BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING
FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

DESIGN AND IMPLEMENTATION OF ALGORITHMS FOR COMPENSATING THE EFFECT OF LINEAR ACCELERATION ON ANGULAR VELOCITY MEASUREMENTS BY MEMS GYROSCOPES

NÁVRH A IMPLEMENTACE ALGORITMŮ PRO KOMPENZACI VLIVU LINEÁRNÍHO ZRYCHLENÍ NA MĚŘENÍ ÚHLOVÉ RYCHLOSTI MEMS GYROSKOPY

DOCTORAL THESIS
DIZERTAČNÍ PRÁCE

AUTHOR — **Ing. Tomáš Spáčil**
AUTOR PRÁCE

SUPERVISOR — **doc. Ing. Robert Grepl, Ph.D.**
ŠKOLITEL

BRNO 2023

Abstrakt

Tato Disertační práce se zabývá vývojem nového typu dopředného kompenzátoru, speciálně určeného k potlačení vlivu lineárního zrychlení a ryvu působícího na MEMS gyroskopické senzory. Z principu funkce inerciálního snímače, dochází při umístění gyroskopického senzoru mimo osu rotace k vzniku dalších nežádoucích sil (normálové a tečné), které negativně ovlivňují měření žádané úhlové rychlosti. Přítomnost těchto sil je však možno měřit nezávislým senzorem, akcelerometrem, a vhodným způsobem kompenzovat výstupní hodnotu gyroskopického senzoru a tím zpřesnit nepřímo měřenou hodnotu úhlové rychlosti. V první části práce identifikujeme vliv zmíněných zrychlení na MEMS gyroskopy, v drůhé části práce pak navrhneme a otestujeme různé kompenzátory, vyhodnotíme jejich efektivitu a v třetí části vhodný algoritmus implementujeme na RT HW.

Abstract

This Dissertation deals with the development of a new type of forward compensator, specifically designed to counteract the effect of linear acceleration and jerk acting on MEMS gyro sensors. From the principle of inertial sensor function, additional undesirable forces (normal and tangential) occur when the gyro sensor is placed off the axis of rotation, which negatively affect the measurement of the desired angular velocity. However, the presence of these forces can be measured by an independent sensor, the accelerometer, and the output value of the gyro sensor can be compensated for in a suitable way, thus refining the indirectly measured value of the angular velocity. In the first part of the paper we identify the effect of the mentioned accelerations on MEMS gyroscopes, in the second part of the paper we design and test different compensators, evaluate their effectiveness and in the third part we implement a suitable algorithm on RT HW.

Klíčová slova

MEMS senzory; gyroskop; akcelerometr; dopředný kompenzátor; mechatronika; Matlab; Simulink; FPGA; mikrokontrolér; výpočty v reálném čase.

Keywords

MEMS sensors; gyroscope; accelerometer; feedforward compensator; mechatronics; Matlab; Simulink; FPGA; microcontroller; real-time calculations.

Bibliographic citation

SPÁČIL, T *Design and implementation of algorithms for compensating the effect of linear acceleration on angular velocity measurements by mems gyroscopes.* Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2023. 100 pages, Supervisor: doc. Ing. Robert Grepl, PhD.

I hereby affirm that the presented doctoral thesis titled Design and Implementation of Algorithms to Compensate the Effect of Linear Acceleration on Angular Velocity Measurements in MEMS Gyroscopes is my genuine work and that it was created with the support of the stated references, under the supervision of the assoc. prof. Ing. Robert Grepl, Ph.D.

Ing. Tomáš Spáčil

Brno, 10.10.2023

Děkuji mojí ženě Denise za dlouhodobou podporu při studiu a našim dětem Albertovi a Teodorovi za to, že nám přinášejí radost do života.

Tímto bych chtěl poděkovat mému školiteli, doc. Ing. Robertu Greplovi, Ph.D., za jeho trpělivost a cenné rady nejen při zpracování této dizertační práce.

Dále bych rád poděkoval kolegům z Mechlabu za vytváření technicky a společensky synergického pracovního prostředí.

Ing. Tomáš Spáčil

Contents

List of Abbreviations

MEMS	Micro-electromechanical systems
SOC	System on a chip
IMU	Inertial measurement unit
RLG	Ring Laser Gyroscope
AHRS	Attitude and Heading Reference System
CCD	Charge-coupled device
KPI	Key performance indicator
SSR	Sum of squared residuals
NLLS	Non-linear least squares
RMSE	Root Mean Square Error
ANN	Artificial neural network
FNN	Feedforward neural network
RNN	Recurrent neural network
NARX	Nonlinear autoregressive exogenous neural network
DSL	Domain specific language
HLS	High level synthesis
FPGA	Field Programmable Gate Array
HDL	Hardware description language
RTL	Register-Transfer level
μC	Microcontroller
BLDC	Brushless DC motor
PMSM	Permanent magnet synchronous motors
PID	Proportional–integral–derivative controller

R^2	Regression squared residuals
ZRO	Zero rate output
LUT	Look-up table
FF	Flip flop
DSP	Digital signal processing
DUT	Device under test
MIPS	Million instructions per second
RT	Real time

1 Introduction

Gyroscopes constitute a specialized class of devices designed to measure the angular velocity of a target object relative to a fixed reference frame. These instruments find applications across a diverse spectrum of human endeavors, ranging from underwater exploration to aerospace systems, and are integral not only to crewed missions but also to autonomous navigation and guidance systems. Essentially, gyroscopic sensors furnish critical data regarding an object's orientation or its rate of change in orientation, referred to as angular velocity. MEMS (Micro-Electro-Mechanical Systems) gyroscopes operate on the principle of angular velocity measurement, from which orientation can be inferred by integration in time relative to an initial known state. Unique to this MEMS-based approach is its reliance on the Coriolis force, equation 2.7, which arises when there is a simultaneous presence of rotational and linear velocities along orthogonal axes. This force serves as the basis for deriving angular velocity measurements in MEMS gyroscopes.

The Coriolis force itself is an extraordinary physical phenomenon that is observable in everyday life because of Earth's rotation. It dictates the deflection of moving objects in the atmosphere and oceans, veering them to the right in the Northern Hemisphere and the left in the Southern Hemisphere. One of the most striking empirical validations of the Earth's rotation via the Coriolis effect is showcased by the Foucault pendulum experiment, artistically portrayed in Figure 1.1. Initiated by French physicist Léon Foucault in 1851, the experiment employs a large pendulum that swings freely, and its plane of oscillation rotates with time, thereby providing visual, empirical evidence for Earth's axial rotation.

Emerging in the late 20th century, MEMS technology represents an innovative fusion of mechanical engineering, electronics, material science, and computational methodologies. Pioneered by advancements in silicon-based fabrication techniques, originally cultivated within the semiconductor industry, MEMS facilitated the miniaturization of an array of sensors and actuators into integrated systems-on-chip (SOC). This miniaturization has yielded unprecedented gains in performance, reliability, and cost-effectiveness, revolutionizing diverse fields from healthcare and automotive systems to consumer electronics and environmental monitoring.

Finally, gyroscopes are commonly grouped under the category of Inertial Measurement Units (IMUs), which capture multiple forms of spatial data. While traditional gyroscopic systems depend on mass properties to measure forces and, by extension, angular velocity, contemporary designs have evolved to exploit alternative phenomena, like light interference, still yielding the same essential metric: the movement of a body.

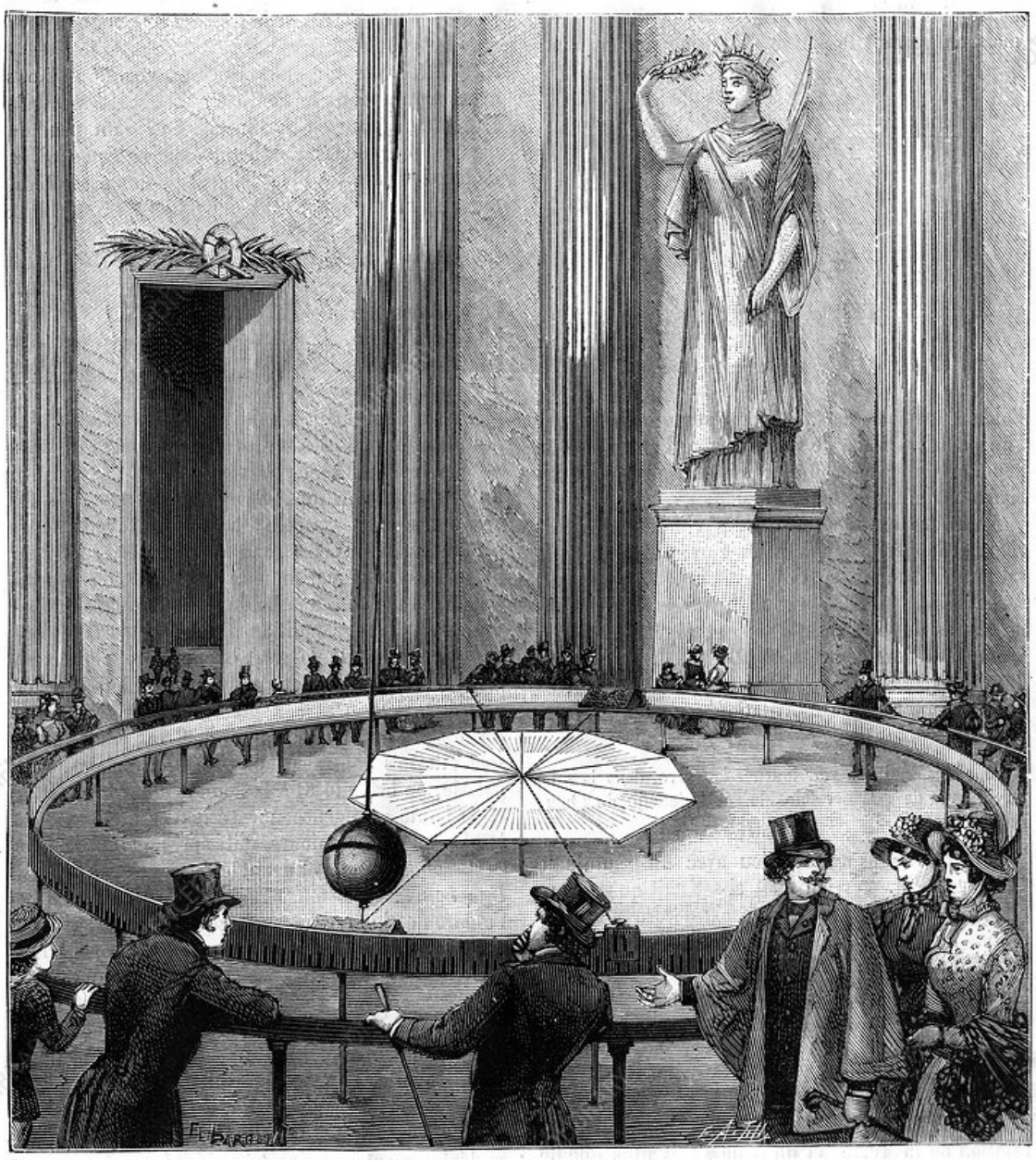

Figure 1.1: Foucault pendulum. [1]

2 Theoretical Survey

In the Mechatronics Laboratory (MechLab) at Brno University of Technology the development of an unstable single-axis two-wheel self-balancing personal transporter [2] was initiated, representing an excellent educational exemplar of a mechatronic system. This system is an example of synergic integration of signal processing, control, and mechanics, resulting in a precise mechatronic system with complex engineering methodologies. Several rigorous redesigns were realized, each aimed at enhancing its operational capabilities and accuracy. However, this project was not without its unique set of unforeseen challenges that required extensive exploration and detailed analysis.

Figure 2.1: Hummer 2011. [3]

During one of our investigative pursuits, a significant observation was made regarding the pronounced impact of the gyroscopic sensor's positioning on its measurement accuracy when it is misaligned with the rotation axis. This misalignment exposed a sensitivity to tangential acceleration, leading to alterations in sensor outputs and compromising the integrity of the data obtained. Empirical insights gained from studying the system's operational behavior led to concentrated efforts to mitigate such anomalies in our research. Given the inherent real-time nature of the personal transporter, special emphasis was placed on implementing algorithms specifically

tailored for the real-time controller.

In the context of MEMS gyroscopes, which are mass-based sensors, various complexities arise that can compromise the fidelity of measurements. These complexities are not confined solely to mechanical issues but extend to electrical considerations as well, as the inherent nature of MEMS fabrication is introduced. Within the scope of this research, our primary emphasis is placed on mechanical, respectively dynamical errors attributable to the fundamental force-mass interactions that govern the sensor's operation. This focus allows us to investigate the nontrivial relationship between dynamic forces and the resulting errors, providing an understanding that is crucial for developing effective compensatory algorithms.

2.1 Inertia sensors

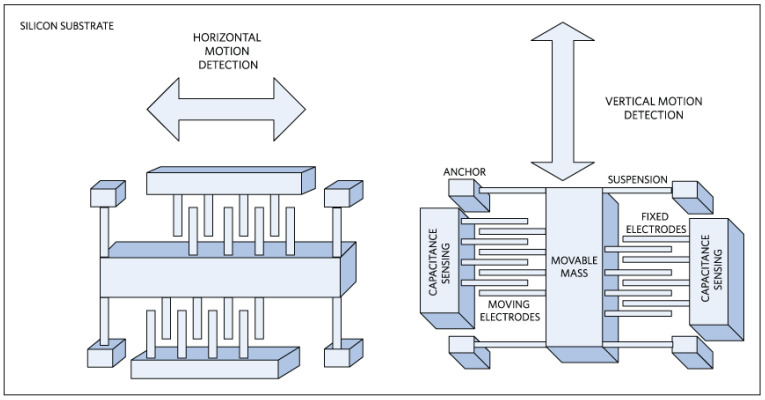

Figure 2.2: Accelerometer schematic. [4]

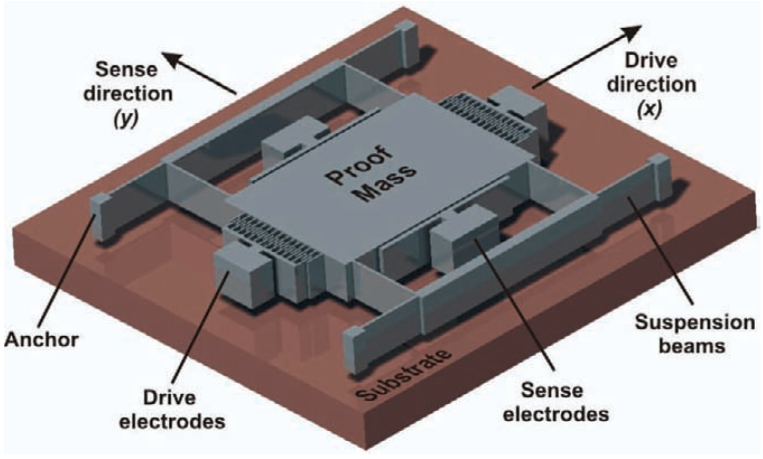

Figure 2.3: Gyroscope 3D model. [5]

Inertial sensors are key components in a broad range of systems, from consumer

electronics and robotics to aerospace and automotive applications. These sensors provide critical information about an object's movement and orientation within a physical space. Various types of inertial sensors exist, but we would point out two, elemental types, of them:

- Accelerometers: These sensors measure linear acceleration along one axis and potentially combine up to 3. They are extensively used in mobile devices for orientation detection, in automotive safety systems for airbag deployment, and robotics for motion control. Their principle is based on the deflection of a spring and measurement of such a deflection, as can be seen in Figure 2.2.

- Gyroscopes: Gyroscopes measure the angular velocity, essentially the speed at which an object rotates around its axis. They find applications in various fields including aerospace for attitude control, automotive systems for stability, and in consumer electronics like smartphones for orientation and navigation. Their principle is not as straightforward as in the case of accelerometers, so we describe them further.

2.1.1 Design of gyroscopic sensors

Numerous designs of gyroscopic sensors exist, tailored for various key properties like precision, stability, price, or robustness. From a rich history, we point out a few successful designs.

Mechanical gyroscopes

Given that aerospace was the primary field demanding gyroscopic technology to furnish essential orientation data during flight—addressing a notable limitation in human spatial orientation skills—the earliest gyroscopes were developed as mechanical systems, illustration of such in Figure 2.4. Mechanical aviation gyroscopes operate on the principles of angular momentum and gyroscopic inertia, featuring a high-speed spinning rotor mounted on a gimbal system for multi-axis freedom. The rotor resists changes in its orientation, and when an external force is applied, it exhibits gyroscopic precession, a movement perpendicular to the force. This precession is measured to determine the angular velocity and, consequently, the orientation of the aircraft. The data is then integrated into the aircraft's navigational and control systems.

MEMS gyroscopes

To describe the mechanical part of the structure, we chose a simple gyroscopic sensor with a single oscillating mass (Proof Mass) in Figure 2.3. The mechanical part of the MEMS gyro sensor consists of:

- Substrate. It can be directly a silicon wafer.

- Suspension beams. They are used for flexible attachment of the oscillating mass.

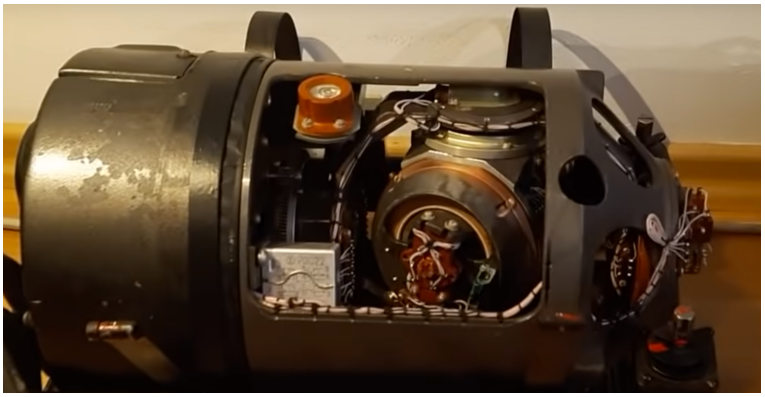

Figure 2.4: Mechanical gyroscope MiG21, 458MKC-15-32.

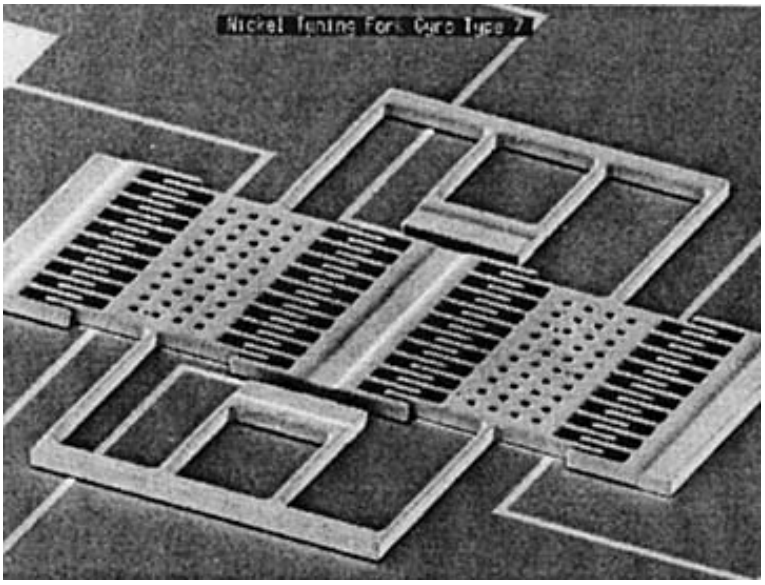

Figure 2.5: Draper lab, first commercial design of MEMS gyroscope [6]

- Oscillating Mass (Proof Mass). A mass that is used to measure the applied forces.
- Sense electrodes. Usually implemented as a capacitive distance sensor. The orientation of the capacitor surfaces corresponds to the sensing direction on the y-axis (Sense direction), into which the Coriolis acceleration is projected.
- Drive electrodes. They are usually implemented as an electrostatic actuator that generates the force necessary to move the oscillating mass in the controlled direction on the x-axis (drive direction).
- Anchor. Fix the moving mechanism to the substrate.

In addition to the mechanical part, the MEMS gyroscope includes electronic elements such as voltage regulators, amplifiers, or filters. The complete MEMS gyroscopic sensor is schematically shown in Figure 2.3. Note: The sensor in Figure

2.5 contains two oscillating masses. The principle of the function of the gyroscopic sensor is to transfer mechanical energy from one actively oscillating axis through the Coriolis force to the other, measuring axis. In the most common form, the active element consists of only one oscillating mass, but there are also gyroscopic sensors that have more oscillating masses (typically 2) and these oscillate with different phase shifts, which favorably affects typically improving the resistance to disturbing accelerations.

Ring Laser Gyroscope

Ring Laser Gyroscopes (RLG) utilize the Sagnac effect to measure angular velocity. In an RLG, a laser beam is split into two counter-propagating beams that traverse a closed-loop optical path. When the gyroscope rotates, the beams experience a phase difference due to the varying path lengths, which is detected as an interference pattern when the beams recombine. This phase difference is proportional to the angular velocity of the system, and by integrating this data over time, the angular orientation can be determined. RLG offers high sensitivity and accuracy, making them ideal for applications like inertial navigation systems in aerospace and maritime contexts. Schematically visualized in Figure 2.6.

