Fundamentals of Motion Sensing and Signal Processing

Farros Alferro [1] **Shunsuke Yamada** [2]

1. Introduction

MEMS, or micro-electro-mechanical systems, is a technology that, in its broadest sense, can be characterized as miniature mechanical and electro-mechanical components (i.e., devices and structures) created utilizing microfabrication methods [1]. MEMS inertia sensors, such as gyroscopes and accelerometers, have recently gotten smaller and less expensive. As a result, the sensor is frequently found in mobile devices such as smartphones, tablets, mobile PCs, HDDs, automotive navigation systems, and gaming controllers. The accelerometer measures the inertia force applied to the gadget, while the gyroscope detects one of the inertia forces, Coriolis' force.

In this experiment, we will study about digital communication between devices via a microprocessing unit (MPU) and a MEMS accelerometer. The MEMS accelerometer and the gyroscope are then used to teach us about the inertia measurement unit (IMU). Summarizing, we learn about the use of MEMS sensors and its applications from these experiments.

2. Principles

2.1. MEMS Accelerometer

The accelerometer works on the premise of a tiny mass being held in place by a tiny spring. The mass was shifted by the inertia force as the gadget was accelerated. The mass displacement is inversely proportional to the applied acceleration if the spring is linear. The displacement is often measured by a piezo-resistor or capacitor [2].

2.2. MEMS Gyroscope

The angular rate is determined using a gyroscope. Gyroscopes of many different varieties, including mechanical, gas, and optical ones, are being proposed. Rotational and oscillational gyroscopes are the two categories of mechanical gyroscopes. The oscillational gyroscope, which is a common type of commercial MEMS gyroscope, operates on a principle that is depicted in Fig. 1. Without rotation, the oscillation mode of the instrument is depicted in Fig. 1(a). To reduce energy loss through the fixed component, each beam moves in the opposite direction, raising the oscillation Q-factor. The definitions of v and m, respectively, refer to the effective velocity and mass of the beam. When the device is rotated with the angular rate of ω, the generated Coriolis force, F_c, is determined as follow:

$$\overrightarrow{F_c} = -2m\vec{\omega} \times \vec{v} \tag{1}$$

The vibration's direction changes from up to down as seen in Fig. 1(b) because the Coriolis force's direction is orthogonal to the direction of the vibration. The angular rate is directly proportional to this out-of-plane displacement. Each beam's displacement direction is opposing, which is easily differentiated by the in-phase displacements caused by the acceleration [2].

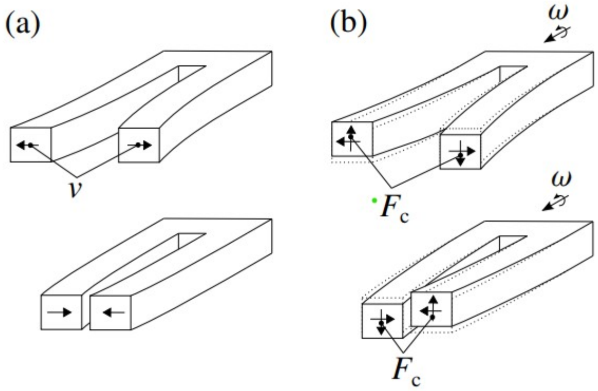


Figure 1. Principle of the gyroscope. Vibration mode (a) without rotation and (b) with rotation.

2.3. Allan Variance

Allan variance is defined as one-half of the time average of the squares of the differences between successive frequency deviation values taken across the sampling period. Because the Allan variance is affected by the time interval between

[1] Computer Vision Lab., Tohoku University, Sendai, Japan [2] Smart System Integration Lab., Tohoku University, Sendai, Japan. Correspondence to: Farros Alferro <farros.alferro.t3@dc.tohoku.ac.jp>, Shunsuke Yamada <santa@tohoku.ac.jp>.

Theme 1 Report on *Laboratory Experiment II* which was done on November 4th, 2022.

samples, it is often expressed as τ; similarly, the measured distribution is displayed as a graph rather than a single number. A clock with a low Allan variance has strong stability across the measured period. Allan deviation is commonly utilized in plots (conveniently in log-log format) and number presentation. It is preferred because it provides relative amplitude stability, which allows for easy comparison with other sources of inaccuracy. An Allan deviation of $1.3 \cdot 10^9$ at observation time (i.e., = 1 s) should be understood as a frequency instability between two observations separated by one second with a relative root mean square (RMS) value of $1.3 \cdot 10^9$. This is equivalent to 13 MHz RMS movement for a 10 MHz clock. If an oscillator's phase stability is required, the time deviation variants should be consulted and used [3].

