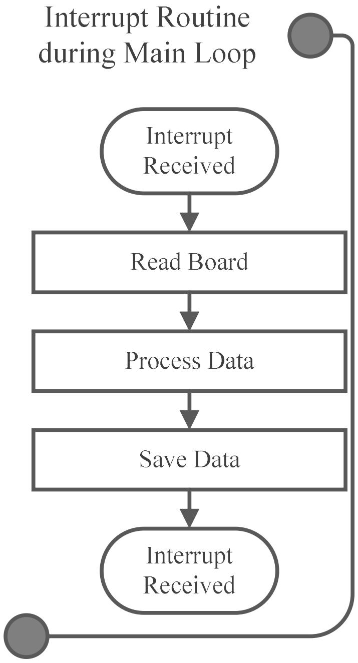
This stage is meant to initialize the components necessary to sample and read from the sensor boards. It will perform the following routines:

1. Board Configuration – Pin configuration and the sensors on the board are set.
2. Device Initialization – Initialization for each device on the board.

3. Data Capture Initialization – Initialize the sampling frequency and interrupt routine.
4. Data Buffer/Storage Solution Initialization – Prepare the buffer and the storage solution.

At the end of this stage, it will wait for a user input before continuing on the capture process. This also alleviates the varied delays which may occur from the SD card initialization which may cause synchronization issues.

B.1.2 Data Capture Stage



There are two sections in the data capture stage – main loop and interrupt routine. In the main work loop, the data logger continuously counts the time elapsed and monitors the data buffer for any dirty data. When dirty data exists, it is written over the data storage API, and the flag is removed.

The interrupt routine is triggered by the internal timer at the specified sampling rate. The devices are then read sequentially using the board level API. Each value is saved in the circular data buffer and flagged appropriately. The buffer is necessary to ensure timely processing of the data every time the interrupt is triggered, and to enable the data to be saved in time at the main loop.

B.1.3 Reset Stage

This stage is meant to reset the components - it will perform the following routines:

1. Board/Device level reset – This will trigger the reset conditions for the sensors
2. Data Storage reset – This will flush the data and ensure all the data are written and saved.

References

[1] V. Kaajakari, Practical MEMS, Las Vegas: Small Gear Publishing, 2009.

[2] E. T. Benser, "Trends in inertial sensors and applications," in *2015 IEEE International Symposium on Inertial Sensors and Systems (ISISS)*, Minnesota, 2015.

[3] S. M. Lavalle, A. Yershova, M. Katsev and M. Antonov, "Head Tracking for the Oculus Rift," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[4] W. Kang, S. Nam, Y. Han and S. Lee, "Improved heading estimation for smartphone-based indoor positioning systems," *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC),* pp. 2449-2453, 2012.

[5] National Institute of Standards and Technology, "Precision Testing for MEMS Accelerometers," 8 April 2016. [Online]. Available: https://www.nist.gov/news-events/news/2016/04/precision-testing-mems-accelerometers.

[6] D. A. J. Rene Rail-Ip, *Evaluation of Low-Cost MEMS Accelerometers and Investigation of Inertial Algorithms for Dead Reckoning in Railway Environment,* Toronto, Ontario: University of Toronto, 2014.

[7] L. Sahawneh and M. Jarrah, "Development and calibration of low cost MEMS IMU for UAV applications," in *5th International Symposium on Mechatronics and Its Applications, 2008.* , 2008.

[8] F. Edalatfar, B. Yaghootkar, A. Q. A. Qureshi, S. Azimi and B. Bahreyni, "Design, fabrication and characterization of a high performance MEMS accelerometer," 2016.

[9] M. A. Y. M. E.-D. Moustafa Youssef, "GAC: Energy-Efficient Hybrid GPS-Accelerometer-Compass GSM Localization," 2010.

[10] A. Noureldin, T. B. Karamat, M. D. Eberts and A. El-Shafie, "Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost Navigation Applications," *IEEE Transactions on Vehicular Technology,* vol. 58, no. 3, pp. 1077-1096, 2009.

[11] D. Gebre-Egziabher, R. C. Hayward and J. D. Powell, "Design of multi-sensor attitude determination systems," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 40, no. 2, pp. 627-649, 2004.

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

[13] R. Zhi, "A Drift Eliminated Attitude & Position Estimation Algorithm In 3D," *University of Vermont,* 2016.

[14] P.-C. Wu, C.-Y. Yeh, H.-H. Tsai and Y.-Z. Juang, "Low-frequency noise reduction technique for accelerometer readout circuit," *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS),* pp. 25-28, 2016.

[15] F. A. K. N. NAVID YAZDI, "Micromachined Inertial Sensors," 1998.

[16] C. W. L. Y. S. O. Y. H. C. K. Y. Park, "Laterally oscillated and force-balanced micro vibratory rate gyroscope supported by fish hook shape springs,," in *IEEE MicroElectro Mechanical Systems Workshop*, Japan, 1997.

[17] R. Mahony, T. Hamel and J.-M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *IEEE Transactions on Automatic Control,* vol. 53, no. 5, pp. 1203-1218, 2008.

[18] Petteri Aimonen, "Wikipedia," 25 November 2011. [Online]. Available: https://en.wikipedia.org/wiki/Kalman_filter#/media/File:Basic_concept_of_Kalman_filtering.svg.

[19] W.J. Riley, *Handbook of Frequency Stability Analysis,* Boulder, Colorado: NIST Special Publication 1065, 2008.

[20] G. C. T. Richard Schreier, Understanding Delta-Sigma Data Converters, New Jersey: Wiley Interscience, 2005.