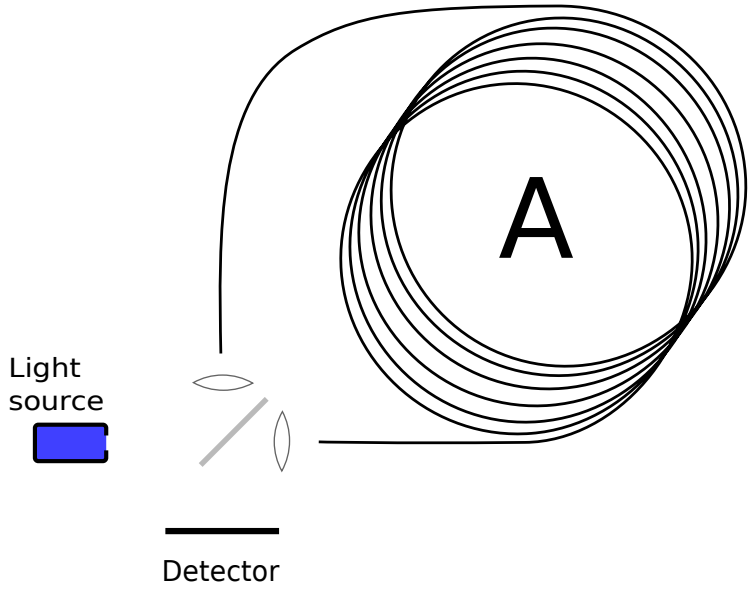

Figure 2.6: Schematic of optical gyroscope. [7]

Atom interferometry based inertia sensor

In 1924, Louis de Broglie introduced the concept that matter, like light, exhibits wave-like properties, a theory with implications for atom interferometry. Atom interferometers typically operate in high-vacuum environments, using a magneto-optical trap to cool and tie rubidium atoms. Utilizing intersecting laser beams and anti-Helmholtz coils, the trap creates a magnetic field gradient that keeps atoms

at its center, where they reach microkelvin temperatures. Upon trap deactivation, specialized laser pulses induce a quantum superposition of states in the atoms, altering their momentum and causing self-interference. A low-power laser collapses this superposition, generating an interference pattern captured by a CCD camera. An illustration of such a principle is in Figure 2.7. This pattern facilitates highly precise measurements of angular velocity and acceleration, simultaneously. [8]

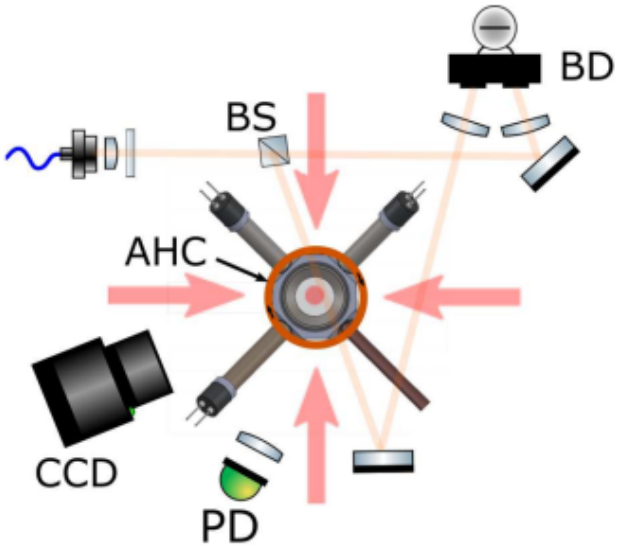

Figure 2.7: Atom intereferometry. [9]

2.1.2 IMU, AHRS

An Inertial Measurement Unit (IMU) primarily consists of accelerometers and gyroscopes, to measure linear accelerations and angular velocities. It outputs raw, high-frequency data ideal for short-term applications but is susceptible to drift errors over time. On the other hand, an Attitude and Heading Reference System (AHRS) incorporates data from an IMU and additional sources like magnetometers or GPS. Using complex algorithms such as sensor fusion techniques [10], AHRS provides a stable and corrected orientation output, making it suitable for long-term, accurate applications.

The key differences between the two lie in the complexity and type of data output, error correction mechanisms, and computational load. While IMUs provide raw sensor data without corrections, AHRS offers refined, filtered, and corrected data. AHRS systems usually include additional data sources and require more computational resources for data fusion and error correction, distinguishing them from IMUs in both functionality and application. Schematically we can discretize the concepts as in Figure 2.8

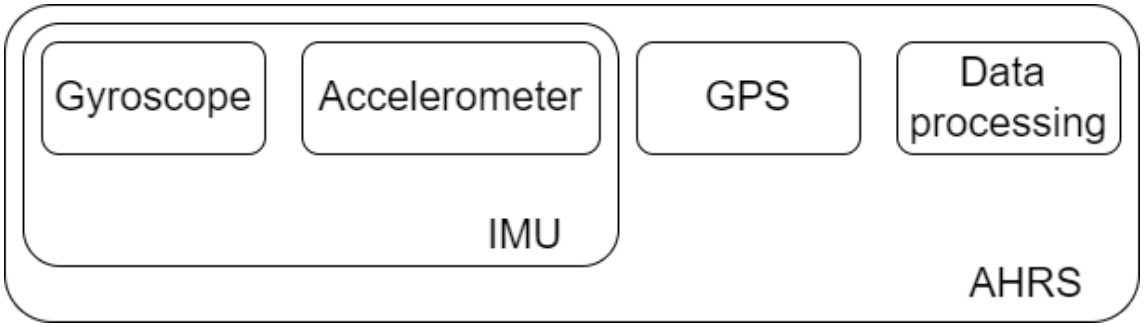

Figure 2.8: Classification of inertial units.

2.2 Model of MEMS gyroscopic sensor

Oscillatory MEMS gyroscopic sensors operate based on a vibrating structure, usually a micro-scale mechanical resonator, as opposed to the spinning rotor found in traditional mechanical gyroscopes. The resonator is set into oscillation along a particular "drive" axis. When the sensor experiences an angular rotation about an axis orthogonal to the drive axis, the Coriolis effect comes into play, giving rise to the Coriolis force. The relation of coordinate systems in the scope of the problem and the origin of Coriolis force can be seen in Figure 2.9.

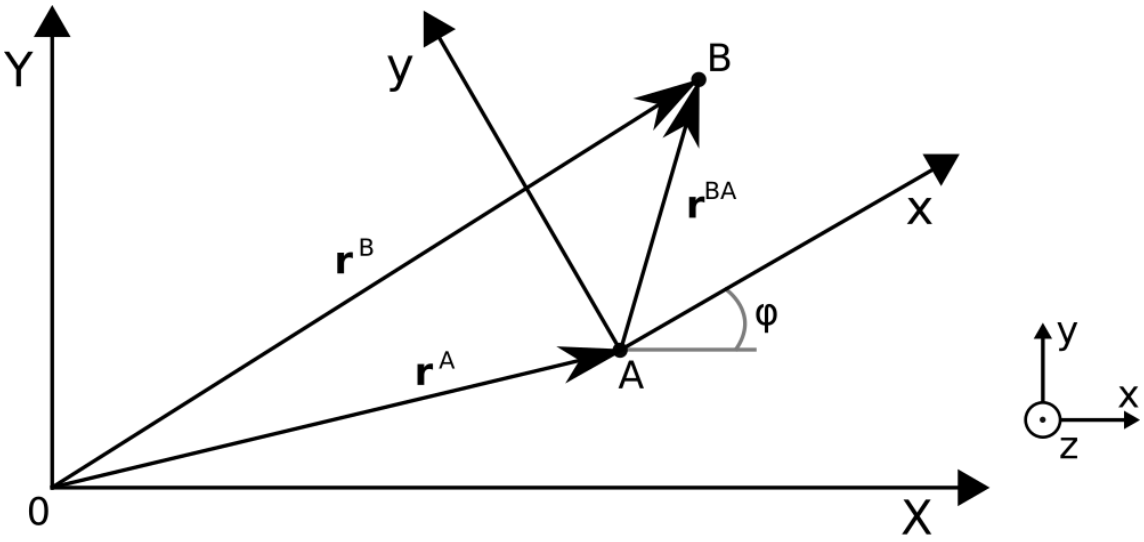

Figure 2.9: Substrate "A" vs. proof mass "B". [11]

$$\boldsymbol{r}^B = \boldsymbol{r}^A + \boldsymbol{r}^{BA} \tag{2.1}$$

$$\dot{\boldsymbol{r}}^B = \dot{\boldsymbol{r}}^A + \dot{\boldsymbol{r}}^{BA} + \dot{\boldsymbol{\varphi}} \times \boldsymbol{r}^{BA} \tag{2.2}$$

$$\ddot{\boldsymbol{r}}^B = \ddot{\boldsymbol{r}}^A + \ddot{\boldsymbol{r}}^{BA} + \dot{\boldsymbol{\varphi}} \times \dot{\boldsymbol{r}}^{BA} + \dot{\boldsymbol{\varphi}} \times \left(\dot{\boldsymbol{\varphi}} \times \boldsymbol{r}^{BA}\right) + \ddot{\boldsymbol{\varphi}} \times \boldsymbol{r}^{BA} + \dot{\boldsymbol{\varphi}} \times \dot{\boldsymbol{r}}^{BA} \tag{2.3}$$

$$\ddot{\boldsymbol{r}}^B = \ddot{\boldsymbol{r}}^A + \ddot{\boldsymbol{r}}^{BA} + \boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \boldsymbol{r}^{BA}\right) + \boldsymbol{\alpha} \times \boldsymbol{r}^{BA} + 2\left(\boldsymbol{\omega} \times \dot{\boldsymbol{r}}^{BA}\right) \tag{2.4}$$

$$m\ddot{x} + c_x\dot{x} + k_x x = F_{i,x} + F_{r,x} + m\omega_z^2 \mathrm{x} + m\dot{\omega}_z y + 2m\omega_z \dot{y} \tag{2.5}$$

$$m\ddot{y} + c_y\dot{y} + k_y y = F_{i,y} + F_{r,y} + m\omega_z^2 \mathrm{y} - m\dot{\omega}_z x - 2m\omega_z \dot{x} \tag{2.6}$$

where m is the weight of the seismic mass, c is the damping constant, k is the spring constant, F_i is the force acting on the seismic mass resulting from the movement of the rotating frame in the inertia frame, F_r is the force resulting from the electrostatic actuator, $m\omega^2 x$ ($m\omega^2 y$) is the term representing the centrifugal force, $m\omega\dot{x}$ ($m\omega\dot{y}$) is the term representing the Euler force, and $2m\omega x$ ($2m\omega y$) is the term representing the Coriolis force. These equations were derived in our previous work [11].

The Coriolis force is a direct consequence of the sensor's rotation within its frame of reference. This force arises as a secondary oscillation orthogonal to the original drive oscillation, typically referred to as the sense axis. Essentially, when the gyroscope experiences an angular velocity ω about an axis perpendicular to the drive axis, the Coriolis force F_C is generated, which is proportional to ω and the velocity v of the vibrating structure. The formula for the Coriolis force is derived in 2.4 as the following term 2.7.

$$\boldsymbol{F_C} = 2m\left(\boldsymbol{\omega} \times \dot{\boldsymbol{r}}^{BA}\right) \tag{2.7}$$

This force is then detected as a change in the amplitude or frequency of the oscillation along the sense axis. By measuring this change, the gyroscope can accurately determine the rate of angular rotation ω that caused the Coriolis force to be generated in the first place.

2.2.1 Sense electrodes

The sensor electrodes are typically made as a transducer–such as capacitive, piezoelectric, or piezoresistive elements–specifically aligned along the sense axis to convert this mechanical motion into an electrical signal. In a capacitive MEMS gyroscope, for example, the oscillation along the sense axis changes the distance between the capacitive plates, thereby altering the capacitance, 2.8 which is subsequently measured. This change in capacitance is directly proportional to the rate of angular rotation experienced by the sensor. Similarly, piezoelectric or piezoresistive mecha-

nisms might convert the mechanical stress or strain caused by the Coriolis-induced oscillation into a measurable electrical signal.

$$C = \varepsilon_0 \frac{wl}{d} \tag{2.8}$$

, where C is the resulting capacity, ϵ_0 is the dielectric constant. According to the Figure 2.10 the capacitance may be converted to voltage as equation 2.9.

$$U = \frac{Q}{C} = \frac{\int i(t)dt}{C} \tag{2.9}$$

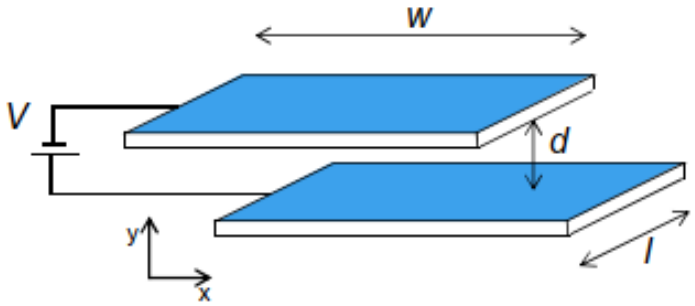

Figure 2.10: Parallel plate capacitor.

2.2.2 Drive actuator

The driving actuator (drive mode) is often accomplished through a microscale mechanical resonator or oscillating structure, typically fabricated using silicon-based microfabrication techniques. The resonator is set to forced oscillation through electrostatic, piezoelectric, or other actuation mechanisms. The primary oscillation along the drive axis establishes the baseline condition for the gyroscope. This constant vibration serves as the reference motion against which changes induced by external angular velocities are measured. Essentially, the drive axis acts as the inertial frame in the context of the gyroscopic sensor. In the case of an electrostatic actuator, the force produced can be in such a way that:

$$U = \frac{1}{2}CV^2, C = \varepsilon_0 \frac{wl}{d}, V = Ed \tag{2.10}$$

$$F = -\frac{dU}{dy} = -\frac{\varepsilon_0}{2}\frac{d}{dy}\left(wld(y)E^2\right) = -\frac{\varepsilon_0}{2}wlE^2 \tag{2.11}$$

, where U is the potential energy, E is the intensity of the electric field, and w,

l, d are the dimensions according to the previous picture.

In MEMS gyroscopes, the stability of drive-mode oscillation is crucial as it directly influences the Coriolis force, thereby affecting the sense-mode response. To ensure stable amplitude, phase, and frequency in drive mode, gyroscopes commonly operate at their resonant frequency. This is facilitated by a self-resonance mechanism, employing an amplitude-regulated positive feedback loop. This loop stabilizes the oscillation by locking it to the resonant frequency and employs an Automatic Gain Control (AGC) loop to maintain a constant amplitude, optimizing performance, and minimizing excitation voltages in steady-state operation. The schematic of such a control loop is in Figure 2.11.

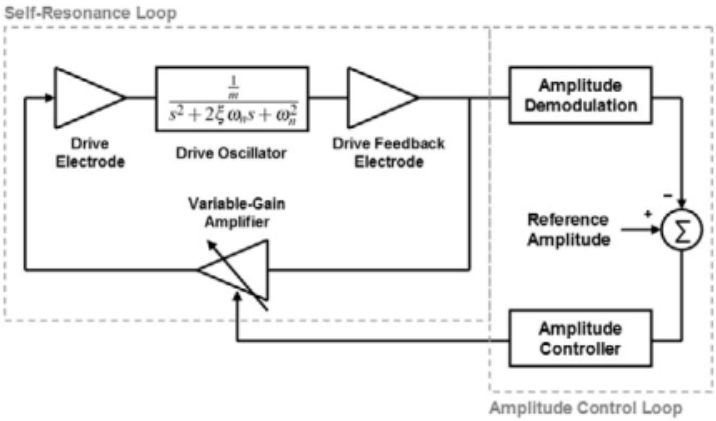

Figure 2.11: Drive mode control schematic. [5]

2.3 Sources of error

Potential sources of error affecting the angular velocity measurement of MEMS gyroscopes generally include geometric inaccuracies in manufacturing, vacuum leakage (gyroscope oscillations are damped by the environment), natural frequencies of the oscillating mass relative to the operating conditions, thermal instability, direct coupling between driving and sensing electrodes, known as quadrature error [12], or immunity to unwanted accelerations from the outside [13].

Due to the indirect measurement of the angular velocity through the Coriolis force, the application of any additional force to the oscillating sensor mass affects the measurement. Forces in scope typically result from vibrations [14] or the operation frequency that can be close to the resonant frequency of the gyroscope or its fraction. In subsequent chapters, we will explore various filtering principles commonly employed in both software and hardware contexts.

2.4 Methods of HW compensation

In the initial design phase of a MEMS gyroscope, establishing key performance indicators (KPIs) is paramount [15]. These KPIs not only influence the mechanical aspects but also determine the operational mode of the gyroscope. Fundamental parameters, such as measurement range, actuation and sensing electrode types, resonator feedback control circuit design, and sensor measurement bandwidth, are pivotal. Additionally, gyroscope-specific attributes, such as the choice between single or dual mass, mode-matching techniques, and the inclusion of active cancellation electrodes, require thorough consideration. These factors play a vital role in ensuring accurate and stable measurements across various operational conditions. In the subsequent chapter, we will delve deeper into these gyroscope-specific properties, emphasizing their contribution to enhancing measurement robustness and mitigating the impact of dynamic disturbances, thereby underscoring their importance in optimizing the reliability and precision of MEMS gyroscopes in diverse applications.

Mode matching and acceleration cancellation electrodes

When the sensor is subjected to constant angular rotation, the proof mass demonstrates oscillatory deflection along the sense axis not a constant state, attributed to the oscillatory movement in the drive direction. Consequently, a discrepancy could arise between the frequencies of the input drive force and the output sensing signal. This discrepancy is susceptible to alterations in the mechanical properties of the sensor, potentially deviating from the resonant frequency at which the sensor is calibrated. To mitigate the impacts of such discrepancies, mode-matching techniques are methodically incorporated directly into the chip, serving to align the resonant frequencies of the drive and sense modes more closely [16]. Referring to Figure 2.12, these implementations are depicted as frequency-tuning electrodes, strategically placed to fine-tune the resonant frequencies and thereby enhance the sensor's reliability and accuracy in detecting angular velocities. These improvements not only optimize the sensor's performance under constant angular rotations but also fortify its resilience against environmental and operational variances, ensuring the performance of precise and consistent readings across diverse applications and conditions. In addition, the referenced image reveals a proposal for a novel implementation of acceleration cancellation sensors and actuators [17], marked in Figure 2.12 as "Quad. Electrodes" and "Acce. Can. Elect." respectively. When a deflection, which is out of resonant frequency, is detected at the "Quadratic Electrodes", an opposite phase signal is applied to the "Acce. Can. Elect.". This application effectively mitigates dynamic disturbances, enhancing the robustness and precision of the sensor's readings under varying conditions.

Dual mass gyroscopes

A dual mass gyroscope, visualized in Figure 2.13 features two proof masses designed to oscillate in opposing directions along the drive axis, producing an anti-phase motion. Such a design inherently ensures system balance and supports the gyro-

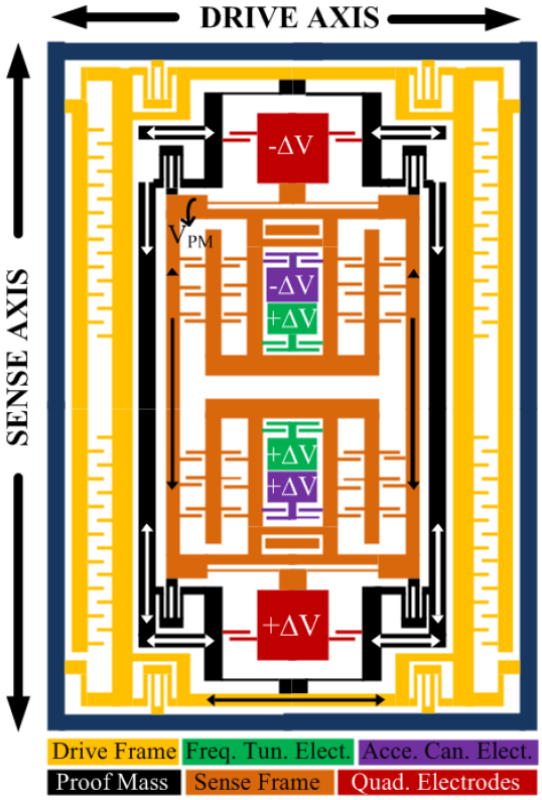

Figure 2.12: Gyroscope with mode-matching and acceleration cancellation electrodes. [18]

scope's stability, making it resilient against external disturbances, which is pivotal for accurate angular velocity measurement. The differential measurement between the masses delivers better sensitivity, effectively negating common-mode noise and linear acceleration, thus improving the linearity and stability of the response. As a result, dual-mass gyroscopes demonstrate remarkable stability for quadrature errors, amplifying their measurement accuracy and reliability. Their enhanced precision makes them a choice for high-end applications such as inertial navigation systems in vehicles and aircraft. However, the complexity of their design can lead to increased manufacturing costs, and potential synchronization challenges between the two masses, and might require sophisticated feedback control algorithms. Still, in the grand scheme of MEMS gyroscopic technology, dual mass gyroscopes emerge as a standout, especially in contexts that prioritize precision and resistance to environmental perturbations.

2.5 Methods of SW compensation

2.5.1 Complementary filter

The complementary filter is a commonly used technique in the field of inertial measurement units (IMUs) to combine the outputs of different sensors and obtain an

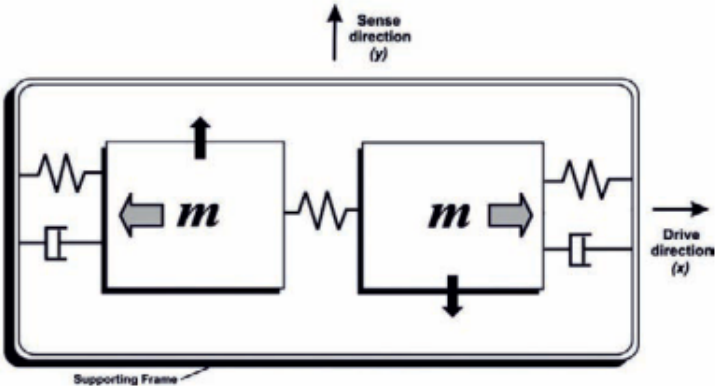

Figure 2.13: Schematic drawing of dual mass MEMS gyroscope. [5]

accurate estimation of an object's orientation [19]. The complementary filter utilizes a combination of low-pass and high-pass filters. The functional principle of the complementary filter lies in the inherent strengths of the accelerometer and gyroscope sensors. Accelerometers measure linear acceleration, in this case especially the gravitational as a reference orientation, while gyroscopes measure angular velocity. By fusing the outputs of these two sensors, the complementary filter compensates for the limitations of each individual sensor.