2.3.1. *M*-Sample Variance

The *M*-sample variance is defined [4] as:

$$\begin{aligned}\sigma_y^2(M,T,\tau) = \frac{1}{M-1}\Bigg\{\sum_{i=0}^{M-1}\left[\frac{x(iT+\tau)-x(iT)}{\tau}\right]^2 \\ - \frac{1}{M}\left[\sum_{i=0}^{M-1}\frac{x(iT+\tau)-x(iT)}{\tau}\right]^2\Bigg\}\end{aligned} \tag{2}$$

or with average fractional frequency time series

$$\sigma_y^2(M,T,\tau) = \frac{1}{M-1}\left\{\sum_{i=0}^{M-1}\bar{y_i}^2 - \frac{1}{M}\left[\sum_{i=0}^{M-1}\bar{y_i}\right]^2\right\} \tag{3}$$

where M is the number of frequency samples used in variance, T is the time between each frequency sample and τ is the time-length of each frequency estimate. An important aspect is that M-sample variance model counter dead-time by letting the time T be different from that of τ.

2.3.2. Allan Variance

The Allan variance is defined as:

$$\sigma_y^2(\tau) = \langle\sigma_y^2(2,\tau,\tau)\rangle \tag{4}$$

which is conveniently expressed as

$$\begin{aligned}\sigma_y^2(\tau) &= \frac{1}{2}\langle(\bar{y_{n+1}} - \bar{y_n})^2\rangle \\ &= \frac{1}{2\tau^2}\langle(x_{n+2} - 2x_{n+1} + x_n)^2\rangle\end{aligned} \tag{5}$$

where τ is the observation period, $\bar{y_n}$ is the *n*th fractional frequency average over the observation time τ. The samples are taken with no dead-time between them, which is achieved by letting $T = \tau$

2.3.3. Allan Deviation

Just as with standard deviation and variance, the Allan deviation is defined as the square root of the Allan variance.

$$\sigma_y(\tau) = \sqrt{\sigma_y^2(\tau)} \tag{6}$$

2.4. Inter Device Communication

There are two types of data transfer methods: analog and digital. The analog approach generates a voltage proportionate to the sensing data (acceleration or angular rate). A standard measurement tool, such as a voltmeter, can easily measure the output signal (voltage). To utilize the device in a digital system, however, an analog-to-digital converter is necessary. Furthermore, the digital technique is noise resistant, as lower external noise means a lower signal-to-noise ratio. Furthermore, electric wires can be shared by multiple devices. As a result, many digital communication protocols have been proposed, such as SPI, I2C, and UART [2].

2.4.1. Serial Peripheral Interface (SPI)

SPI, or Serial Peripheral Interface, is a serial communication protocol that is commonly used in embedded systems to transport high-speed data between devices on the bus. It uses a master-slave paradigm with at least four signals: a clock (SCLK), a master output/slave input (MOSI), a master input/slave output (MISO), and a slave select (SS). On the bus, all devices share the SCLK, MOSI, and MISO signals. The master device generates the SCLK signal for synchronization, while the MOSI and MISO lines are employed for data interchange. Furthermore, each slave device connected to the bus has its own SS line. To pick a device for communication, the master pulls low on a slave's SS line.

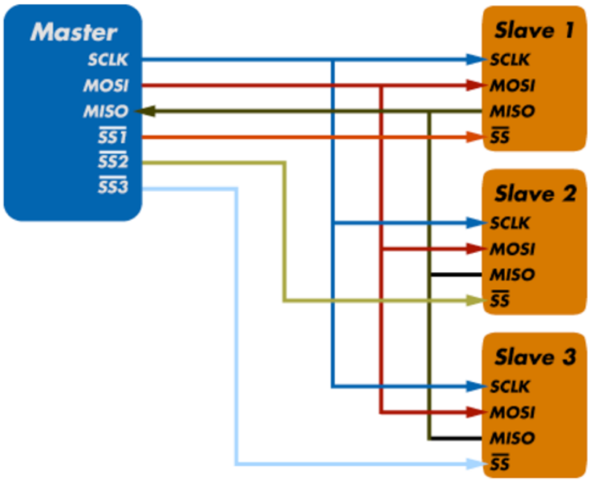


Figure 2. An example of SPI bus configuration [5].

SPI communication provides full-duplex communication, which means that both the master and slave can send data at the same time. Because there is no predefined protocol for the exchange, SPI is appropriate for data-streaming applications. It also does not have a maximum speed; data rates of more than 100 MHz have been obtained [5].

2.4.2. Inter-Integrated Circuit (I^{2}C)

Inter-Integrated Circuit, or I^{2}C, is a simple communication protocol used in embedded systems to send data between a master (or many masters) and a single slave (or multiple slaves) device. It is a bidirectional two-wire serial bus that sends and manages data bit by bit between devices attached to the bus using serial clock (SCL) and serial data (SDA) lines. The master regulates data sharing between devices in I^{2}C activities. In order to transfer data or request a response, a master device will send a signal to a slave. To accomplish this, each slave device must provide a unique address in the I^{2}C message.