The complementary filter implementation involves a two-step process as seen in Figure 2.14. First, the accelerometer data is passed through a low-pass filter to attenuate high-frequency noise and extract the gravity component [20]. This low-pass filter is designed as an infinite impulse response (IIR) filter, which allows it to achieve a balance between smoothness and responsiveness. By emphasizing the gravity component, the filter effectively estimates the orientation of the object in relation to the Earth's gravitational field.

Simultaneously, the gyroscope data is processed through a high-pass filter, also implemented as an IIR filter. This high-pass filter suppresses low-frequency drift, which is inherent in gyroscope measurements due to sensor design and manufacturing errors. The resulting output represents the short-term changes in angular velocity, eliminating the long-term drift component.

In the second step, the filtered accelerometer and gyroscope outputs are combined. The complementary filter applies a weighted fusion of these two signals, where the gyroscope's output contributes to short-term changes in orientation, and the accelerometer's output provides an estimate of the overall tilt angle. The weights assigned to each sensor's output can be adjusted based on the specific application requirements and sensor characteristics.

The ease of implementation is one of the key advantages of the complementary filter. It requires minimal computational resources and can be efficiently implemented on microcontrollers or embedded systems. The filter's simplicity allows for real-time estimation of orientation without significant delays or complex algorithms, making it suitable for applications with limited processing capabilities.

The filter's adaptability is another valuable aspect. The fusion weights assigned

to the accelerometer and gyroscope can be adjusted based on the specific application requirements. These weights determine the trade-off between responsiveness and stability. For instance, in applications where rapid changes in orientation are crucial, a higher weight can be assigned to the gyroscope, prioritizing its responsiveness, however, this might introduce undesired accelerometer characteristics, such as noise, into the filtered signal. Conversely, in scenarios where stability is paramount, a higher weight can be allocated to the accelerometer.

In discrete form, the complementary filter should be described as

$$\varphi_n = \alpha \left(\varphi_{n-1} + \omega_n \Delta t\right) + (1 - \alpha) g_n \tag{2.12}$$

$$\alpha = \frac{\frac{\tau}{\Delta t}}{1 + \frac{\tau}{\Delta t}} \tag{2.13}$$

,where φ is the angle, ω is the rotational rate, g is the tilt defined by the accelerometer, α is the weighting parameter between the measured values, τ is the cut-off frequency of the IIR filter, and δt is the time step.

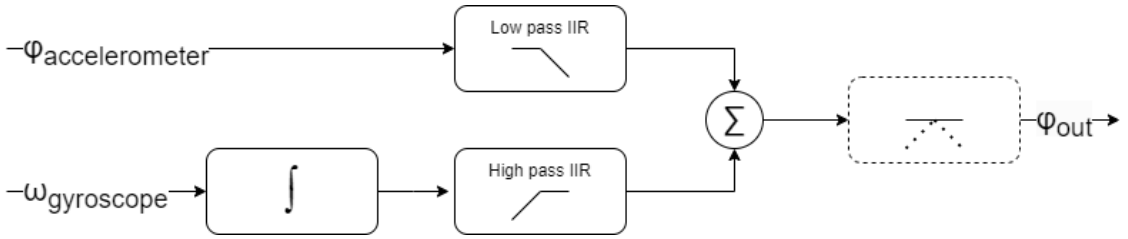

Figure 2.14: Schematical implementation of complementary filter.

In one of our previous studies [11] we have explored the possibilities of this approach for the compensation of linear dynamic disturbances of the gyroscope with unsatisfactory results. As can be seen from the Figure 2.15.

Summary

The complementary filter is a powerful technique for combining accelerometer and gyroscope data from an IMU. By utilizing IIR filters and fusing the filtered outputs, it provides an accurate estimation of an object's orientation, effectively compensating for the limitations of each sensor. The filter's ease of implementation and long-term stability, robustness to noise, and adaptability make it a popular choice in a wide range of applications, including robotics, navigation systems, virtual reality, and motion tracking. Nonetheless, this particular filter does not align with our objective of compensating for the linear acceleration impacting MEMS gyroscopic sensors as visible in Figure 2.15.

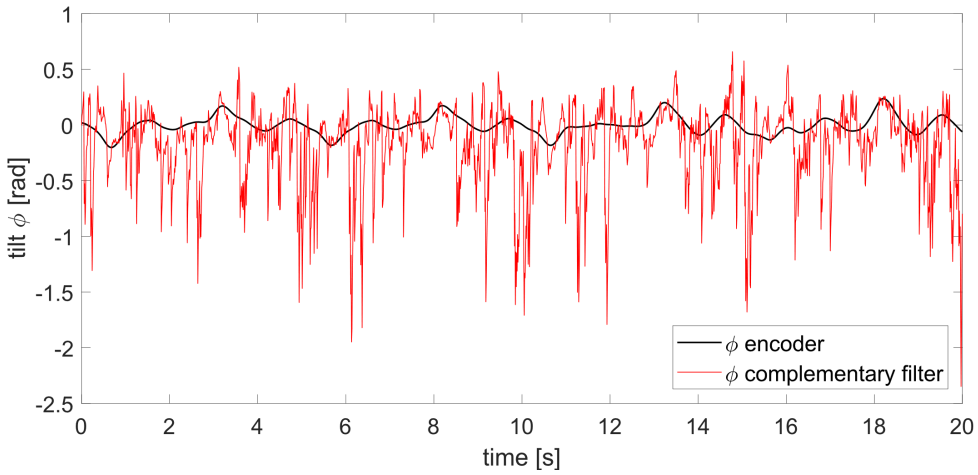

Figure 2.15: Complementary filter as a linear dynamic compensator. [11]

2.5.2 Kalman filter

In general, the most versatile tool for a wide range of engineering applications, the Kalman filter, can be used to correct and fuse measurements from various sensors, including accelerometers and gyroscopes. Having a dynamical model and measurement, i.e. signal consisting of the actual value, the real state, corrupted by normally distributed noise, we can leverage all that information by implementing the Kalman filter into signal processing and potentially reveal the states that are not even measurable. All of these assume that the noise in the model and measurements are quantified, experimentally or analytically. Kalman filter working principle can be categorized into 4 working steps described in the following text.

Initialization

- State estimate $\hat{x}_0$: Initial state estimation of the system.
- Error covariance matrix P_0: initialize the error covariance matrix, representing the estimated accuracy of the state estimate.

Prediction

- State prediction $\hat{x}_k^-$: predict the next state using the defined dynamical model

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{2.14}$$

- Error covariance prediction P_k^-: Predict the error covariance.

$$P_k^- = AP_{k-1}A^T + Q \tag{2.15}$$

, where A is the state transition matrix, B is the control-input model matrix

related to the control input u, and Q is the process noise covariance matrix, representing the uncertainty in the model.

Update

- Kalman gain K_k: Compute the Kalman Gain, which determines how much the prediction should be corrected based on the new measurement.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{2.16}$$

- State update $\hat{x}_k$: Correct the state prediction using the Kalman Gain and the measurement residual.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.17}$$

- Error covariance update P_k: update the error covariance matrix

$$P_k = (I - K_k H)P_k^- \tag{2.18}$$

, where H is the measurement model matrix, mapping the state to the measurement domain, R is the measurement noise covariance matrix, representing the uncertainty in the measurements, z_k is the actual measurement, and I is the identity matrix.

Loop

- Repeat steps of prediction and update for each time step as new measurements become available, continually updating the state estimate and error covariance.

Summary

The Kalman Filter operates recursively through prediction and update cycles, using a series of noisy measurements to produce optimal state estimates for linear dynamic systems. By iteratively predicting and correcting, it minimizes the estimation error covariance and efficiently tracks the true state of the system, even in the presence of process and measurement noise. Alternatives for nonlinear systems and computationally efficient algorithms exist like Extended Kalman Filter, which still has its limitations with highly nonlinear dynamics and scenarios lacking robust priori state estimates. [21]

2.5.3 Linear compensation

According to the paper [22], it is proposed to compensate for linear acceleration using 3-axis acceleration sensing and a linear mapping compensation algorithm. The authors implement both the accelerometer and the compensation algorithm directly

into the gyroscopic sensor as an analog controller. They formulate the gyroscope sensitivity as a multiplicative error, which can be interpreted in the following matrix format.

$$\begin{bmatrix} \delta\omega_x \\ \delta\omega_y \\ \delta\omega_z \end{bmatrix} = \begin{bmatrix} \beta_x \\ \beta_y \\ \beta_z \end{bmatrix} + \begin{bmatrix} \beta_{xx} + G_{xx}a_y & \beta_{xy} & \beta_{xz} \\ \beta_{yx} & \beta_{yy} + G_{yy}a_z & \beta_{yz} \\ \beta_{zx} & \beta_{zy} & \beta_{zz} + G_{zz}a_x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.19)$$

where $\delta\omega_{x,y,z}$ is the angular velocity error, $\beta_{x,y,z}$ is the bias error, $\beta_{xx,yy,zz}$ is the multiplicative factor error, $G_{xx,yy,zz}$ is the sensitivity to linear acceleration, $a_{x,y,z}$ is the acceleration in the corresponding axes, ω_x is the actual angular velocity.

Summary

In the proposed algorithm no dynamic effects are taken into account, what we consider fundamental. The approach provides an enhanced method for decoupling each gyroscope axis from each other and naturally present gravitational acceleration.

2.5.4 (Non)linear Least Squares methods

Linear Least Squares Method

This method endeavors to find the linear equation that minimally sums the squared errors between the observed values and those predicted by the model [23]. The notion of linearity in this context primarily refers to the parameters (or coefficients) of the model [24], signifying that each term is either a constant or the product of a parameter and an independent variable. Thus, this method is particularly useful when the parameters are linear, optimizing the fit of the model to the data points by minimizing the sum of the squared residuals. Given a set of n data points (x_i, y_i), the linear model can be represented as:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad (2.20)$$

where y_i is the dependent variable, x_i is the independent variable, β_0 and β_1 are the parameters to be estimated, and ϵ_i is the error term. The objective is to minimize the sum of squared residuals (SSR):

$$SSR = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2 \quad (2.21)$$

The Normal Equations are derived by taking the derivative of SSR with respect to the parameters and setting them to zero. For linear least squares, the Normal Equations are as follows:

$$X^T X\beta = X^T Y \quad (2.22)$$

Where X is the design matrix containing a column of ones and the observed values for the independent variable, β is the vector of coefficients, and Y is the vector of observed values for the dependent variable. The solution for β is given by:

$$\beta = (X^T X)^{-1} X^T Y \tag{2.23}$$

Non-Linear Least Squares Method

When the relationship between the dependent and independent variables is non-linear, the "Non-linear Least Squares method" is used. The model can be represented as:

$$y_i = f(x_i, \theta) + \epsilon_i \tag{2.24}$$

here, f is a non-linear function, and θ is the vector of parameters to be estimated. The Non-linear Least Squares objective is to minimize the sum of squared residuals:

$$SSR(\theta) = \sum_{i=1}^{n} (y_i - f(x_i, \theta))^2 \tag{2.25}$$

Solving non-linear least squares problems is generally more complex than linear ones and usually requires iterative numerical methods, such as the Gauss-Newton or Levenberg-Marquardt algorithms.

Gauss-Newton Algorithm

For the Gauss-Newton algorithm, visualized in Figure 2.16, the iterative process can be outlined as follows:

1. Initialize the parameter vector, θ_0.
2. Compute the Jacobian matrix, J, of partial derivatives of the model function with respect to the parameters at the current iteration, t:

$$J_{ij} = \left. \frac{\partial f(x_i, \theta)}{\partial \theta_j} \right|_{\theta = \theta_t} \tag{2.26}$$

3. Update the parameter vector, θ_{t+1}, by solving the linear system:

$$J^T J \Delta\theta = -J^T r \tag{2.27}$$

, where r is the residual vector, $r_i = y_i - f(x_i, \theta_t)$, and $\Delta\theta$ is the change in the parameter vector.

4. Update the parameter vector:

$$\theta_{t+1} = \theta_t + \Delta\theta \tag{2.28}$$

5. Repeat steps 2-4 until convergence.

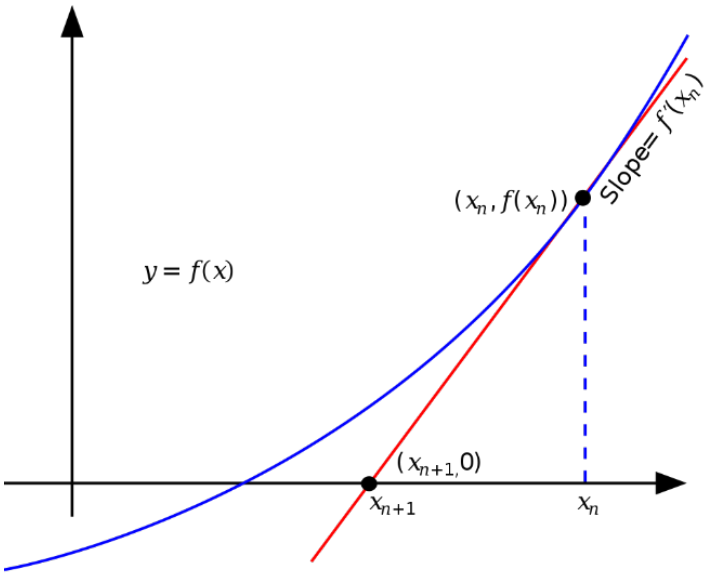

Figure 2.16: Visualization of Newtons method. [25]

Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm [26] is a modification of the Gauss-Newton method, introducing a damping factor to improve convergence in the presence of high curvatures:

$$(J^T J + \lambda I)\Delta\theta = -J^T r \tag{2.29}$$

, where, λ is the damping factor, and I is the identity matrix.

NLS for harmonic signals

To quantitatively determine the amplitude, frequency, and phase shift of a harmonic signal via the non-linear least squares (NLS) method, one would typically model the signal using a harmonic equation of the form:

$$y(t) = A\sin(2\pi ft + \varphi) + \epsilon \tag{2.30}$$

, where A represents the amplitude, f denotes the frequency, φ is the phase shift, t is time, and ϵ is a random error term that follows a normal distribution with zero means. Such a definition generally fits the problem defined in Section 5.1 between acceleration and gyroscope error.

The procedure for estimating these steps is taken:

Data Collection: Collect a time series data set $(t_1, y_{AccX1}, y_{GyroZ1}), (t_2, y_{AccX2}, y_{GyroZ2}), \ldots, (t_n, y_{AccXn}, y_{GyroZn})$ that represents the signal over a defined time interval.

Initial Guess: The process of parameter identification to fit data to a sinusoidal function is highly susceptible to the choice of initial guess parameters. The complex topography of the sine function contains a multitude of local minima, which poses challenges for optimization algorithms that may become trapped in these local solutions. One can think of a particularly illustrative example of this issue when the initial guess for frequency is set to zero. Provide an initial guess for the parameters A, f, and φ. This could be achieved via a Fourier transform method, or by using simpler techniques like identifying the peak-to-peak amplitude.

Cost Function: Define an objective function $J(A, \omega, \varphi)$ to quantify the difference between the observed data $y(t)$ and the model $y(t; A, \omega, \varphi)$:

$$J(A, \omega, \varphi) = \sum_{i=1}^{n} \left[y\left(t_i\right) - \hat{y}\left(t_i; A, f, \varphi\right) \right]^2 \tag{2.31}$$

Optimization: Minimize $J(A, f, \varphi)$ using a nonlinear optimization algorithm such as the Levenberg-Marquardt method, which is commonly used for nonlinear least squares problems. [27] [28]

Parameter Estimation: Extract the optimized values of A, f, φ that minimize J as the estimates for amplitude, frequency, and phase shift, respectively.

Model Validation: Validate the model by comparing the estimated signal with the observed data and compute relevant goodness-of-fit metrics like R^2 or the root-mean-square error (RMSE).

Residual Analysis: Examine the residuals $y(t) - y(t; A, f, \varphi)$ to ensure that they satisfy the assumptions of normality and constant variance, which validates the applicability of the NLLS method.

Summary

The choice between linear and non-linear least squares methods depends on the underlying relationship between the dependent and independent variables, and while linear least squares have an analytical solution, non-linear least squares typically require numerical methods to find the optimal parameters.

2.5.5 Artificial Neural Network

Elements of Artificial Neural Networks (ANN)

An Artificial Neural Network (ANN) is a computational model inspired by the way biological neural networks in the human brain work. A basic ANN model 2.18 is composed of three types of layers:

- Input Layer: It represents the input features.
- Hidden Layers: They perform computations and transformations of the inputs.
- Output Layer: It produces the final output.

Each layer is made up of nodes, visualized in Figure 2.17, or neurons, with each neuron in one layer connected to every neuron in the next layer. The connections between neurons are associated with weights, which are optimized during training. Such networks we call Feedforward Neural Networks (FNN).

The mathematical representation of a neuron is given by:

$$a_j^{(l)} = f\left(\sum_{i=0}^{n} w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right) \tag{2.32}$$

where $a_j^{(l)}$: Activation of the j^{th} neuron in the l^{th} layer, $w_{ji}^{(l)}$: Weight connecting the i^{th} neuron in the $(l-1)^{th}$ layer to the j^{th} neuron in the l^{th} layer, $b_j^{(l)}$: Bias term for the j^{th} neuron in the l^{th} layer, f: Activation function

The primary goal is to approximate some function [29]. The network is trained using an optimization algorithm, typically stochastic gradient descent (SGD) to minimize a loss function, which measures the difference between the network's prediction and the true target values.

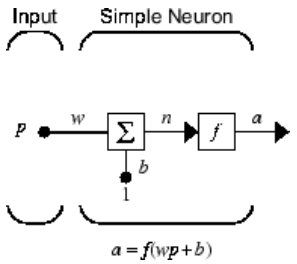

Figure 2.17: Schematic of artificial neuron. [30]

Such a network is just a static mapping input to output with no time dynamics, as we expect to be present in our system, due to phase delay shifting and inertias acting in our system. Such that this concept needs to be further extended.

Incorporation of Memory

A crucial limitation of FNN is their lack of memory or state, i.e., their inability to incorporate past information or sequence data, which is important for dynamic systems modeling. This restriction naturally leads to the exploration of models that can handle sequential data like time series, leading to the concept of Recurrent Neural Networks (RNN). RNN possess connections that form directed cycles, meaning information can be recycled in the network, which acts as a sort of memory. This enables RNN to maintain information about previous inputs in the sequence, addressing the limitations of FNN in handling sequential data.

Time Series Modeling

The inherent ability of RNNs to handle sequence data makes them suitable for time series modeling. However, in many real-world problems, time series data often depends not only on internal dynamics but also on external or exogenous inputs. This understanding leads to the development of models that can incorporate both types of inputs, setting the stage for models like NARX.

Nonlinear Autoregressive Exogenous Neural Network (NARX)

A NARX neural network is designed to make predictions based on the current and past values of the target variable (autoregressive inputs) and the current and past values of other external variables (exogenous inputs). Training a NARX network involves learning the parameters θ that minimize the difference between the predicted output and the true output for a given sequence of inputs and outputs.

Neural network training

In the context of neural networks, we talk about backpropagation. Backpropagation is realized in the following manner in the types of discussed AAN.

Backpropagation in Elementary Neural Networks

In elementary neural networks, backpropagation is employed to optimize the weights of the network. It works by calculating the gradient of the loss function with respect to each weight by applying the chain rule and propagating the gradient backward from the output layer to the input layer. The weights are then updated in the direction that minimally increases the loss function. In mathematical terms, for each weight w, the update rule is:

$$w = w - \eta \frac{\partial L}{\partial w} \tag{2.33}$$

, where η is the learning rate, and $\frac{\partial L}{\partial w}$ is the partial derivative of the loss function L with respect to the weight w.

Backpropagation in Recurrent Neural Networks

For Recurrent Neural Networks, backpropagation through time (BPTT) is used, which is an extension of the standard backpropagation algorithm. BPTT works by unfolding the RNN through time and then applying the standard backpropagation algorithm on the unfolded network.

Backpropagation in NARX Neural Networks

Backpropagation is also pivotal in training NARX networks. Like RNNs, NARX models are trained using a variant of backpropagation through time since they also have a kind of recurrent connection due to the inclusion of previous time step inputs and outputs. The error gradients are propagated back in time to update the weights and biases to minimize the prediction error at each time step in the sequence. The optimization in NARX using backpropagation ensures that the model learns the underlying patterns and dependencies in the input sequences and the relationship between past inputs, past outputs, and current outputs.

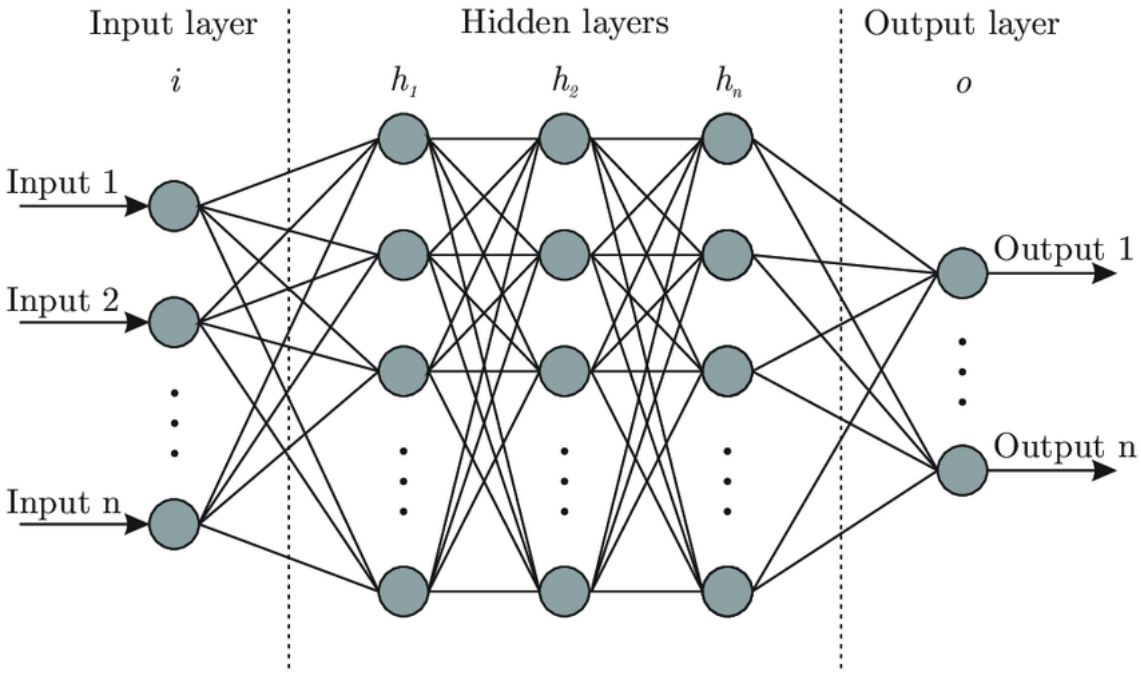

Figure 2.18: Schematic of artificial neuron network. [31]

Defining the neural network

The computational complexity inherent to any neural network architecture is a critical factor that directly impacts its feasibility for real-time microcontroller implementation.

It's worth emphasizing that the computational overhead associated with neural networks, particularly those of a nonlinear autoregressive nature, must be judiciously assessed in the context of resource-constrained embedded systems. Our ongoing ef-

forts are geared towards optimizing the network architecture and training algorithms to minimize computational requirements while preserving model accuracy and responsiveness. This approach aims to render the compensatory framework not only theoretically robust but also practically implementable in real-time microcontroller environments. It is imperative to emphasize that the complexity of a neural network ought to be precisely characterized and rigorously assessed in the context of the specific problem it is designed to address. Prior to the development or deployment of the network, several critical factors warrant preemptive consideration.

Number of Layers and Nodes

- Input Layer: The number of nodes in the input layer is generally determined by the dimensionality of the data. For example, if you're working with 28x28 pixel grayscale images, you'd typically have 28 * 28 = 784 input nodes.
- Output Layer: The number of nodes in the output layer is determined by the number of classes for classification problems or the dimensionality of the output for regression problems.
- Hidden Layers: The number of hidden layers and nodes is highly problem-specific. A good starting point might be one hidden layer with several nodes that is a mean of the input and output layer sizes, but this is a very rough heuristic. Generally, more layers and nodes allow the network to learn more complex representations but also make it prone to overfitting.

Factors to Consider

- Overfitting and Underfitting: A small network might not capture the complexity of the data (underfitting), while a very large network might learn the training data too well, including its noise (overfitting).
- Computational Constraints: Larger networks require more computational resources both for training and inference.
- Data Availability: A larger dataset can justify a larger network. However, if you're limited by data, it might be more effective to use a smaller network or data augmentation techniques.
- Problem Complexity: More complex problems may require larger networks. Sometimes domain knowledge can be useful in determining an appropriate network size.

Methods to Find the Right Size

- Grid Search: This involves training multiple networks with different architectures and selecting the one that performs the best on a validation set.
- Random Search: Randomly select architectures within specified limits and train each, selecting the best-performing one.

- Early Stopping: Start with a smaller network and gradually make it larger until the performance on a validation set starts to decrease.
- Pruning: Start with a larger network and remove nodes or layers whose removal does not significantly impact performance.
- Automated Methods: There are automated methods like neural architecture search (NAS) that automatically search for the best-performing architecture, although these are often computationally expensive.
- Expert Advice: Sometimes consulting the scientific literature or experts in your specific application domain can provide valuable insights.

2.6 High-Level code synthesis

High-level code synthesis, or automatic code generation, is an emerging field aimed at the automated generation of source code from a higher-level specification, be it natural language, formal logic, or a graphical representation. As software systems grow increasingly complex, the potential for automation in code generation becomes an invaluable asset. This chapter describes the various paradigms, technologies, and applications that utilize high-level code synthesis.

In the landscape of software engineering and computer science, few ideas have been as transformative as automation. Automation in code development can dramatically reduce the time-to-develop and improve software quality by minimizing human errors. High-level code synthesis offers a path towards this automation, providing methods and tools for generating optimized, production-quality code from high-level specifications. The notion of automating code generation isn't new, but also isn't generally applied in the industry. Early work in compilers, interpreters, and domain-specific languages (DSLs) presented the possibility of translating high-level syntax to machine code. However, high-level code synthesis differs in its aim to generate human-readable, maintainable code that is often indistinguishable from that written by expert programmers.

High-Level Synthesis (HLS) for Field-Programmable Gate Arrays (FPGA) has emerged as a pivotal approach that significantly streamlines the hardware design cycle, removes specific syntax barriers, and optimizes resource utilization. In the traditional design flow for FPGAs, engineers would manually design hardware at the Register-Transfer Level (RTL) using Hardware Description Languages (HDL) like VHDL or Verilog, which requires a deep understanding of both the architecture of the FPGA and the intricacies of parallel computation. However, HLS tools allow for the abstraction of these complex processes by enabling programmers to write algorithms in high-level programming languages like C, C++, or theoretically any other proprietary language. These high-level representations are then automatically synthesized into optimized RTL descriptions that can be directly mapped onto FPGA hardware. In effect, HLS enables software engineers to engage in FPGA design without having to delve into the low-level hardware details, thus blurring the

boundaries between software and hardware engineering.

The technology leverages advanced techniques such as loop unrolling, pipelining, and dataflow optimization to automatically generate code that is highly optimized for latency, throughput, and power consumption. Moreover, HLS tools often come with integrated simulation and verification functionalities, thus aiding in the rapid prototyping and iterative development of complex systems. While the benefits are numerous, there are also challenges like the difficulty in estimating performance without full hardware emulation. Nevertheless, HLS is increasingly being adopted in various applications ranging from embedded systems and digital signal processing to high-performance computing, making it a key technology in the evolving landscape of hardware-accelerated computing.

2.6.1 Mathworks Matlab & Simulink

Matlab, a versatile engineering tool, offers a comprehensive suite of toolboxes designed to address a wide array of technical challenges. One of its key capabilities is the streamlined generation of code suitable for execution on embedded devices like microcontrollers (μC) Field-Programmable Gate Arrays (FPGAs). Through the use of various specialized toolboxes—such as Simulink Coder, Matlab Coder, Embedded Coder, or HDL Coder—users are afforded the flexibility to generate either C or HDL code compatible with a range of platforms, without being restricted to a particular vendor. However, leveraging this flexibility does require a foundational understanding of each level of code processing involved. Typical workflow visualized in Figure 2.19. One of the major advantages of utilizing Matlab for code generation lies in its vast repository of predefined functions. These functions can be directly exported from Matlab to various embedded devices, making the creation of technically advanced systems significantly more straightforward. Companies affiliated with Matlab, such as dSpace and SpeedGoat, offer optimized solutions for hardware-level code generation. These solutions are tailored to meet the specific needs of various industrial segments, thereby catering to a diverse customer base.

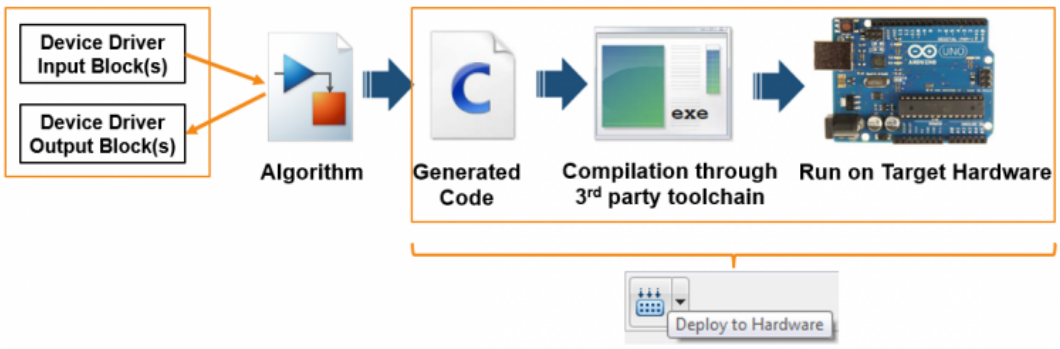

Figure 2.19: Matlab, Simulink workflow. [32]

In the context of our research, we focused on the application of high-level code synthesis using Matlab for the generation of code intended for PIC microcontrollers, an example of Simulink content for such a project is in Figure 2.20. In this scenario, the intermediate code was written in C, and the final assembly code was synthesized through API calls to Microchip's MPLab environment. This approach demonstrates Matlab's versatility and efficacy in generating optimized, platform-independent code for embedded systems. In our scope, we use Matlab for data processing, algorithm development, and deployment to our own custom HW.

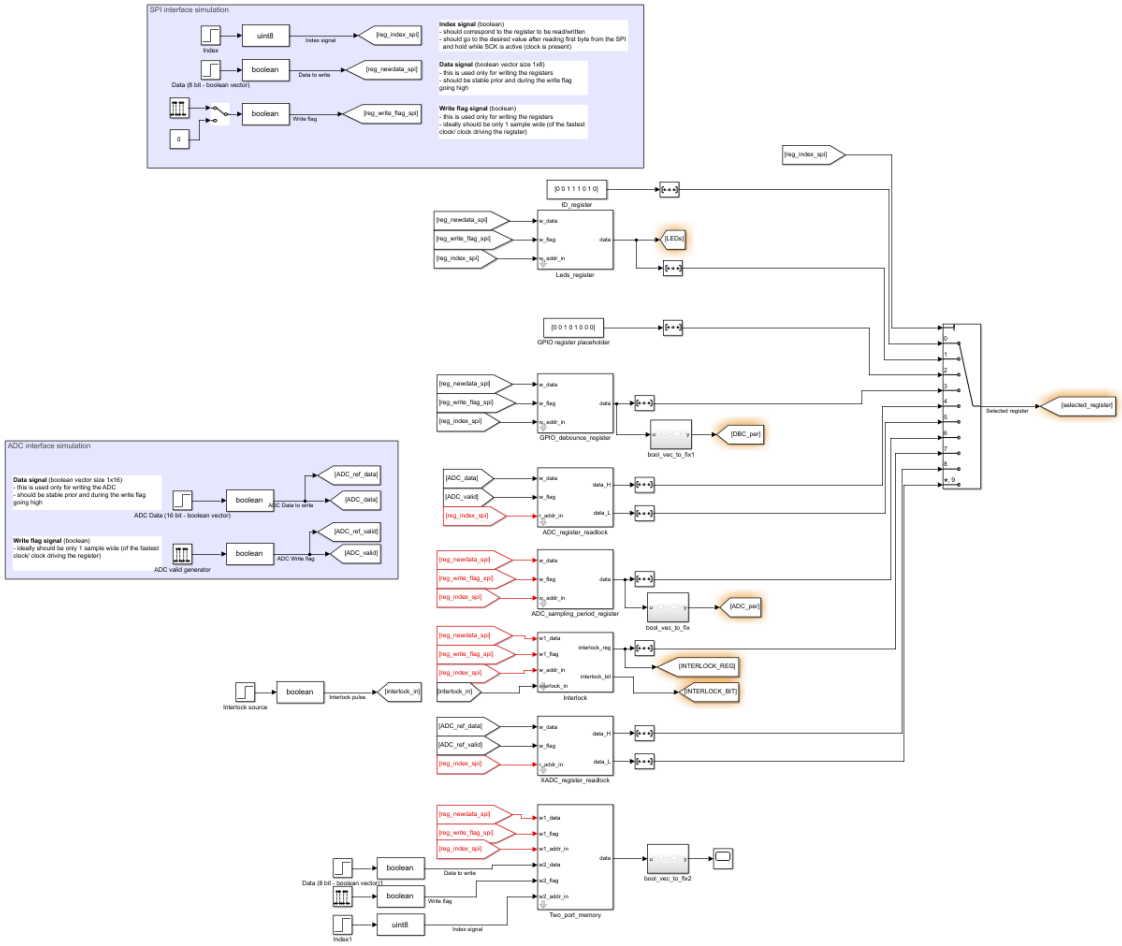

Figure 2.20: Matlab, Simulink example of code for embedded application.

2.6.2 LabVIEW

LabVIEW offers an integrated ecosystem of hardware and software tools, creating a seamless user experience that enables users to concentrate on the core problem at hand without unnecessary subsystem knowledge. While this integration promotes ease of use, it imposes limitations on the range of applications and locks users to a proprietary technology stack, restricting the ability to integrate these solutions into diverse systems.

LabVIEW employs its specialized G programming language, visualized in Figure 2.21, a graphical language founded on a dataflow paradigm, contrasting it with traditional text-based languages such as C and Python. The G language embodies the dataflow mechanism by executing nodes as soon as all requisite input data are available and subsequently generating output data for downstream nodes. This architecture not only facilitates inherent parallelism but also endorses modular design through Virtual Instruments (VIs). VIs act as modular building blocks within LabVIEW applications and can encapsulate other VIs, thereby offering multiple layers of abstraction and reusability. Every VI comes with a user-friendly front panel that serves as the Graphical User Interface (GUI), as well as a block diagram that houses the G code logic. Designed to interoperate easily with a diverse array of hardware from different vendors—including data acquisition devices, sensors, and actuators—LabVIEW and its G programming language are explicitly crafted for engineering tasks such as data acquisition, control systems, and automation. Regarding its application in Field Programmable Gate Array (FPGA) programming, the G language's dataflow paradigm mirrors the hardware description language (HDL) code, making it exceptionally well-suited for this purpose. However, LabVIEW typically outputs bitfiles that are exclusively compatible with National Instruments (NI) hardware. Consequently, the platform's closed architecture limits portability and restricts users from exporting the generated HDL code to hardware from different vendors. This limitation also constrains the ability to reconfigure the pin assignments for specific FPGAs. In our research endeavors, we leverage NI's hardware infrastructure for real-time data acquisition and processing, benefitting from the cohesiveness and specialized features of the LabVIEW environment, while acknowledging its inherent limitations in terms of hardware portability and application scope.

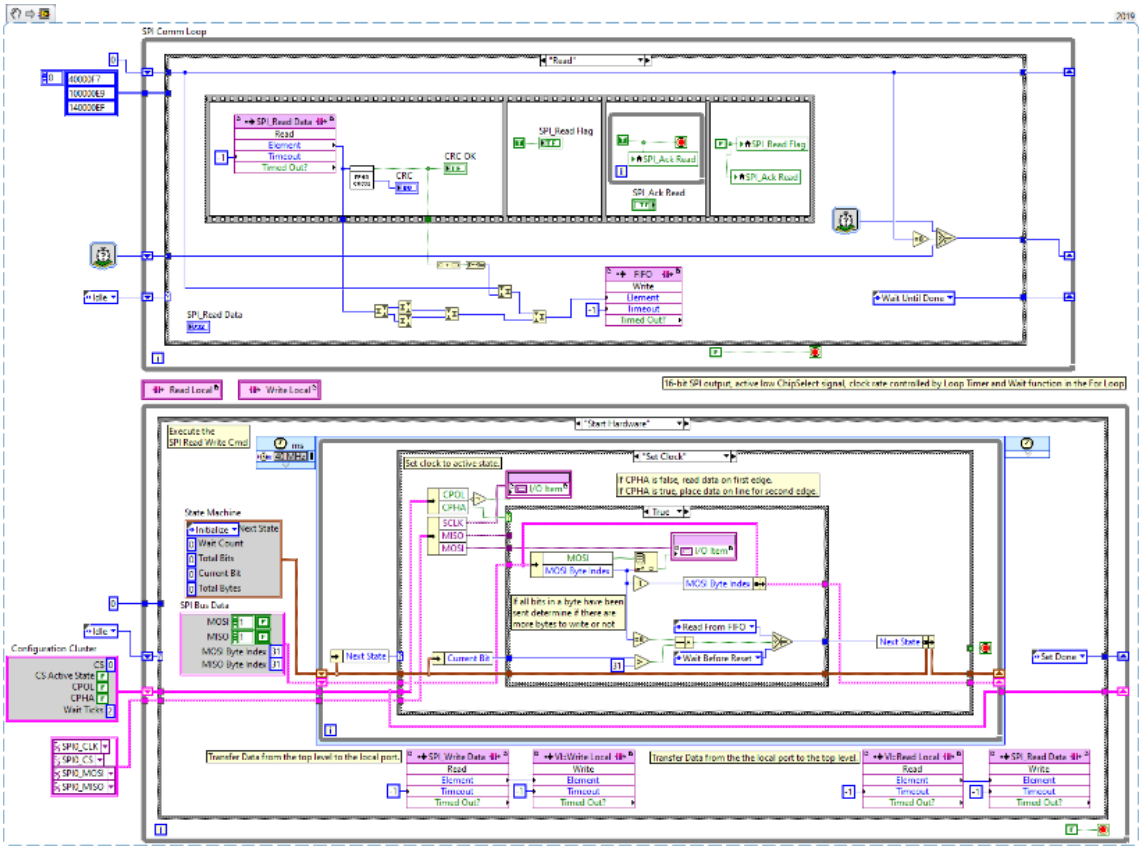

Figure 2.21: LabView example of code for embedded application.

3 Formulation of the thesis goals

Currently, there are many fusion algorithms for estimating IMU rotation in space, but very little attention has been paid to parasitic effects acting on MEMS gyroscopes, which are an elementary part of the IMU. According to our study so far, we conclude that the linear acceleration, whose compensation is not entirely trivial and is often neglected, has a major influence on the measurements. This thesis will deal with the development of a new compensation algorithm that will be able to reduce the influence of linear acceleration on a MEMS gyroscopic sensor in real-time. We assume that this will subsequently lead to a more accurate fusion algorithm for estimating the rotation of the IMU unit in space. The individual objectives can be characterized as follows:

3.1 Theoretical objective 1: Analysis of the effect of linear acceleration and jerk on MEMS gyroscopic sensors and their quantification

The effect of linear acceleration and jerk on MEMS gyroscopic sensors is a marginally investigated topic, but one that significantly affects sensor performance. Sensor manufacturers typically report sensitivity to constant linear acceleration, such as gravitational acceleration, in the form of a constant, which does not adequately capture the problem due to the complexity of the problem. Manufacturers very rarely report sensitivity to dynamic disturbance. The first objective of this dissertation will result in the definition of a unified procedure for quantifying the effect of linear acceleration and jerk on MEMS gyroscopic sensors. We will experimentally quantify the effect of linear acceleration and jerk on real sensors of different price categories.

3.2 Theoretical objective 2: Design of new models for linear acceleration compensation

A method will be proposed to compensate for linear acceleration using acceleration measurements and further processing. We assume that the compensation method will be generally nonlinear and the models used will be based on ANN, local linear models, or polynomial models.

If we compensate for the effect of linear acceleration before entering the fusion

algorithm to estimate the rotation of the IMU unit in space, we will be able to estimate the rotation angle more accurately, or cheaper sensors can be used while maintaining the performance. Graphically, the resulting structure of the compensation algorithm can be defined as shown in Figure 3.1. The results from published papers [33] or [11] propose that the sensitivity to linear acceleration can be reduced considerably.

The output will be a mutual comparison of the robustness of several variants of our proposed approximators through simulations and experiments. We anticipate that the RCP approach on the dSPace HW platform or NI PXI will be used for experimental testing of the algorithms.

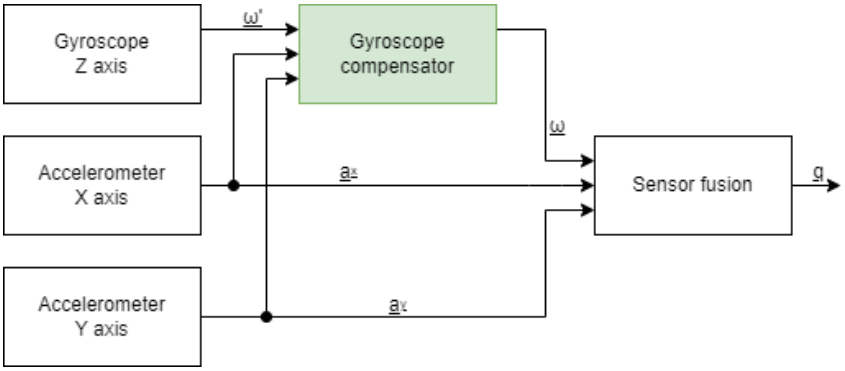

Figure 3.1: Compensator principle.