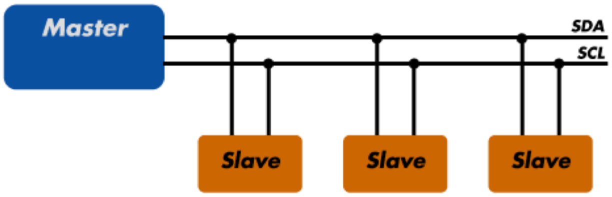


Figure 3. An example of I^{2}C bus configuration [5].

Each I^{2}C message sent over the bus includes an address frame for the slave device and one or more data frames containing the transferred data. The message also contains start and stop conditions, read/write bits from either the master or slave, and ACK/NACK bits transmitted from the recipient to check for errors. I^{2}C is considered synchronous, which means it uses a serial clock. The master device drives the clock, allowing the output of bits to be synchronized to the sampling of bits by the clock signal shared by the master and the slave. The I^{2}C protocol has a normal data transfer rate of *100 kbps*, although data speeds of up to *5 Mbps* are feasible with I^{2}C devices configured in ”fast mode” or ”ultra-fast mode.” [5].

2.4.3. Universal Asynchronous Receiver/ Transmitter (UART)

The Universal Asynchronous Receiver/Transmitter (UART) is a physical microcontroller or single integrated circuit (IC) that is used to implement serial communication between devices in an embedded system. A UART's primary function is to transmit and receive serial data. Two UARTs communicate directly with each other in UART communication; the UART on the sending device, or the transmitting UART, receives parallel data from the CPU (microprocessor or microcontroller) and converts it to serial data. This serial data is sent to the receiver device's UART or the receiving UART. The receiving UART translates the serial data received back to parallel data and feeds it to the CPU. Shift registers on the transmitting and receiving UARTs are used to convert serial-to-parallel and parallel-to-serial data. Only two wires are required for communication in UART communication: data travels from the Tx pin of the transmitting UART (Transmitter Tx) to the Rx pin of the receiving UART (Receiver Rx).

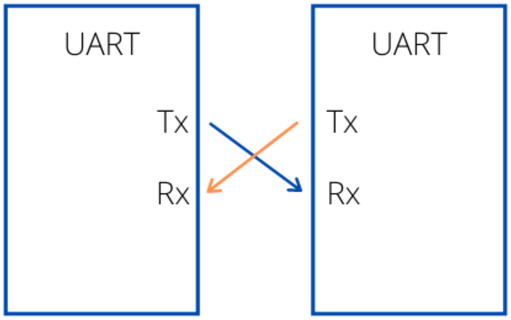


Figure 4. An example of UART bus configuration [5].

UART data is transmitted in a packet over the bus. A packet is made up of start bits, data frames, parity bits, and stop bits. To assist maintain data integrity, the parity bit is utilized as an error checking method. Because the settings, such as transfer speed and data speed, are customizable by the developer, UART is termed ”universal.” Bidirectional data transmission, including half-duplex and full-duplex operations, is supported by UART. It's also asynchronous, which means it doesn't need a clock signal to synchronize the transmitting UART's output bits with the receiving UART's sampling bits. In the absence of a clock, the receiving and sending UARTs must operate at the same baud rate or bit rate. This allows the system to determine where and when the bits were clocked [5].

2.5. Inertia Measurement

Inertia measurement is a technique for determining position and orientation by employing inertia sensors such as an accelerometer and a gyroscope. The accelerometer and gyroscope monitor 3-axes acceleration and angular rate, respectively. Integrating these accelerations and angular rates yields the position and orientation. Figure 5 depicts the definition of space-fixed coordinates *(X-Y-Z)* and the measurement unit *(x-y-z)*. Using *X-Y-Z* coordinates, the origin of the *x-y-z* is represented as $\boldsymbol{R_0} = (X_0, Y_0, Z_0)^T$. A quatanion, which consists of four components, is used to symbolize the rotation from *X-Y-Z* to *x-y-z* [2].

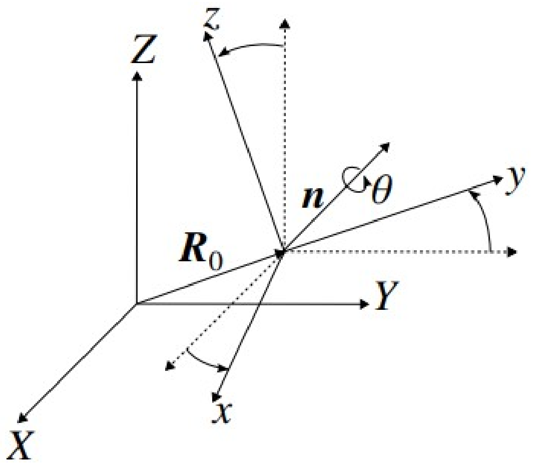


Figure 5. Space fixed coordinate *(X-Y-Z)* and IMU coordinate *(x-y-z)* [2]

3. Experiment

Figure 6 depicts an IMU diagram. An MPU (STM32F405, STMicroelectronics), a 9-axis MEMS combo-sensor (MPU-9250, Invensense Inc., 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer), a GPS receiver, and an SD-card connector comprise the IMU. Two independent SPI buses connect the MEMS sensor and SD card to the MPU. UART connects the GPS receiver to the MPU (Universal Asynchronous Receiver Transmitter). The master device is the MPU. The measured data is sent to the PC via the USB bus [2].

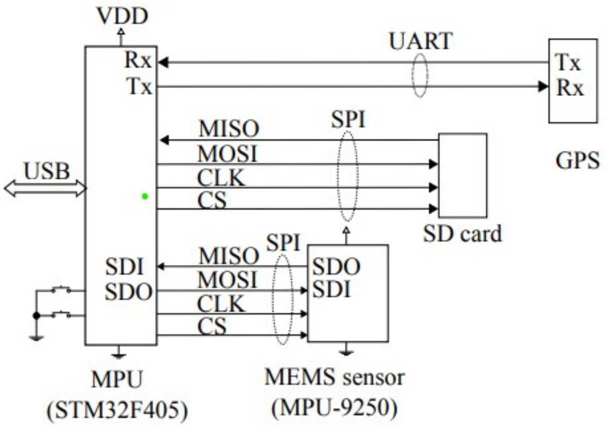


Figure 6. Diagram of IMU board [2].

3.0.1. Observation of The SPI Bus Signal

Initially, the SPI bus's signal lines are observed by an oscilloscope. Then, the measured result is manually decoded according to Fig. 7 and the MPU-9250 datasheet [6].

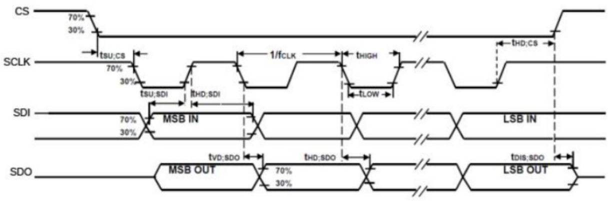


Figure 7. Timing chart of SPI data transfer [6].

3.0.2. Gravity Measurement Using The Accelerometer

The measured acceleration is sent to the computer. The relationship between the rotation angle of the IMU board and the sensor output is recorded. Spirit levels and a protractor are used to determine the rotation angle. This experiment determines the accelerometer's sensitivity (X and Y-axis).

3.0.3. Inertia Measurement

To begin, the IMU is in a stationary condition to determine the starting location and orientation (about 1 second). The IMU is then moved along the table's designated path. The detected acceleration and rotation signals are saved to the computer. The recorded data is then processed using GNU Octave [7]. The IMU's orientation is computed using gyroscope signals. The global acceleration can then be determined. By integrating these signals, the IMU's location and orientation may be computed. However, because sensor signals typically contain noise and bias, the calculated position diverges due to numerical integration. To reduce such a low-frequency offset, a digital filter is used. GNU Octave is used in this experiment to create a basic high-pass filter to improve the IMU outcome.

4. Results and Discussions

4.1. Assignment 0

Discussion: *Make a charts of SPI data transfer (CS, SCLK, MISO, MOSI). Decode the '1' or '0' data (MISO and MOSI).*

The device used in this experiment, the MPU-9250, has a configuration of *CPOL (***c***lock* ***pol***arity*)* $= 1$, which means the clock idles at 1, and each cycle consists of a pulse of 0. That is, the leading edge is a falling edge, and the trailing edge is a rising edge. Furthermore, the *CPHA (***c***lock* ***pha***se)* is also set to 1, meaning that the data is changed and captured during the falling edge and rising edge of the clock cycle, respectively. Figure 9 depicts how a bit of the MISO signal is captured [8].

As shown in Figure 8, the data transfer starts when the CS signal drops from 1 to 0. By looking at the SCLK signal, we can decode both the MOSI and MISO signal, where each bar of the SCLK signal indicates one byte of data transfer. In this experiment, there are 15 bars: the first bar belongs to MOSI, while the rest belongs to MISO. Moreover, we can observe that MOSI sends the signal first, followed by the MISO signal. This indicates that MOSI, as the master device, initiates communication with the slave devices.

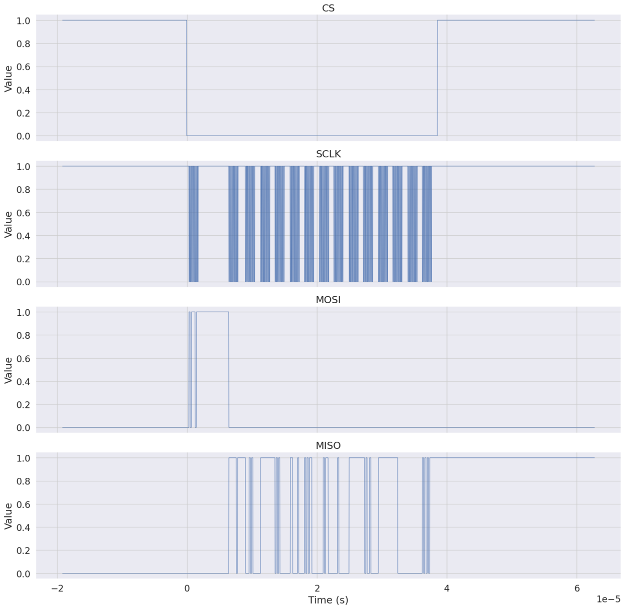


Figure 8. Measured SPI signals.