3.3 Practical goal: Implementation of the proposed method on RT-HW and experimental measurement

Based on the comparison of different algorithms in the previous objective, we will select a suitable compensation algorithm and implement it on a prototype Real-Time HW. It is suggested to use a microcontroller or an FPGA. We will evaluate the practicality of implementing the solution and determine the required HW. According to the resulting behavior in a real laboratory model, the contribution of the implemented compensator model will be quantified. A prototype control unit will be used, containing MEMS gyroscopes, accelerometers (IMUs), and a suitable processor, enabling signal processing by the newly proposed methods in real-time. In addition, a mechatronics measurement system will be constructed for measurement purposes and experiments will be performed on it. This system can then be used as a laboratory model for teaching intelligent machine control.

4 Analysis of gyroscope measurement errors

An introduction to the relevant physical environment is provided in this chapter followed by the measurements and observations realized to support our further work.

Motivated by our initial system of interest described in the Chapter 2 we constrained the frequency range under examination to a maximum of 10 Hz, a decision based on our observations that this range was most frequently relevant to the system's performance. The subsequent chapters provide a standardized template for determining sensor sensitivity to linear acceleration, in support of theoretical objective 1.

4.1 Physical setup of the system

Since our work involves dealing with actual hardware, it was crucial to develop an appropriate testing environment and choose the right sensors for evaluation, with further details provided in this section.

4.1.1 Sensors selection and relative environment

To rigorously ascertain the influential parameters within our domain of study, it is necessary to design the experimental framework. This ensures the minimization or elimination of confounding variables, such as parasitic properties that are susceptible to environmental conditions and can unintentionally skew gyroscope measurements. Notably, one such parasitic property is temperature drift [34], characterized by a relatively long thermal time constant. We have mitigated this by taking short measurements in constant environmental conditions. Also to ensure the stable supply voltage the sensor modules were powered by a lithium-ion battery, which was regulated with a low drop-out (LDO) voltage regulator. This architecture was designed to provide a long-term, stable power supply, thereby minimizing fluctuations that could adversely affect sensor readings.

In the interest of enhancing measurement accuracy, our experimental design incorporated a dual-measurement approach, wherein each test case was simultaneously run with its inverted, complementary sensor. The inherent advantage of this paired-measurement methodology lies in its potential for nullifying random measurement errors. Under the assumption that the sum of the signal values read out from the

paired measurements should ideally result in a zero mean, the data would display minimized variance. Moreover, this technique has the added benefit of mitigating false negative errors, thereby reinforcing the deterministic attributes of the system's behavior. By employing this multifaceted, yet rigorously controlled experimental approach, we aimed to establish a methodological framework that balances both the operational robustness and the statistical validity of our findings.

The subsequent sensors were chosen for evaluation.

InvenSense MPU-6050

The InvenSense MPU-6050 sensor [35], in Figure 4.1 is a distinguished MEMS sensor, integrating a 3-axis gyroscope and a 3-axis accelerometer. The gyroscope in the sensor has selectable range of ±250, ±500, ±1000, and ±2000 deg/s, and the accelerometer offers selectable scales of ±2, ±4, ±8, and ±16 g. Moreover, the sensor can deliver output data at a high rate, up to 8 kHz, ensuring that users can access real-time motion data. It also houses an embedded Digital Motion Processor (DMP), providing an additional layer of flexibility and capability for users. With these advanced features, the sensor maintains low noise levels, with the gyroscope and accelerometer having the noise of 0.05 $deg/s/\sqrt{Hz}$ and 400 $\mu g/\sqrt{Hz}$, respectively. All these capabilities are balanced with efficient power management, operating normally at 3.9 mA. Specific gyroscope setting in table 4.1.

parameter	Gyroscope range	Acceleration range	Read-out frequency
value	±250 deg/s	±2 g	800 Hz

Table 4.1: InvenSense MPU-6050 custom settings.

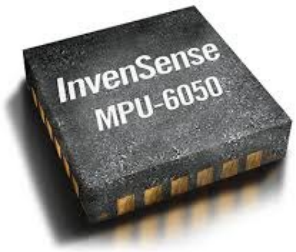

Figure 4.1: InvenSense MPU-6050. [35]

ST L3G4200D

The STMicroelectronics L3G4200D sensor [36], in Figure 4.2 is a high-precision MEMS 3-axis gyroscopic device. It offers selectable full scales of ±250, ±500, and ±2000 deg/s and can output data at rates of 100 or 800 Hz. The sensor boasts a low noise level, with a rate noise density of 0.03 $deg/s/\sqrt{Hz}$, ensuring accurate angular rate measurements. It operates within a supply voltage range of 2.4 V to 3.6 V and exhibits energy-efficient properties, with a typical operating current of

6.1 mA. Additionally, the incorporation of both I2C and SPI interfaces facilitates versatile connections and integrations with a wide range of devices and platforms. Specific gyroscope setting in table 4.2.

parameter	Gyroscope range	Read-out frequency
value	$\pm 250\ deg/s$	$800\ Hz$

Table 4.2: ST L3G4200D custom settings.

Figure 4.2: STMicroelectronics L3G4200D. [36]

Murata SCC2000

The Murata SCC2000 series sensor [37], in Figure 4.3 is a high-performing and robust device that combines gyroscope and accelerometer functionalities, operating within gyroscopic measurement ranges of ± 75, ± 150, and $\pm 300\ deg/s$ and accelerometer ranges of ± 2 and $\pm 6\ g$. It is distinguished by its exceptionally low noise, at 0.006 $deg/s/\sqrt{Hz}$ for the gyroscope and 37 $\mu g/\sqrt{Hz}$ for the accelerometer, and its remarkable shock tolerance of $10,000\ g$, making it particularly suitable for demanding automotive and industrial applications. It provides reliable performance over a wide operating temperature range of $233\ K$ to $400\ K$ and offers ease of integration with various systems through its digital SPI interface output, all while operating at a single supply voltage ranging from $4.75\ V$ to $5.25\ V$. Specific gyroscope setting in table 4.3.

parameter	Gyroscope range	Acceleration range	Read-out frequency
value	$\pm 75 deg/s$	$\pm 2g$	$800 Hz$

Table 4.3: Murata SCC2000 custom settings. [37]

4.1.2 Design of the test bench

As we research motion sensing devices, we need to consider the right design of test bench capable of a high level of fidelity in dynamic aspects and reproducible results

Figure 4.3: Murata SCC2000. [37]

without many uncontrolled variables. What properties it should reflect and which not was solely based on our expert knowledge of electromechanical systems. Another critical determinant that influenced the mechanical design of our measurement setup was the jerk, the rate of acceleration change. Our objective was to achieve a jerk waveform that was continuous and devoid of any saturation effects, as observed from the sensor's output data.

To systematically mitigate the undesirable impulsive mechanical behavior, we considered the motor, a key component in the whole design. We consider the following aspects in the design of test bench:

- **Enhancement of Mechanical Inertia**

  An approach is to increase the mechanical inertia or weight of the moving components within the system. While this strategy preserves the system's original dynamic characteristics, it introduces new complexities. Specifically, increasing mechanical inertia imposes higher requirements on the structural rigidity of the test bench, as well as necessitating more robust mounting solutions to ensure stability during operation.

- **Type of motor**

  Usage of electrically commutated motors with sinusoidal commutation: brushless DC (BLDC) motors or permanent magnet synchronous motors (PMSM) motors with sinusoidal commutation. This alternative offers advantages in terms of a potentially smoother torque profile. However, it may necessitate complex electronic control schemes, and often, the motor drivers come at a higher initial cost. DC Motors on the other hand can feature an increased number of commutation segments, coupled with a current feedback control mechanism that provides a robust stable system. By maximizing the commutation points, the motor can achieve finer torque generation and distribution. The feedback loop, often implemented via PID (Proportional-Integral-Derivative) controllers, allows for real-time adjustments to the motor current.

The employment of the Maxon RE35 DC Motor, featuring 13 commutation segments, was a strategic decision aimed at reducing the confounding effects of non-uniform torque distribution during a turn, as visible in Figure 4.4. This selection not only improved torque consistency but also upheld the merits of a simplistic yet robust system architecture.

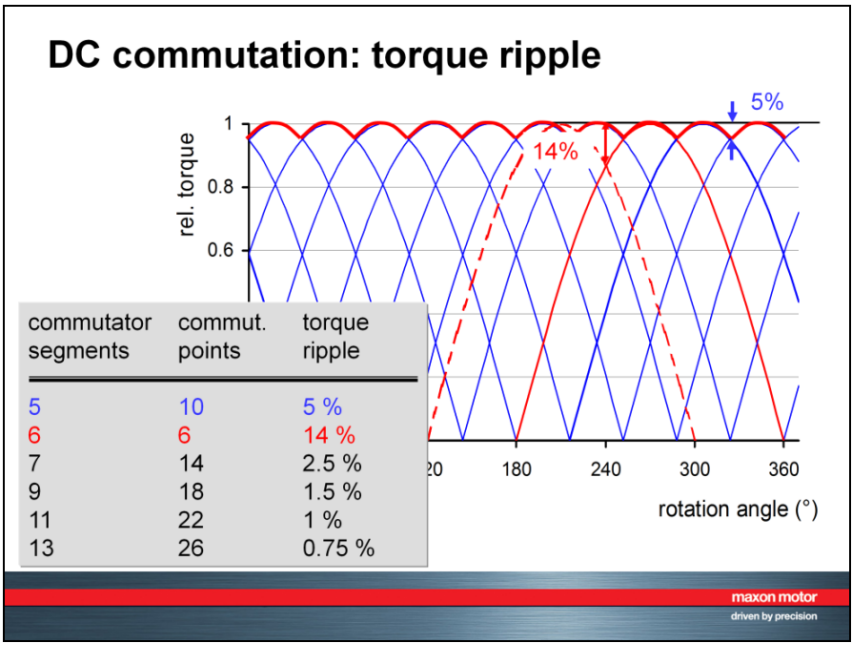

Figure 4.4: Current ripple for different numbers of DC motor commutator contacts. [38]

In scenarios requiring only smooth motion, we employed a straight linear guidance system to facilitate the movement. The cart is outfitted with sensor arrays. This approach proved to be highly effective; it allowed for the rapid execution of a broad spectrum of experiments, while simultaneously ensuring a smooth motion profile. Moreover, the system provided a frequency range adequately reflecting the conditions relevant to the target application.

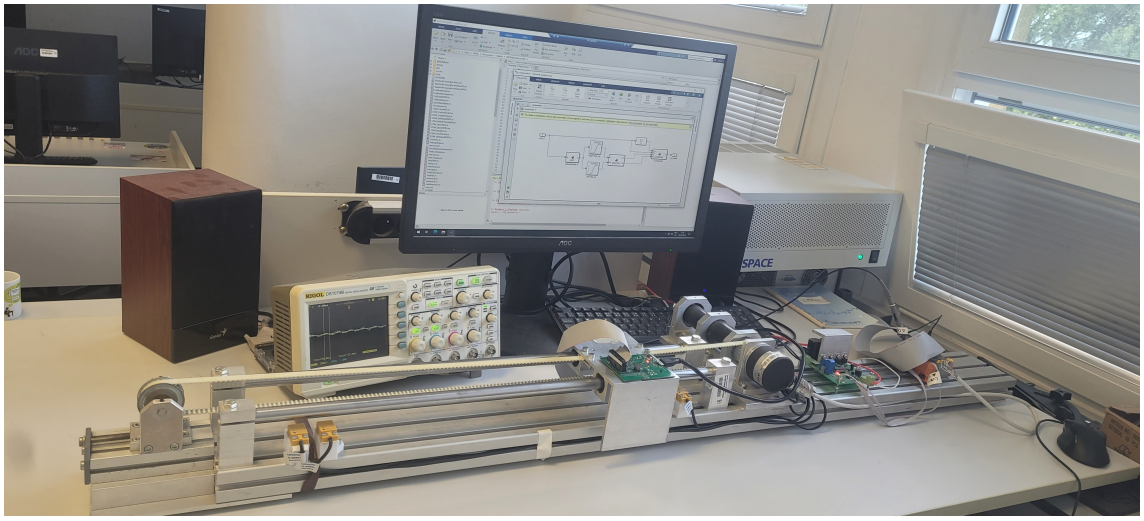

Figure 4.5: Linear movement test-bench.

To guarantee experimental robustness, particularly concerning the fidelity of excitation frequencies and acceleration amplitudes, we custom-engineered a dedicated

test bench. This was pivotal for the introduction of precise linear acceleration stimuli to the gyroscopic sensors under scrutiny. To this end, we employed industrial-grade linear guides furnished with 12mm-diameter guide rods and linear ball bearings. These components were affixed to a robust aluminum industrial modular profile, ensuring both mechanical integrity and mitigation of undesired structural deformations or spurious accelerations along unintended axes. For actuation, we opted for a Maxon RE35 motor in tandem with the Maxon ESCON 70/20 industrial module serving as the motor control unit.

In addressing the need for deterministic measurements, we incorporated a National Instruments sbRIO-9632 development card equipped with a Xilinx Spartan 3 FPGA. This configuration was selected based on its capacity for high-frequency analog current readings (up to 50 kHz) and minimal temporal jitter. Programming for this hardware was exclusively conducted via the LabView environment, utilizing both the real-time and FPGA toolbox modules. The real-time toolbox facilitated data acquisition directly onto the FLASH disk of the development kit, while the FPGA toolbox was employed for programming critical functionalities, which include:

- Real-time analog current monitoring for the motor.
- Encoder enumeration for positional feedback interlocks.
- Implementation of a feedback loop for sinusoidal current control, enabling variable frequency and amplitude modulation per experimental needs.
- SPI (Serial Peripheral Interface) communication.
- I2C (Inter-Integrated Circuit) communication.
- In-FPGA RAM data storage.
- DMA (Direct Memory Access) to the coprocessor located on the sbRIO board, facilitating data offloading into FLASH memory for subsequent analytical processing.

Importantly, both the motor control and gyroscopic sensor readout tasks were executed in a synchronized and deterministic manner. This was made possible by consolidating these tasks on the Xilinx Spartan 3 FPGA that is integral to the NI sbRIO-9632 unit.

Regarding the motor control strategy, we implemented direct position control with a sinusoidal waveform, predicated on the simple following assumptions:

$$\overline{a} = \frac{d^2\overline{x}}{dt^2} \tag{4.1}$$

This experimental methodology was designed to control key variables, thereby substantively fortifying both the reproducibility and reliability of the collected data.

However, real-world complexities such as substantial frictional forces within the mechanical setup introduced perturbations. These perturbations led to an inherently non-sinusoidal and smooth force distribution exerted on the gyroscopic sensors under evaluation and so the current and related acceleration. Consequently, our control strategy pivoted from a comprehensive position-based loop to only a sinusoidal current regulation. While this modified control paradigm rendered an exemplary force waveform, it concurrently relinquished positional control, resulting in an operationally ambiguous range. Subsequent iterative empirical adjustments facilitated a mechanical configuration that precluded any mechanical collisions throughout the data acquisition phase. Lastly, we were able to maintain a stable position-bounded behavior with smooth sinusoidal moment generation as can be seen in Figure 4.6 on the measured position and actual motor current.

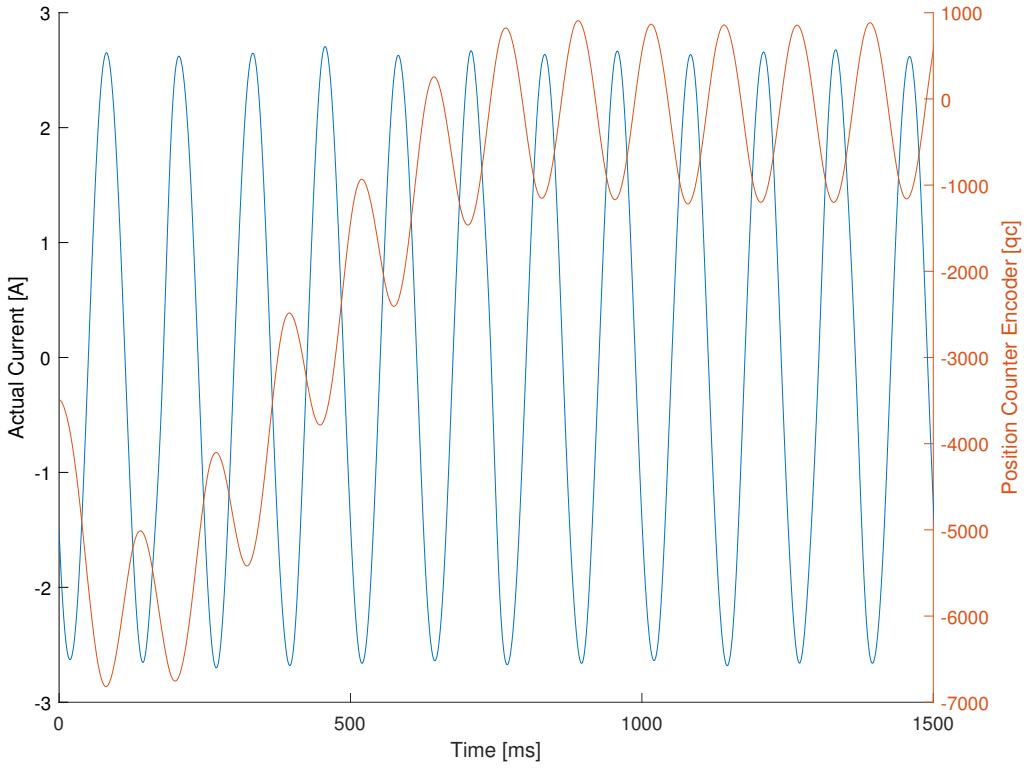

Figure 4.6: Motor current and position.

To assess the sensitivity of the gyroscopic sensor to linear accelerations across a frequency spectrum ranging from single to ten of Hz, we employed a dedicated measurement approach, potentially extendable to user-defined boundaries and frequency resolution. Acceleration metrics were directly gauged using a triaxial accelerometer, which was synchronized with the gyroscopic output data. To mitigate off-axis accelerations that could compromise measurement fidelity, all sensors were rigidly affixed to the linear motion carriage utilizing both adhesive and mechanical fasteners. Prior experiments have indicated that mechanical compliance can introduce considerable bias into sensor readings, and thus, this approach was instrumental in circumventing

such confounders. Moreover, the substantial mass of the base ensured the overall mechanical stability of the linear guide system.

To further reinforce the robustness of our measurements, gyroscopic sensors were always engaged such that their axes were intentionally aligned in the same direction. Drawing from previous research insights, we expected a one-to-one correspondence in the readings from these sensors. This dual-sensor strategy allowed for rigorous verification of the determinism underlying the observed phenomena, effectively mitigating concerns that the recorded data might be corrupted by stochastic interference or random noise in individual gyroscopes.

4.2 Observation A: cross-correlation between same gyroscopic sensors, random excitation

4.2.1 Hypothesis

In formal terms, the hypothesis posits that variations in linear acceleration and jerk exert a direct and quantifiable impact on gyroscopic sensor readings. Furthermore, these influences manifest in a deterministic manner, meaning that under identical experimental conditions, the effects on the gyro measurements will consistently reproduce the same outcomes across different temporal instances.

- **Objective**

  Quantify the extent to which linear acceleration and jerk impact gyroscopic measurements.

- **Significance**

  Understanding this feature will help us to determine the applicable area for the potential compensators and define the scope of further research.

- **Methodology**

  The hypothesis will be tested through a series of controlled experiments designed to isolate and manipulate linear acceleration and angular velocity parameters. Data will be collected over multiple runs to assess both the deterministic behavior and repeatability of the observed effects.

4.2.2 Experiment

By mounting two discrete InvenSense MPU-6050 gyroscopic sensors on a single linear guide, we ensured unidirectional rigid motion along a single axis, thereby effectively mitigating the risk of unintended rotational artifacts and auxiliary motions along the perpendicular axis during the experiment. In our examination of the test sample, we pinpointed linear dynamic acceleration and jerk excitations as the predominant error inducers in the gyroscope's sensing axis—the axis designated for Coriolis force measurements. In this chapter, we focus on the influence of acceleration solely on the sense axis.

Regarding the excitation mechanism for the linear guidance system and the gyroscopic sensor pair, a stochastic random approach was adopted in this particular set of experiments. Specifically, manual hand movements were employed to induce motion in the carriage system. Within this context, the gyroscopic sensors were meticulously positioned such that the positive axes corresponding to sensing electrodes were inverted to each other, see Figure 4.7.

This experiment's design was purposefully tailored to facilitate a granular understanding of the variables contributing to measurement errors, and in doing so, it has revealed potential direction for future investigations aimed at sensor optimization and error minimization.

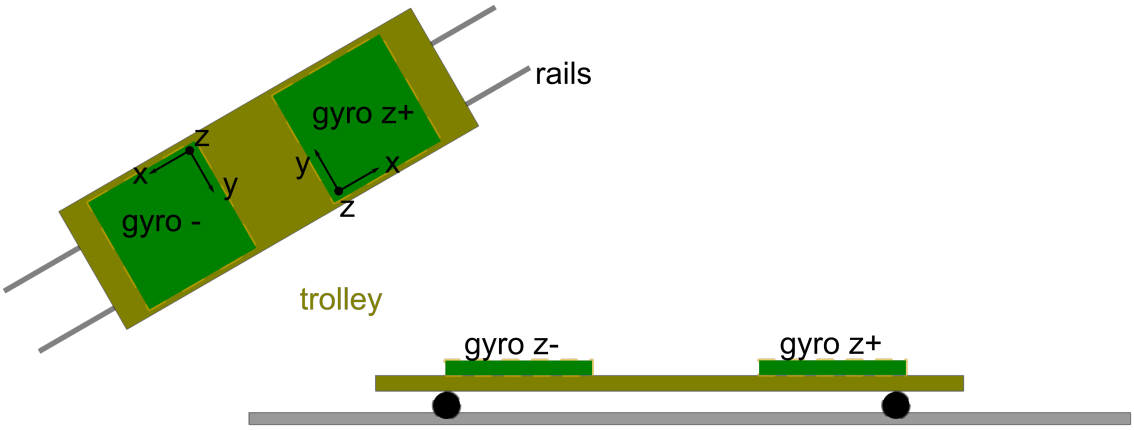

Figure 4.7: Schema of experiment.