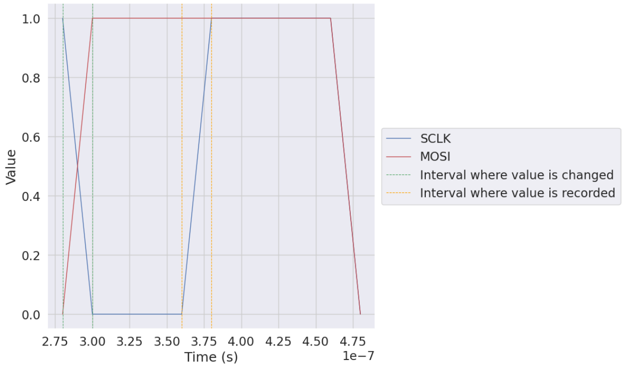


Figure 9. Detailed view of data changing and capturing by SCLK.

After thorough examination, bytes for each bar of SCLK signal is obtained as follows:

- MOSI signal: 10111011
- MISO signal: $11111100, 00010100, 11111111, 01010000, 11000010, 10101100, 00010110, 00010000, 11111111, 10100100, 11111111, 11110000, 00000000, 10101011$

From the datasheet [6], the corresponding data of each byte can be decoded from the register map. However, there is no information regarding the position of the Most Significant Bit (MSB) or the Least Significant Bit (LSB), meaning that the manner of reading the order of the byte is unknown. Once the MSB or LSB position is known, we convert the binary sequence into a decimal number and match it with the map table provided by the datasheet to check what the signal does.

If we assume that the MSB position is at the very-left part of the byte (the usual interpretation of a byte) and take an example of the MISO signal: 10111011. The first bit of the byte sequence is the CPHA bit that issues the first SCLK edge in CPHA $= 1$ format. Then, we convert the leftover bits to decimal, which in this case is equal to 59. After that, we look up the map register available in the datasheet [6]. For this byte, its register name is the ACCEL_XOUT_H, which contains the 8 most significant bits, *ACCEL _XOUT [15:8]*, of the 16-bit X-axis accelerometer measurement, *ACCEL _XOUT*.

4.2. Assignment 1

Discussion: *Make a chart of measured X- and Y- acceleration with respect to rotation angle.*

For this assignment, the chart between the measured acceleration for each angle is depicted in Figure 10. For both the X and Y axes, the shape of the acceleration plot follows a periodic function. This is true as when an axis direction coincides with the gravity acceleration, it will either have the maximum value (same direction) or minimum value (opposite direction). On the other hand, the acceleration is 0 if its direction is perpendicular to gravity. Moreover, since the X and Y axes are directed perpendicular to each other, the X axis is shifted by $90\,\mathrm{deg}$ preceding the Y axis. Separately, the Z axis only shows a constant value of 0 as it is not rotated.

4.3. Assignment 2

Discussion: *From the result, obtain the scale factor $(m/s2/LSB)$. The gravity acceleration is assumed to be* $9.806\,65\,\mathrm{m/s^2}$.

The measured acceleration in Figure 10 can be utilized to derive the scale factor, a constant used to convert the measured value to its real acceleration value. Since the value of the gravity acceleration is given, the scale factor is calculated as follow:

$$Scalefactor = \frac{amplitude}{2 \cdot gravity} \tag{7}$$

where amplitude is the value difference between the peak

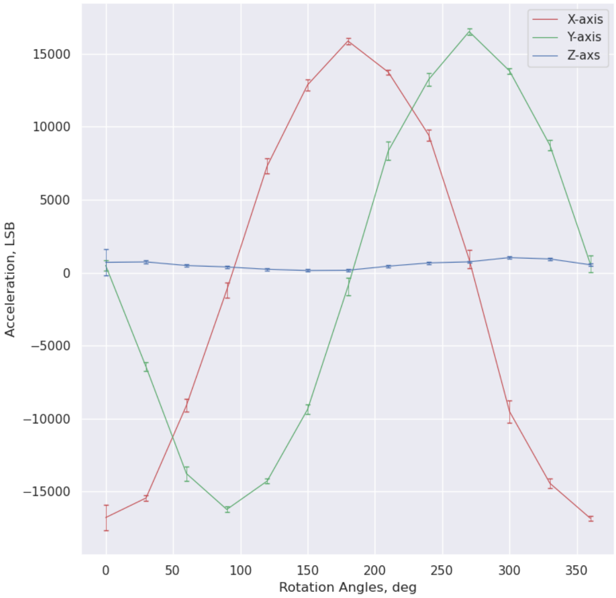


Figure 10. Measured X, Y, and Z axes acceleration with respect to rotation angle.

and the crest of the plot.

The scale factor is obtained for each axis: $1668.03\ LSBg^{-1}$ and $1670.06\ LSBg^{-1}$ for the X and Y axis, respectively. It can be seen that the values for both axes are consistent. However, there is a significant difference between these values with the value provided by the datasheet [9], where the real value is $16,384\ LSBg^{-1}$, 10 times bigger than the obtained values. This difference might be caused by several reasons, such as: (1) There is an error in processing the data; one example is a mistake in the measurement units. (2) Author makes an error in doing the analysis. However, the real value from the datasheet is used for the following experiments' analysis to ensure the obtained value is correct.

4.4. Assignment 3

Discussion: *Make a charts of Allan variances of X, Y and Z of the gyroscope and accelerometer. Discuss about the results.*

Plots for Allan variances of accelerometer and gyroscope in the X, Y, and Z axes are depicted in Figure 11. Figure 11(a) and 11(b) shows the plot of logarithmic value between the σ (see section 2.3) and the time. It can be seen that the trends for both sensors are quite similar on all axes. In the beginning, the value is quite high due to the gathered initial noise. However, as t increases, the deviation averages out to approximately 10^{-4} times its initial value. This shrinking deviation value is called the white noise, which is defined as a random signal with equal intensity at all frequencies, which means a constant power spectral density [10].

In addition, the minimum value of the deviation is called the Bias Instability, which is a measurement of how the bias will drift over time at a constant temperature [11]. This parameter also represents the best possible accuracy for estimating a sensor's bias. After some time, the Allan deviation escalates. This surge might be caused by the shift in the environmental condition like temperature changes. These arguments can be supported by observing the Figure 11(c) and Figure 11(d), which shows the oscillation of the measured temperature value.

4.5. Assignment 4

Discussion: *Make a X-Y (or X-Y-Z) plots of inertial measurement. (Without gyroscope, with gyroscope but not HPF, with both gyroscope and HPF) Discuss about the results.*

From the IMU Measurement experiment, we obtain the 3-axes acceleration and angular rate from the sensors: accelerometer and gyroscope. Then we integrate these data to get the device's coordinate over time. However, an intermediate step to transform the coordinate is needed as the measured coordinate is not the global coordinate, which is the real-world coordinate.

In addition, we also need to filter the data because when we add the orientation information, it adds noise and bias to the coordinate. Even though the noise is initially small, it eventually builds up over time, and when integrated, its magnitude becomes unbounded, completely obscuring the signal. The gravity acceleration influence must also be taken into account as it generates pseudo acceleration. However, these unwanted features can be rid of by applying the High-Pass Filter (HPF) algorithm to the data. The reason why HPF is used is because during measurement, the rotation angle changes in a slow manner, which means it only contributes to the low-frequency region. The same concept also applies to gravity influence: We only move the device in the X-Y plane; thus the value of the Z-axis acceleration does not transform very much. This also concludes that the HPF will only detect the fast movement of the device, which was the circular motion done in the experiment.

The 3D plots of the obtained coordinate for three cases: without gyroscope, with gyroscope but without filter, and with both gyroscope and filter are shown in Figure 12. From Figure 12(a), we can see that the plot creates a curve motion with high-spanning value on all three axes. This trend of having a large value and small curvature movement happens because the accelerometer does not receive the data about the rotation angle. After adding orientation information, the device's movement is more clearly depicted as a circular motion (Figure 12(b)). However, in this case, it can be seen that the center of rotation is not in point $(0, 0)$ of the X-Y plane and there is an offset occurring in the Z axis. This might be caused by low-frequency noise that has yet to be filtered exists in the data. Lastly, Figure 12(c) shows the clearest path of the movement of the device: rectangular-circular motion. Furthermore, the HPF works perfectly, removing the noise since the motion path no longer possesses Z-axis offset. Moreover, the center of rotation is now located in the center of the X-Y plane with an accurate radius value ($r = \pm 0.15\ m$)

4.6. Assignment 5

Discussion: *Study and report some digital communication methods (such as SPI, I2C, UART and so on).*

This assignment has been discussed in Section 2.4.

4.7. Assignment 6

Discussion: *Study and report about Allan variance.*

This assignment has been discussed in Section 2.3.

5. Conclusion

From this experiment, we learn about several inter-device communication methods and Allan variance. We also study to read a binary signal of the SPI method of an accelerometer and decode the signal to its corresponding data command. Finally, we discuss inertia measurement using accelerometer and gyroscope sensors and interpret the result of three different cases: without gyroscope, with gyroscope but without HPF, and with both gyroscope and HPF.

References

[1] *What is MEMS Technology?* URL: https://www.mems-exchange.org/MEMS/what-is.html (see p. 1).

[2] Shunsuke Yamada. “Fundamentals of Motion Sensing and Signal Processing”. In: 2022. URL: https://drive.google.com/file/d/1Vp9Pp0iVck93fgcKUB-4Xy0o1S4zOikO/view (see pp. 1–4).