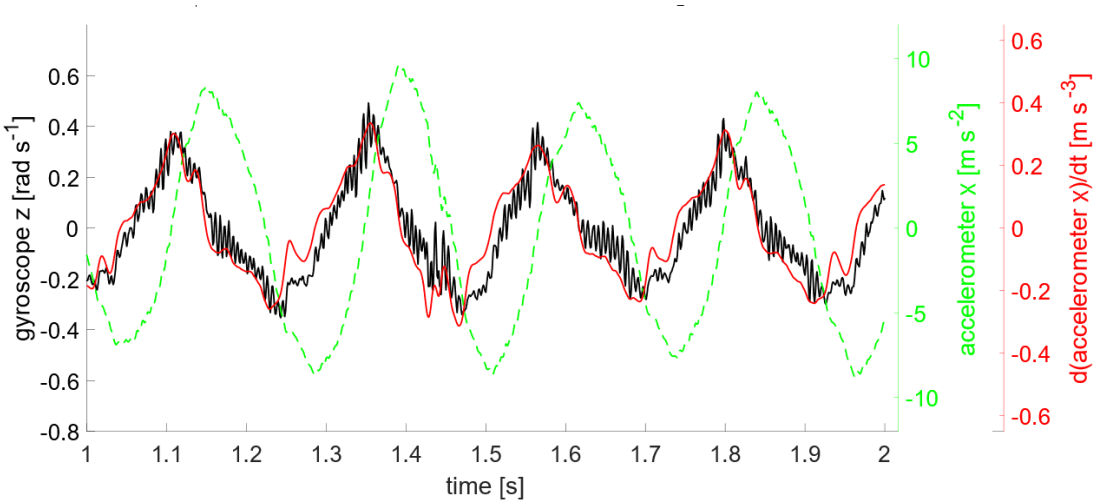

Figure 4.8: Acceleration, jerk, and gyroscope output in the time domain. [11]

The presence of measurement error under conditions of a pseudo constant disturbance frequency is observed to propagate with a phase shift that is directly relative to the correspondingly applied acceleration, as visible in Figure 4.8. This implies

that the resultant error can be characterized as a function of both the applied acceleration and its rate of change – formally jerk, its derivative. In the context of a gyroscope, these effects are particularly salient on the oscillating mass within the sensor. The ratio of these influencing factors provides a quantifiable metric that can define the extent of the phase shift. Comparing the two inverted sensors demonstrates a deterministic effect of dynamic acceleration disturbances on the gyroscope's sensing axis. Refer to Figure 4.9 for a time-domain interpretation and Figure 4.10 for signal cross-correlation.

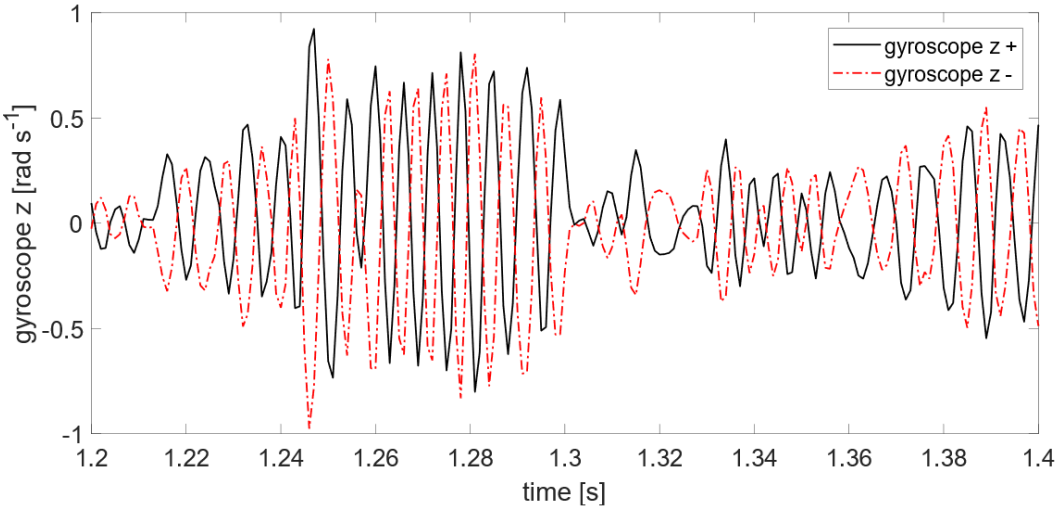

Figure 4.9: Measurement of mechanically coupled gyroscopes. [11]

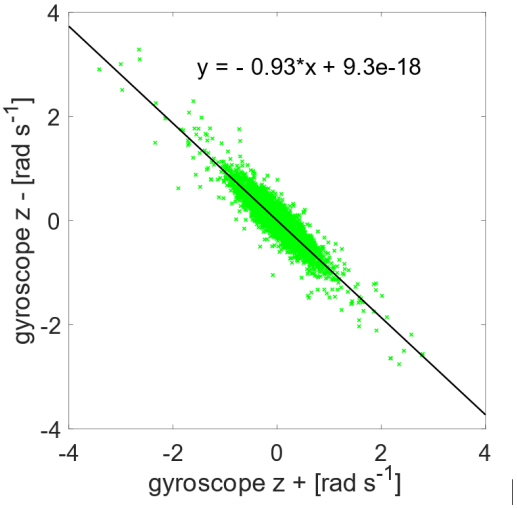

Figure 4.10: Cross-correlation of two identical gyroscopes at zero rotation under dynamic disturbances. [11]

4.2.3 Summary

The observed linear correlation between two gyroscopes that are identical in specification but opposite in orientation lends strong empirical support for the deterministic nature of the phenomenon under investigation. Drawing from the data presented in Figures 4.9 and 4.10, we can robustly validate our initial hypothesis: namely, that mechanical acceleration and corresponding jerk exert a direct, quantifiable impact on the measurements obtained from the gyroscopes sense axis.

This empirical evidence not only corroborates the deterministic characteristics of the sensor behavior under varying mechanical conditions but also underscores the reliability and repeatability of these measurements. Consequently, this finding represents a significant advancement in our understanding of gyroscope sensor dynamics and offers a validated framework for future research aimed at minimizing sensor errors or optimizing performance metrics.

4.3 Observation B: cross-correlation between different gyroscopic sensors, random excitation

The aim is to investigate the effect of linear acceleration and jerk on MEMS gyroscopic sensors and their effect on the drive and sense axis. Multiple manufacturers are selected to mitigate the risk of specimen-specific behavior.

4.3.1 Hypothesis

The influence of linear acceleration and jerk on measurement outcome is an intrinsic characteristic unique to MEMS gyroscopic sensors.

- **Objective**

  Empirically analyze how linear acceleration in each orthogonal direction affects the measured output of various MEMS gyroscopic sensors.

- **Significance**

  Understanding this feature could lead to improvements in MEMS gyroscopic sensor accuracy which could lead to more robust error-correction algorithms for these sensors.

- **Methodology**

  Collect data using multiple MEMS gyroscopic sensors under varying conditions of linear acceleration and jerk.

4.3.2 Experiment

In the forthcoming experimental investigation, our primary objective is to mitigate the confounding influence attributed to the structural design of specific gyroscopic

sensors or manufacturing batch. To address this concern, we have selected an alternative sensor from a different manufacturer based on the same measurement principle of angular rotation.

Specifically, the gyroscopic sensors examined in this section encompass:

- InvenSense MPU-6050
- ST L3G4200D
- Murata SCC2000

During the experimental phase, a nuanced observation was made. Accelerations in orthogonal axes, that are the drive axis and the sense axis, exert divergent influences on the gyroscopic measurements. Consequently, our initial task was to address the spatial orientations of these sensor axes of selected gyroscopes. The primary metric for evaluation in this context was variance in axis-specific sensitivities.

This experimental protocol builds upon previous research findings [11], which have demonstrated that dynamic excitation along the sensing axis contributes to gyroscopic measurements at a magnitude approximately five times greater than that along the drive axis.

For the two selected sensors, we notice that the propagation of dynamic disturbances in the sensing direction of the gyroscope to its output is almost identical, based on the evaluation of linear fit R^2 between both gyroscopes. Comparing the Zero Rate Outputs (ZRO) of both gyroscopes in a single graph, we can get meaningful insights into the similarities in the performance of both sensors under identical conditions, see Figure 4.11. The closer the value of R^2 is to 1, the more deterministic the response of the gyroscopes to identical input. It is evident that, in the first case as referenced in Figure 4.12, the linear fit R^2 is 0.95, revealing the same behavior of both sensors.

When dynamic disturbance was introduced in the orthogonal direction, specifically the drive axis, in Figure 4.13, the response of both sensors did not align as well, showing a R^2 value of 0.34, as in Figure 4.14. This behavior might be attributed to the active mode-matching technique employed in the MEMS gyroscope design, implying that the gyroscope is modulating the resonant frequency to offset the induced disturbance. However, such design specifics are not disclosed in the datasheet. Consequently, the deterministic behavior is not as strong as it is in the sense axis.

Comparison with dual-mass MEMS gyroscope

To screen the performance of gyroscopes, we modified the previous testing protocol, to a single dual-mass MEMS gyroscope from Murata, specifically the SCC2000 model. Although the amplitude and frequency spectrum of the test conditions are aligned with our region of interest, it is important to clarify that our preliminary screening tests were not explicitly tailored to match the identical excitation profile between the tests.

Upon analyzing time-domain graphical data, in the following Figures 4.15 and 4.16, it was evident that the gyroscope's output remained largely unaltered in response to external force applications. Owing to the fact that we opted for data

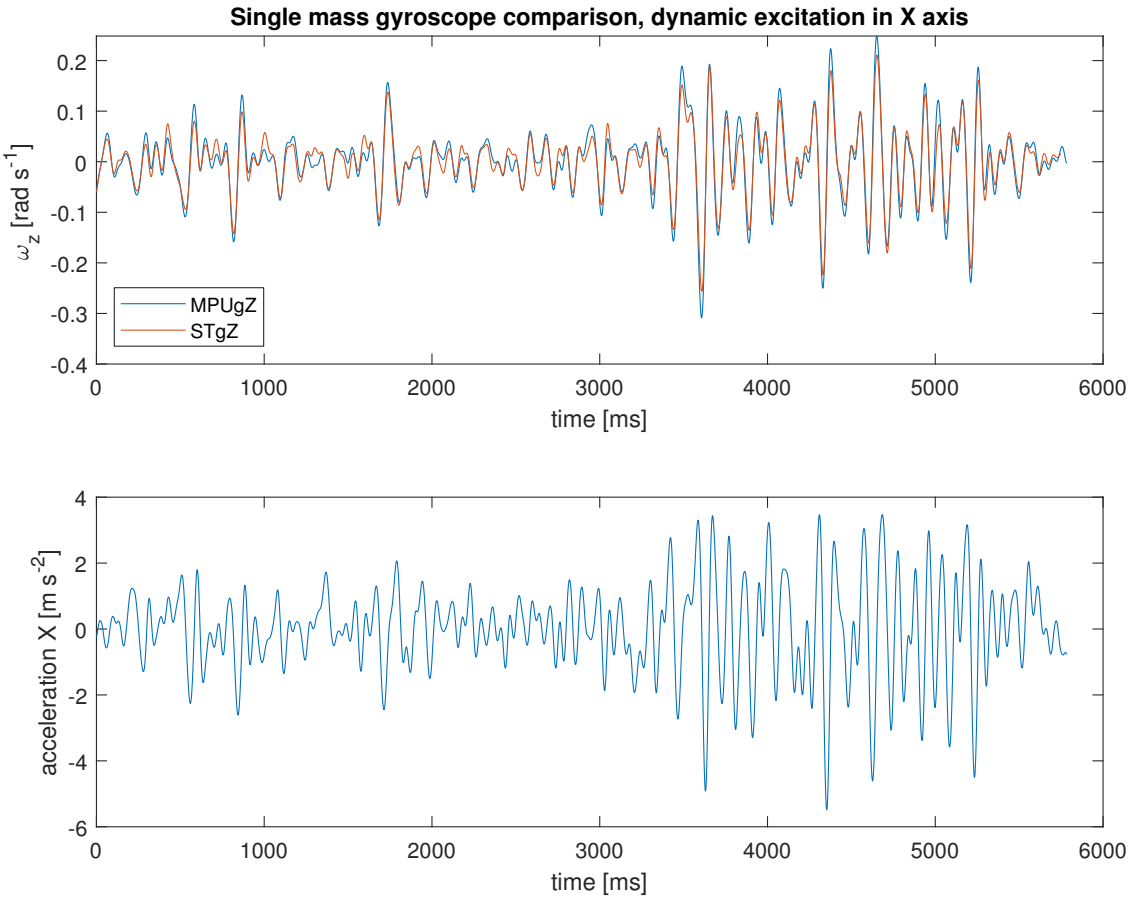

Figure 4.11: Single mass gyroscopes comparison, X axis dynamics.

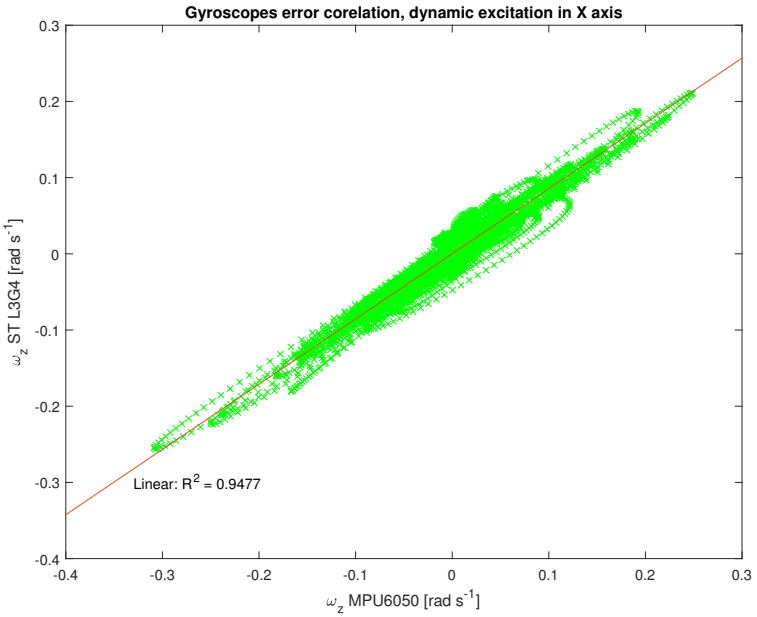

Figure 4.12: Correlation of two single mass gyroscopes, X axis dynamics.

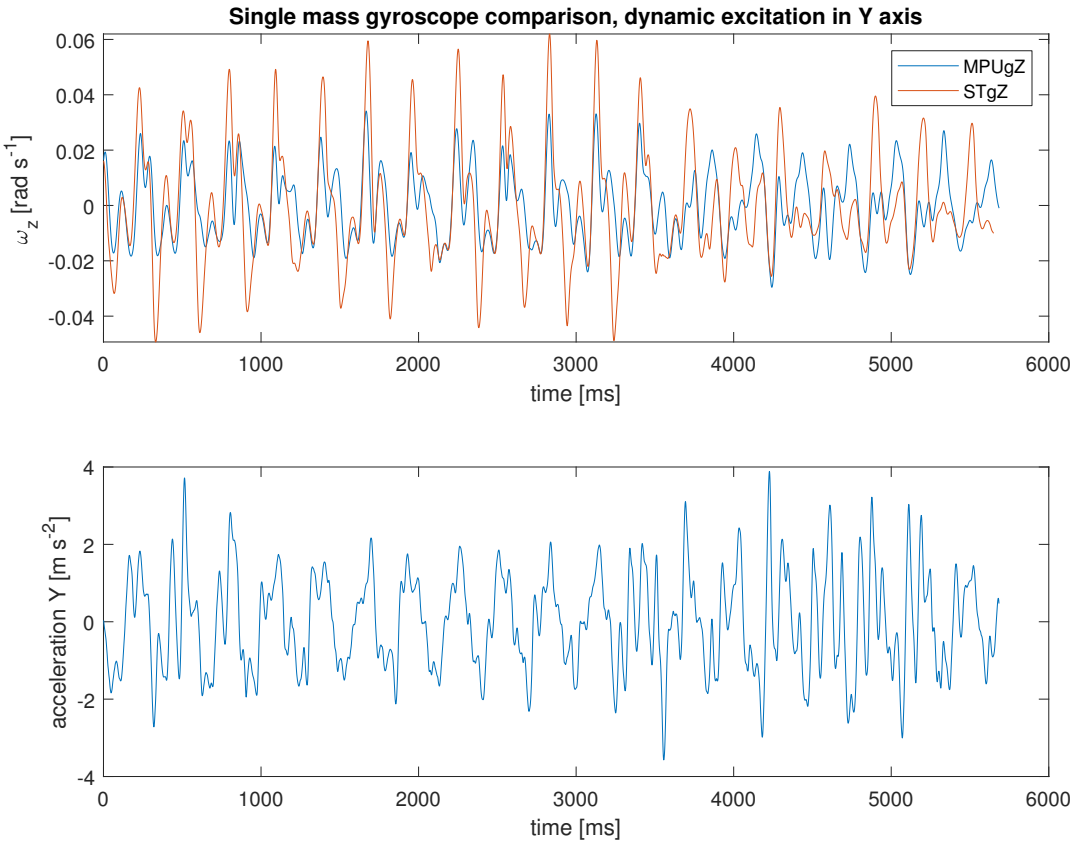

Figure 4.13: Single mass gyroscopes comparison, Y axis dynamics.

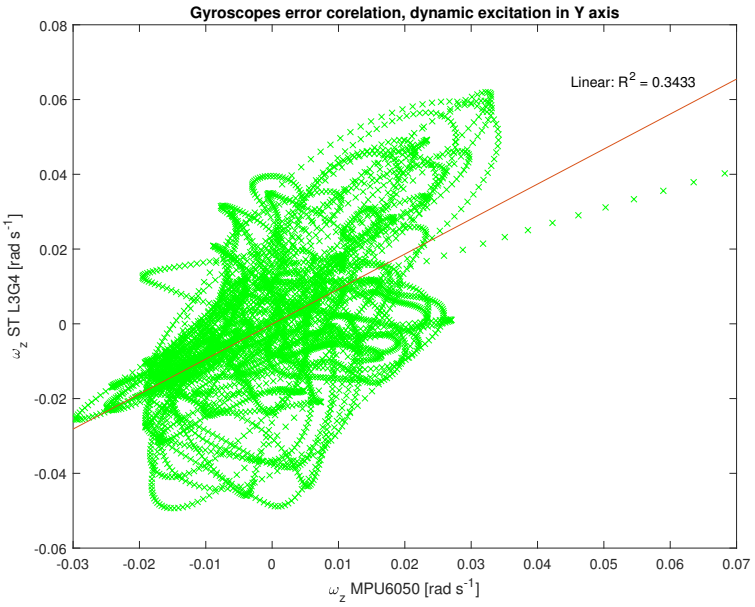

Figure 4.14: Correlation of two single mass gyroscopes, Y axis dynamics.

acquisition at the maximum allowable frequency, no filtering mechanisms were implemented, consequently, periodic noise is visible in the output graphs. Even when subject to external dynamic perturbations, the gyroscope's measurement demonstrated remarkable stability, deviating minimally from baseline levels, direct comparison in Figure 4.17.

In the realm of dual-mass MEMS gyroscopes, the device's unique mechanical architecture significantly mitigates the adverse impact of linear acceleration and so the jerk on sensor output. This stability is substantiated by the counter-phase oscillations of the paired masses, which produce a measured angular velocity derived from the differential forces exerted on them. Thus, we posit that these gyroscopes are inherently less vulnerable to disruptions caused by linear acceleration on the sensing axis, a claim that is supported by comparative data visualized in subsequent Figures 4.15 and 4.16.