[3] *Allan Variance*. URL: https://home.engineering.iastate.edu/~shermanp/AERE432/lectures/Rate%5C%20Gyros/Allan%5C%20variance.pdf (see p. 2).

[4] David Allan. “Statistics of Atomic Frequency Standards”. In: *Proceedings of the IEEE* 54.2 (Feb. 1966), pp. 221–230. DOI: https://doi.org/10.1109\%2FIEEESTD.1999.90575 (see p. 2).

[5] Jessica Hopkins. *I2C vs SPI vs UART – Introduction and Comparison of their Similarities and Differences*. Dec. 2021. URL: https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-comparison-similarities-differences/ (see pp. 2, 3).

[6] *MPU-9250 Register Map and Description*. Version 1.6. July 2015. URL: https://invensense.tdk.com/wp-content/uploads/2015/02/RM-MPU-9250A-00-v1.6.pdf (see pp. 4, 5).

[7] *GNU Octave*. URL: https://octave.org/ (see p. 4).

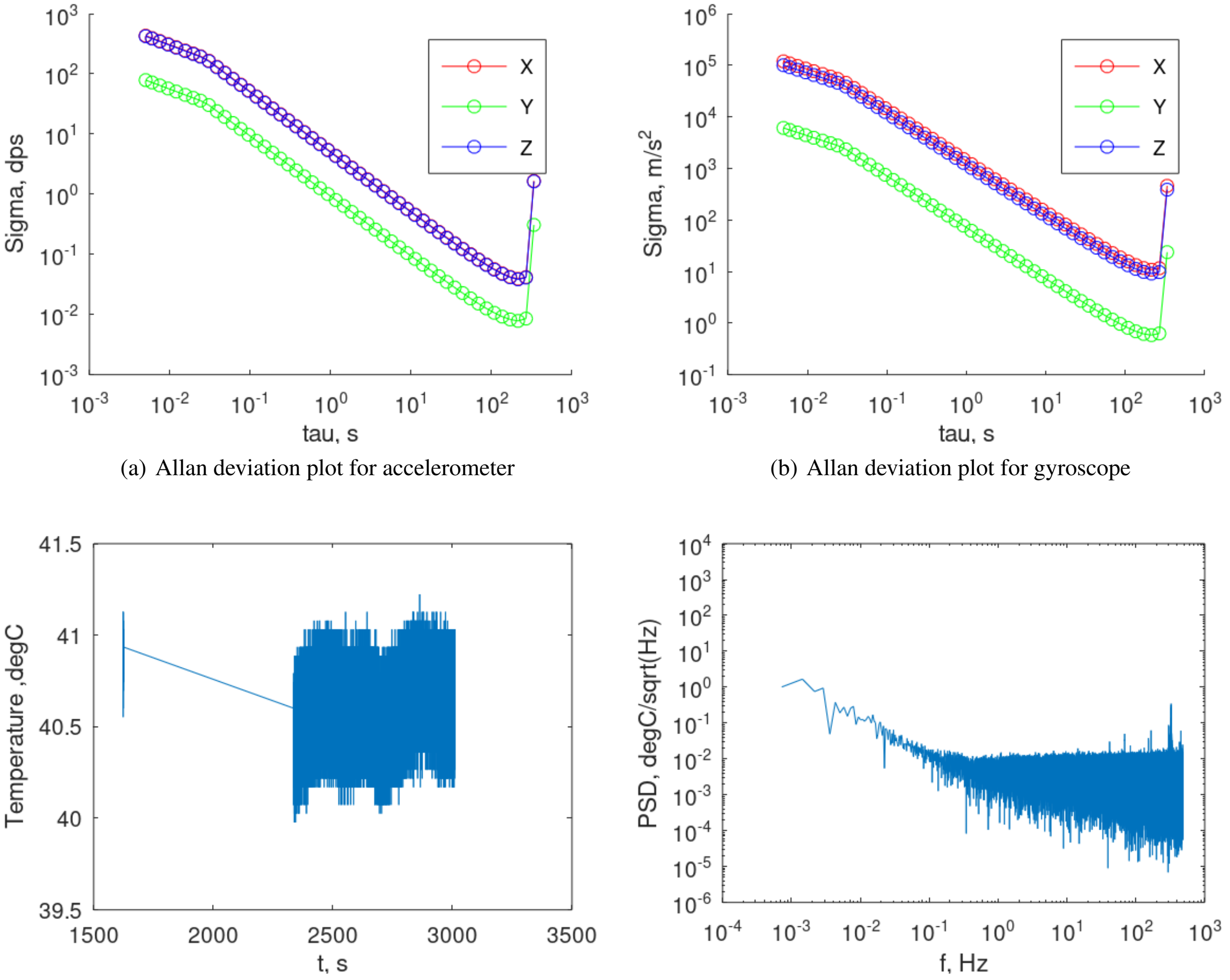
[8] *SPI Block Guide*. Jan. 2000. URL: https://web.archive.org/web/20150413003534/http://www.ee.nmt.edu/~teare/ee308l/datasheets/S12SPIV3.pdf (see p. 4).

[9] *MPU-9250 Product Specification*. Version 1.1. June 2016. URL: https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf (see p. 6).

[10] Bruce Carter. In: *Op Amps for Everyone*. Texas Instruments, 2009, pp. 10–11. URL: https://web.mit.edu/6.101/www/reference/op_amps_everyone.pdf (see p. 6).

[11] *IMU Specifications*. URL: https://www.vectornav.com/resources/inertial-navigation-primer/specifications--%20and--error-budgets/specs-imuspecs (see p. 6).

A. Result on Assignment 3



(a) Allan deviation plot for accelerometer

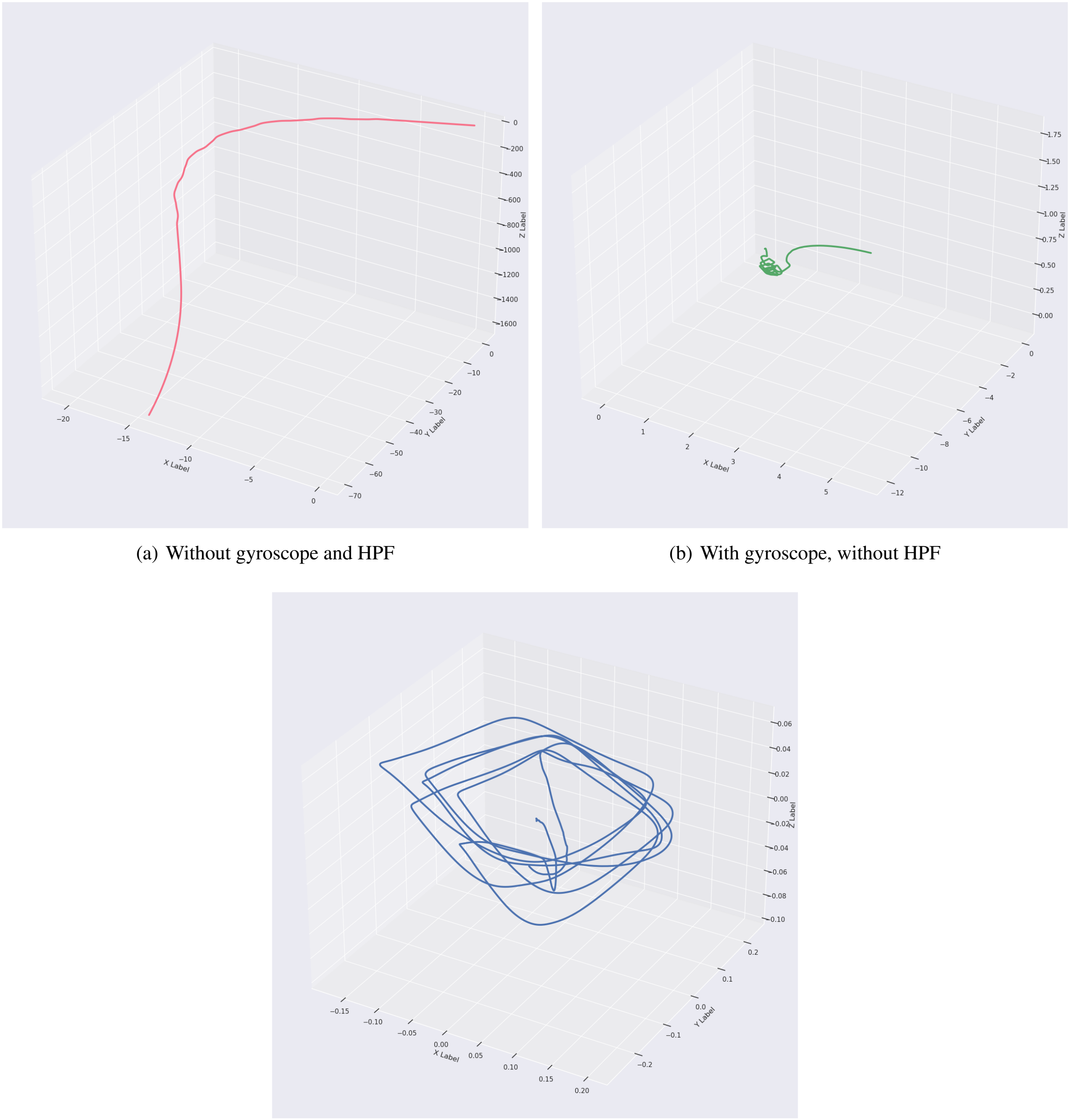
(b) Allan deviation plot for gyroscope

(c) Temperature measurement vs. time

(d) Frequency measurement vs. temperature measurement

Figure 11. Overall figures for Assignment 3

B. Result on Assignment 4



(a) Without gyroscope and HPF

(b) With gyroscope, without HPF

(c) With both gyroscope and HPF

Figure 12. Overall figures for Assignment 3