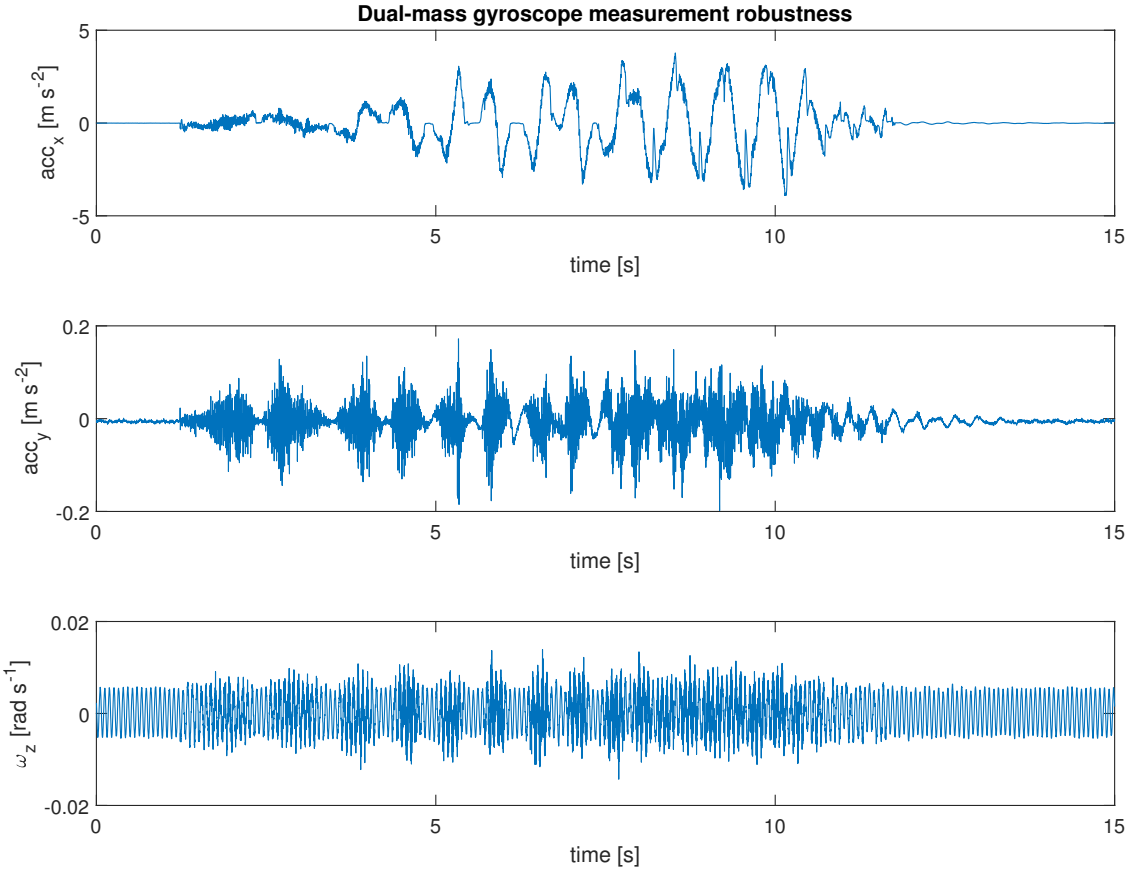

Figure 4.15: Murata SCC2000 IMU performance. X axis disturbance

.

4.3.3 Summary

Visualized measurements were repeatedly conducted with the results written in Table 4.4. Considering the comprehensive dataset acquired through our experiments,

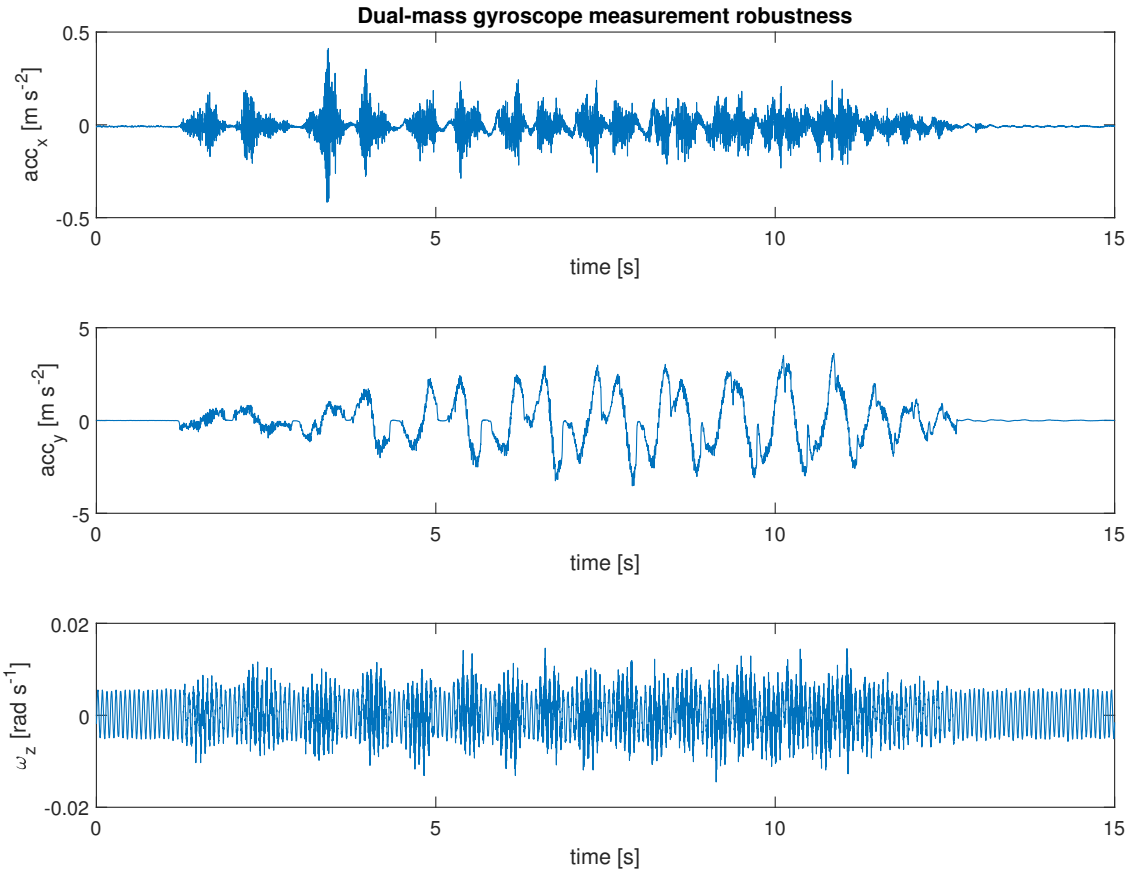

Figure 4.16: Murata SCC2000 IMU performance. Y axis disturbance

.

we can assertively conclude that the dynamical effects exert a pronounced influence on gyroscope measurements, particularly along the sensing axis. Specifically, when subject to dynamic excitations in the 'X' axis—which we have identified as the sensing axis—the error propagation into the measurement output is markedly present. Moreover, our data elucidates that the error propagation along the sensing axis is approximately five times more potent than when the same dynamic influences are applied along the drive axis.

R^2 of ω_z	1	2	3	4	5	6
X axis disturbance	0.9477	0.9612	0.9388	0.9547	0.9496	0.9569
Y axis disturbance	0.3433	0.2863	0.4201	0.3755	0.3311	0.4187

Table 4.4: R^2 values for different sets of data and single mass MEMS gyroscopes.

This significant differential in error susceptibility between the two axes underscores the necessity for targeted error correction algorithms that give preferential attention to the sensing axis. The findings not only corroborate our initial hypotheses but also pave the way for advancements in MEMS gyroscope sensor accuracy, particularly in real-world applications where dynamic excitations are inevitable.

Therefore, future work should focus on the development of more robust axis-specific error correction mechanisms to further mitigate these observed disparities in error propagation.

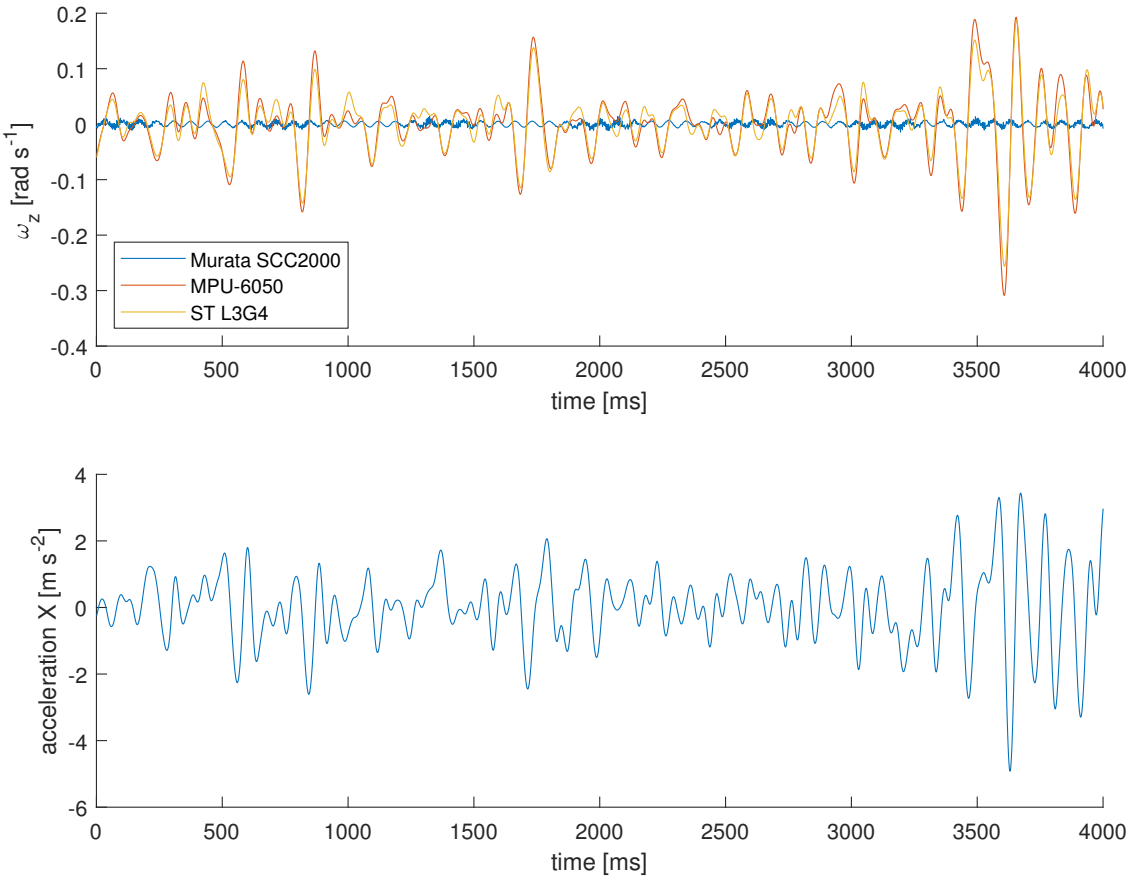

Figure 4.17: Direct comparison of gyroscope sensors affected in sense axis measured under dynamic load.

The antiparallel movement of dual mass MEMS gyroscope and differential signal processing effectively acts as a compensatory mechanism, mitigating distortions that would otherwise introduce significant biases as in single-mass systems. As a result, the dual-mass MEMS gyroscope exhibits a remarkable capacity for maintaining rigid and stable measurements, thereby substantively diminishing the susceptibility to external parasitic influences that frequently plague its single-mass counterparts.

4.4 Observation C: different gyroscopic sensors, wide range of excitations

Contrary to the observations previously described, we have evaluated the gyroscopic sensors in the presence of a discrete harmonic disturbance, maintaining a constant amplitude while varying the frequency of motion. The datasets gathered serve as a foundational basis for the advancement of compensation algorithms, pivotal for

the subsequent design and verification of compensators. Figures 4.18, 4.19, and 4.20 depict the captured data. A comparison of individual test runs and the quantitative assessment of the gyroscope's performance through standard deviation is illustrated in Figure 5.6, showcasing the raw gyroscope readings.

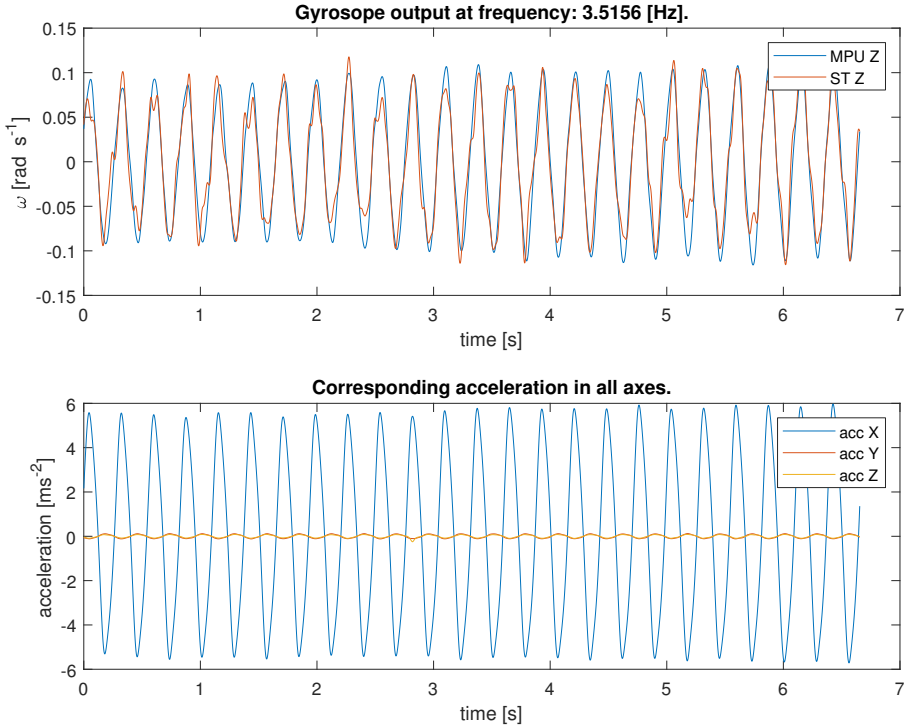

Figure 4.18: Sinusoidal harmonic excitation 3.5 Hz.

4.4.1 Summary

Our empirical data unambiguously reveal a highly consistent pattern of error across single-mass gyroscopic sensors. Specifically, when these sensors are subjected to harmonic translational motions with a sinusoidal profile, they provide a non-zero angular velocity. This behavior not only shares the frequency of the original sinusoidal excitation but also manifests with a distinct amplitude and a predictable phase shift relative to the excitation itself.

Upon evaluation of the single-mass MEMS gyroscope, it becomes evident that both acceleration and jerk have a direct and deterministic influence on the sensor's measurements. This was examined by applying targeted linear accelerations along the sensing axis of the gyroscopic sensor. In these controlled experiments, we observed a systematic bias in the sensor's output, affirming that the distortions were not stochastic but followed a well-defined pattern.

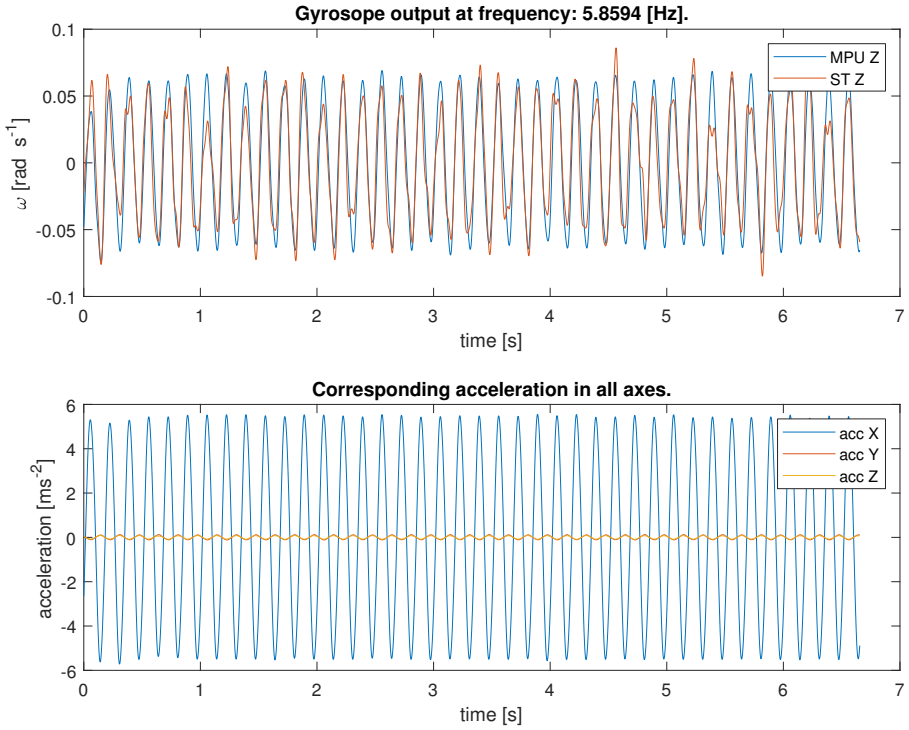

Figure 4.19: Sinusoidal harmonic excitation 5.9 Hz.

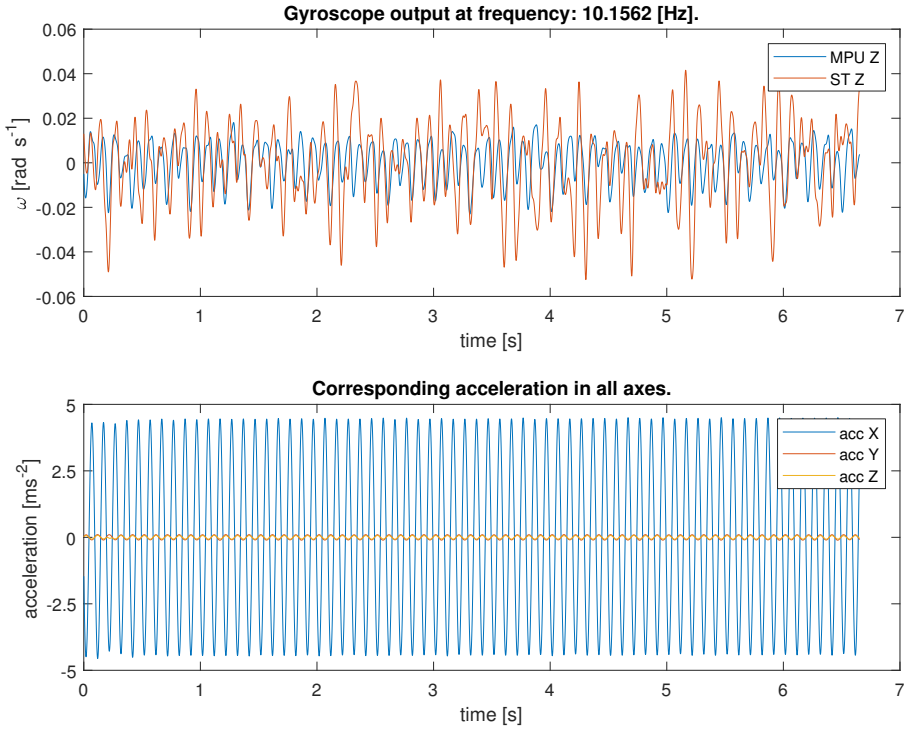

Figure 4.20: Sinusoidal harmonic excitation 10.1 Hz.

5 Compensational models

Based on comprehensive empirical analysis, we've conclusively determined that the specified boundary conditions exert a deterministic influence on the gyroscope's operational behavior. This understanding opens up the possibility of developing specialized compensation algorithms that, when provided with information about the inertial properties at play, can effectively mitigate unwanted behaviors.

In our experimental evaluation, we examined two distinct types of compensation algorithms:

- A nonlinear model parameterized via the nonlinear least squares method, which allows for precise tailoring of the model to fit observed system behavior.
- An artificial neural network-based model, which leverages machine learning techniques to adaptively correct for undesired gyroscope responses.

To ensure the integrity and reliability of our experiments, we implemented a set of following initial conditions:

- Absence of rotation around all axes of the measured system to avoid any external disturbances.
- Measurement of acceleration specific to the axes aligned with the gyroscope's excitation and measurement axes.
- Capture of gyroscope sensor output related to rotation about the axis that is orthogonal to both the excitation and measurement axes.
- Maintenance of stable ambient conditions, encompassing parameters such as temperature, pressure, and humidity, with a restricted measurement duration capped at five minutes to preclude long-term environmental drifts.

By establishing these initial conditions, we were able to isolate the variables under investigation and therefore ensure a more focused and accurate evaluation of the compensation algorithms. This meticulous approach strengthens the generalizability and applicability of our findings, providing a robust foundation for future endeavors aimed at gyroscope performance optimization.

5.1 Non-Linear Least Squares

Utilizing the nonlinear least squares methodology, we estimated the model parameters by invoking a specific harmonic equation of motion as our foundational analytical framework in Chapter 4. This approach allows for an optimal parameterization

that maximizes the fidelity between the empirical observations and the theoretical constructs. The harmonic equation serves as a critical mathematical representation that encapsulates the system's dynamic behavior, providing a robust basis for subsequent analyses and predictions. This optimized modeling technique not only enhances the accuracy of our simulation but also offers significant insights into the inherent nonlinearities affecting the system. Having a harmonic acceleration disturbance y_{acc}:

$$y_{acc}(t) = A_a \sin(2\pi f t) \tag{5.1}$$

We observe the gyroscope signal $y_{gyroZRO}$:

$$y_{gyroZRO}(t) = A_g \sin(2\pi f t + \varphi_g) \tag{5.2}$$

, where A_a stands for the acceleration amplitude of excitation, A_g stands for the amplitude of the ZRO gyroscope, f for the frequency of the harmonic signal and φ_g the ZRO phase change. In such a situation we can consider the jerk also smooth and harmonic.

Through empirical analysis, we observed a direct correlative relationship between the amplitude of the excitation oscillations and the amplitude of the error in gyroscope measurements. While the frequency of these oscillations remained invariant, a nuanced interplay was observed: namely, a phase shift existed between the excitation and the resulting error. This observation leads us to the necessity of finding the transfer function between dynamical disturbance measured by the accelerometer and the gyroscope's zero rate output, such that:

$$y_{gyroZRO}(s) = F(s) y_{acc}(s) \tag{5.3}$$

, where $F(s)$ can be considered as a feedforward compensator model in the s-domain.

In the scope of our work, we could think about the gyroscope measurement such that:

$$\omega_{gyro} = \omega_{real} + y_{gyroZRO} + y_{errors} \tag{5.4}$$

, where ω_{gyro} is MEMS gyroscope measurement, ω_{real} is real angular rotation, $y_{gyroZRO}$ is the error caused by dynamic disturbance, and y_{errors} are other error terms out of the scope of this work.

In the discrete form, we propose to derive the $y_{gyroZRO}$ as in equation 5.5.

$$y_{gyroZRO}[n] = A(f) y_{acc}[n - \varphi(f)] \tag{5.5}$$

, where n is the sample number in a discrete system, $A(f)$ is the compensation parameter of the amplitude as a function of the frequency of the harmonic disturbance, and $\varphi(f)$ is the phase shift as a function of the frequency of the harmonic disturbance.

As the phase shift is a parameter of a non-linear function, sinus, the NLS algorithm has to be involved in the evaluation of the error model. Besides the phase shift, the error was accompanied by fluctuating amplification levels contingent upon the frequency of the oscillations, such a dependency is clearly linear. This reveals a complex, multi-dimensional dynamic between the input excitation and the resultant error, underscoring the necessity for intricate models to comprehensively capture and potentially mitigate these dependencies.

Building upon the methodological framework from our research section 2.5.4, we applied the nonlinear least squares estimation technique to a distinct set of observations, captured across two different gyroscopic systems. The meticulous application of this computational approach facilitated a multi-faceted analysis, unveiling nuanced behaviors and correlations within each individual data set, thereby significantly enhancing the granular understanding of the gyroscopes under investigation. As a testament to the efficiency and precision of this optimization-based methodology, some of the findings, notably capturing trends in amplitude, frequency, and phase shift, have been described through graphical visualizations presented in the Figures 5.1, 5.2, and 5.3. Such dataset and accompanying analysis serve as a comprehensive empirical foundation.

Upon analysis of the error distribution from our gyroscope error compensated measurements, we have ascertained that our model's capability to account for dynamic disturbances is empirically substantiated, provided certain conditions are met. Specifically, the post-compensation error distribution ought to exhibit Gaussian characteristics for the model to be deemed valid, in Figure 5.4. Gaussian-like behavior in the error distribution becomes markedly consistent for frequencies exceeding 3.5 Hz.

By leveraging statistical measures such as skewness and kurtosis, in table 5.5, we have quantitatively confirmed the normality of the error distribution in the frequency range of interest. This validation not only lends credence to the model but also delineates the specific frequency range where it can be reliably deployed. Consequently, the model holds significant utility in scenarios with high-frequency gyroscope measurements, where dynamic disturbances are notably influential.

Conducting a comparative analysis, it becomes evident that the compensatory mechanism exerts a significant suppression of error, respectively the standard deviation of the Zero Rate Output (ZRO) under dynamic disturbances beyond the frequency of 3.5 Hz, see Figure 5.6. This suggests that the compensator is particularly effective in mitigating dynamic disturbances in frequencies between 3 to 12 Hz and potentially beyond, thereby stabilizing the ZRO's variability. This observation is pivotal, not only for substantiating the compensator's utility but also for delineating the frequency-dependent operational boundaries where the system exhibits optimal performance. Consequently, these findings provide invaluable insights for algorithmic refinement and set the stage for further research into adaptive

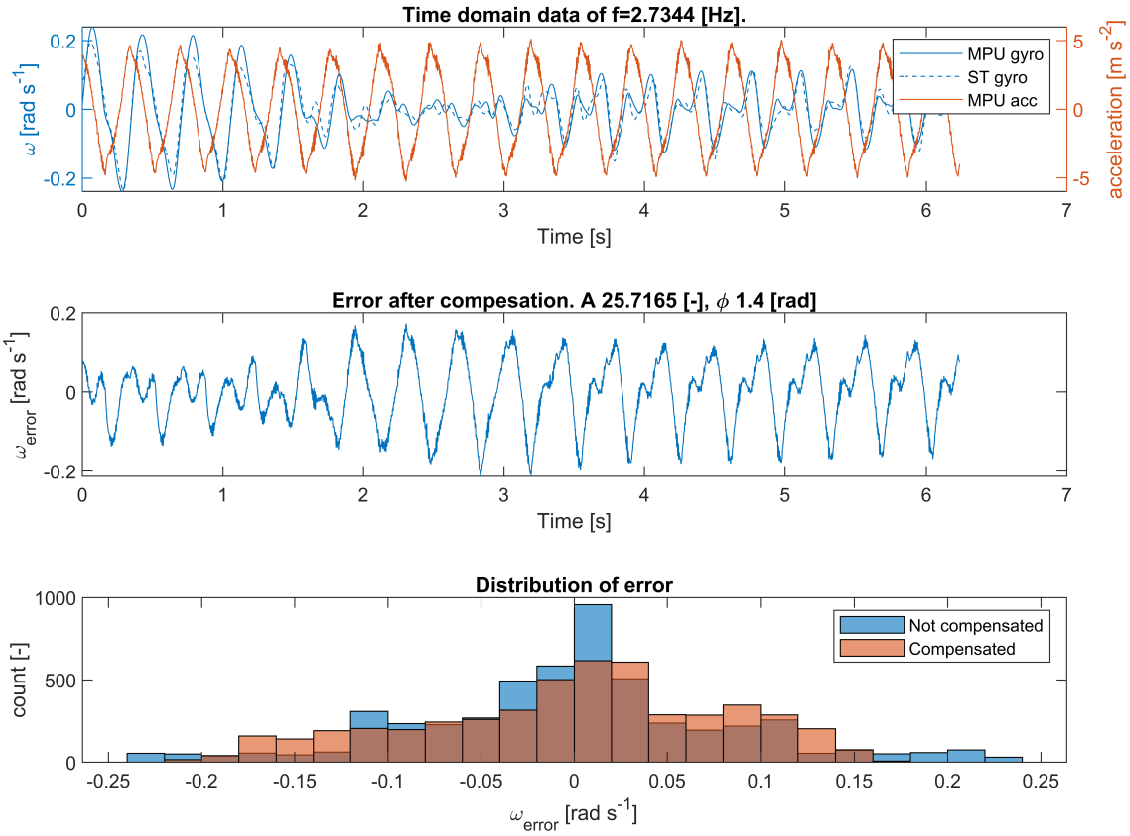

Figure 5.1: Sine harmonic excitation 2.7 Hz.

compensatory mechanisms in mechatronic systems.

Based on our data assessment, we would propose the storage of compensator parameters in a look-up table (LUT) equipped with linear interpolation functionalities for in-between data points. This structure offers an optimized solution for potential embedded systems by enabling rapid parameter retrieval and computational efficiency—a crucial requirement given the resource limitations typical in embedded environments. For future investigations, we have identified an intriguing pattern in the behavior of the gain parameter, which serves as the direct conduit for the propagation of acceleration signals to the gyroscope output. Specifically, the gain parameter manifests a quadratic growth trajectory, with the quadratic minima located around a frequency of 3.5 Hz. Bellow frequency of 3.5 Hz, i.e. quadratic minima, we have applied constant parameter A. In the context of phase shift parameters, our preliminary analysis reveals a linearly ascending trend across the measured dataset. The evolution of both parameters on frequency is in Figure 5.7.

5.2 Artificial Neural Network - ANN

As a complementary method for addressing zero-rate output gyroscope error compensation, more sophisticated methodologies, such as artificial neural networks, were

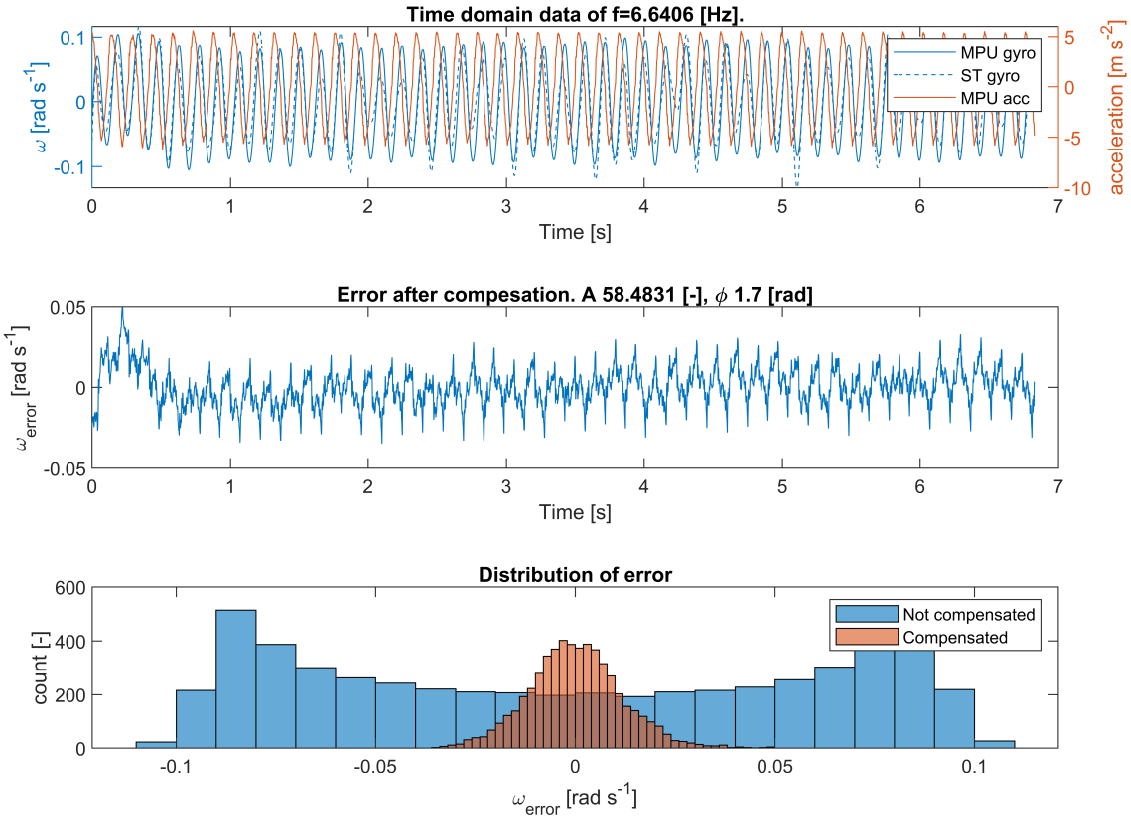

Figure 5.2: Sine harmonic excitation 6 Hz.

explored. In our prior research [11], we have achieved a promising compensatory framework by employing a nonlinear autoregressive neural network with feedback loops.

As of the present moment, there exists no universally accepted methodology for definitively determining the optimal complexity of an artificial neural network, as we have discussed in Section 2.5.5. The effectiveness of any chosen architecture must undergo empirical validation—initially on a dedicated validation dataset and subsequently in real-world applications. Given the diverse and unconstrained landscape of available methods, it is essential to establish a foundational framework from which we can iteratively approach an optimal architectural configuration.

In the context of our specific problem, we are working with a single-input, single-output topology where the input represents acceleration along a specific axis, and the output models the error associated with a gyroscope. The primary question revolves around the architecture of the hidden layers and the potential necessity for feedback mechanisms within the network.

One critical constraint to consider is the real-time hardware implementation, particularly with a focus on Field-Programmable Gate Arrays. Specifically, we intend to utilize the Xilinx ZYNQ 7000 series, which is readily accessible in our laboratory setup. This FPGA family offers a resource range consisting of several hundred to upwards of 2000 Digital Signal Processing (DSP) slices. This information is instrumental in defining not only the dimensions of the hidden layers but also the degree

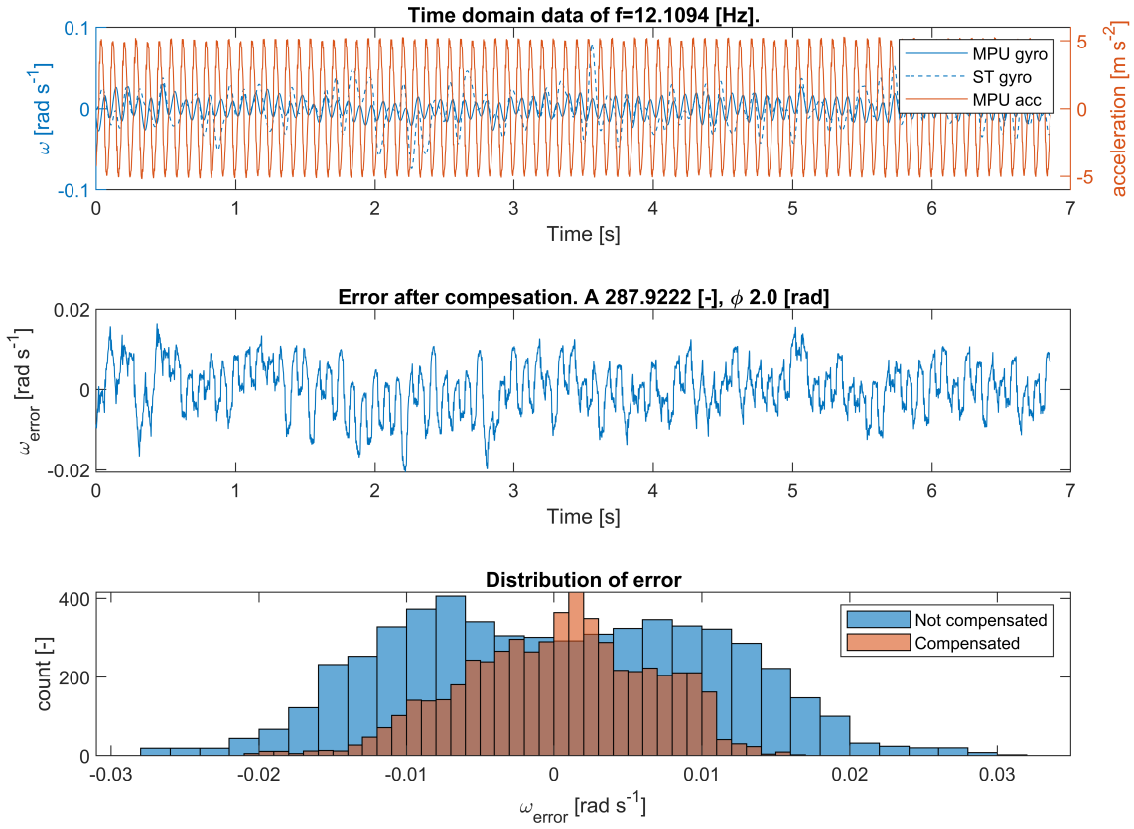

Figure 5.3: Sine harmonic excitation 12 Hz.

of interconnectivity within the network.

It is essential to note that each connection between neurons usually entails a multiplication of the input by a corresponding weight, followed by an addition operation to aggregate these weighted inputs. Such fundamental arithmetic operations will invariably consume the available FPGA resources, particularly the DSP slices in the case of the Xilinx ZYNQ 7000. Therefore, a keen understanding of these hardware constraints is vital in effectively determining the optimal complexity and size of the artificial neural network we seek to develop.

- Multiplication: DSP slices that are optimized for multiplication operations. A single multiplication can be done in one DSP slice.

- Addition: Adders are usually simpler and can be implemented in the FPGA's general logic slices. A single adder might consume a few LUTs (Look-Up Tables) and Flip-Flops (FF) within a slice.

- Wires and Routing: Interconnections also consume routing resources, although these are generally harder to quantify than logic or DSP slices.

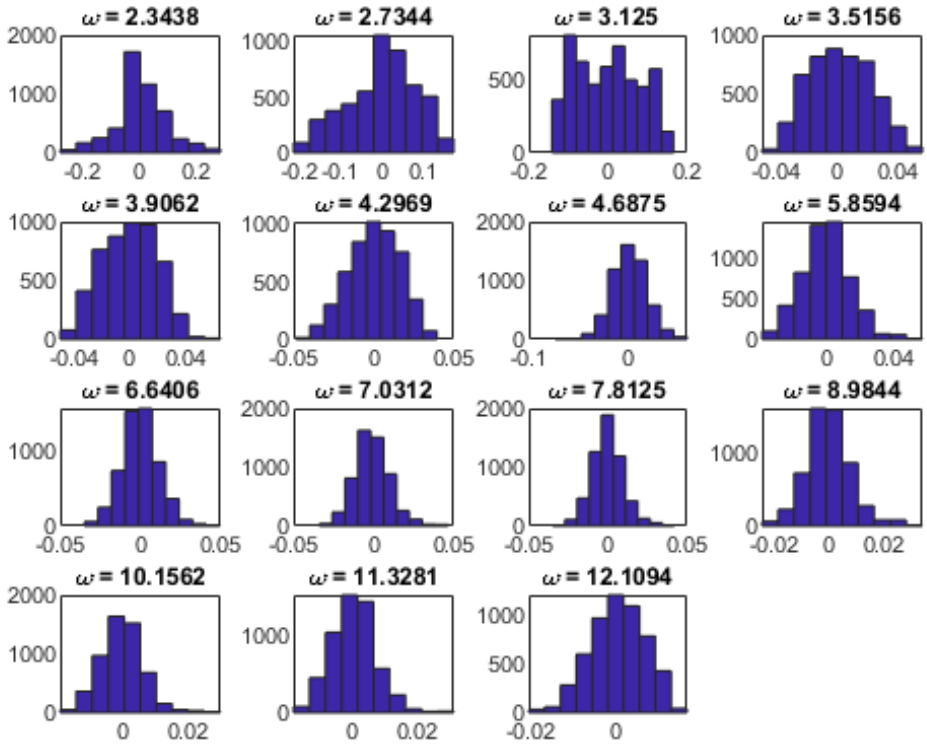

Figure 5.4: Histograms of error across multiple frequencies.

$\omega[Hz]$	3,13	3,52	3,91	4,30	4,69	5,86	6,64	7,03	7,81	8,98	10,16	11,33	12,11
Skewness	0,13	0,11	-0,02	-0,18	-0,03	0,24	0,24	0,48	0,28	0,38	0,44	0,42	-0,25
Kurtosis	1,83	2,27	2,36	2,48	3,44	3,36	3,53	3,92	3,71	3,82	4,16	3,83	2,70

Figure 5.5: Characteristics of compensated error.

Considering these points, one interconnection might consume the following:

- 1 DSP slice for multiplication.
- A few LUTs and FFs for the addition.

This is a simplified estimate and assumes fixed-point arithmetic and general availability of LUTs and FFs. Floating-point operations would consume considerably more resources. Also, the actual resource utilization can vary based on optimization techniques, pipelining, and other architecture-specific details. We can get the exact amount of resources necessary by the compilation of actual code, but for the selection of baseline, this would require an inadequate amount of time.

Layer A consisting of N_a neurons interfacing the layer B consisting of N_b neurons combinatorically result into n_{ab} interconnection such that:

$$n_{ab} = N_a N_b \tag{5.6}$$

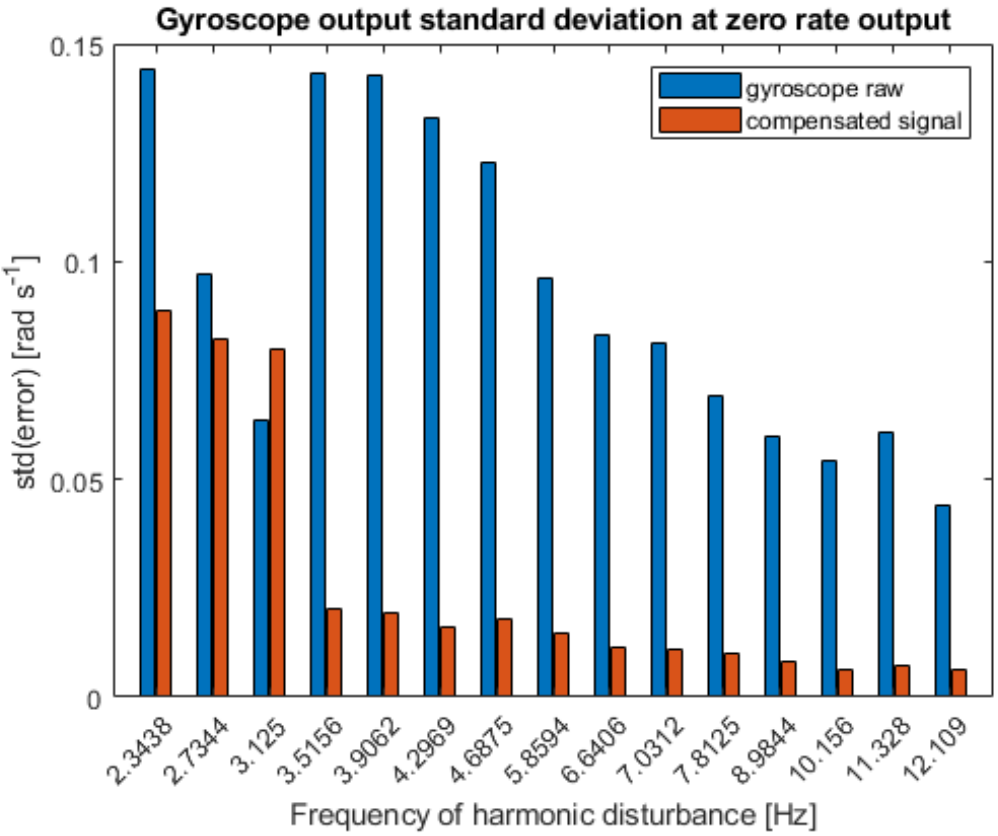

Figure 5.6: Comparison of ZRO output. With and without compensator.

Having more than two layers:

$$n_{a\ldots i} = \sum_{i=m}^{n} N_m N_{m+1} \tag{5.7}$$

As we have available 2000 DSP slices, we selected the 4-layer structure with 30, 30, 20, and 10 neurons in each layer as the most complex, resulting in 1700 interconnections inside the hidden layer and so the adequate amount of DSPs.

5.2.1 NN definition and training

In the development of our artificial neural network architecture, we leveraged the capabilities of MATLAB's Deep Learning Toolbox as it is in Figure 5.8. This toolbox offers an intuitive user interface that streamlines various essential tasks, such as specifying the number of neurons in each layer and configuring the feedback connections between layers, among other architecture-related parameters.

To systematically explore the architecture space for an optimal neural network configuration tailored to our specific task, we devised an automated grid search algorithm. This algorithm programmatically iterates through a predefined set of neural network structures, as outlined below:

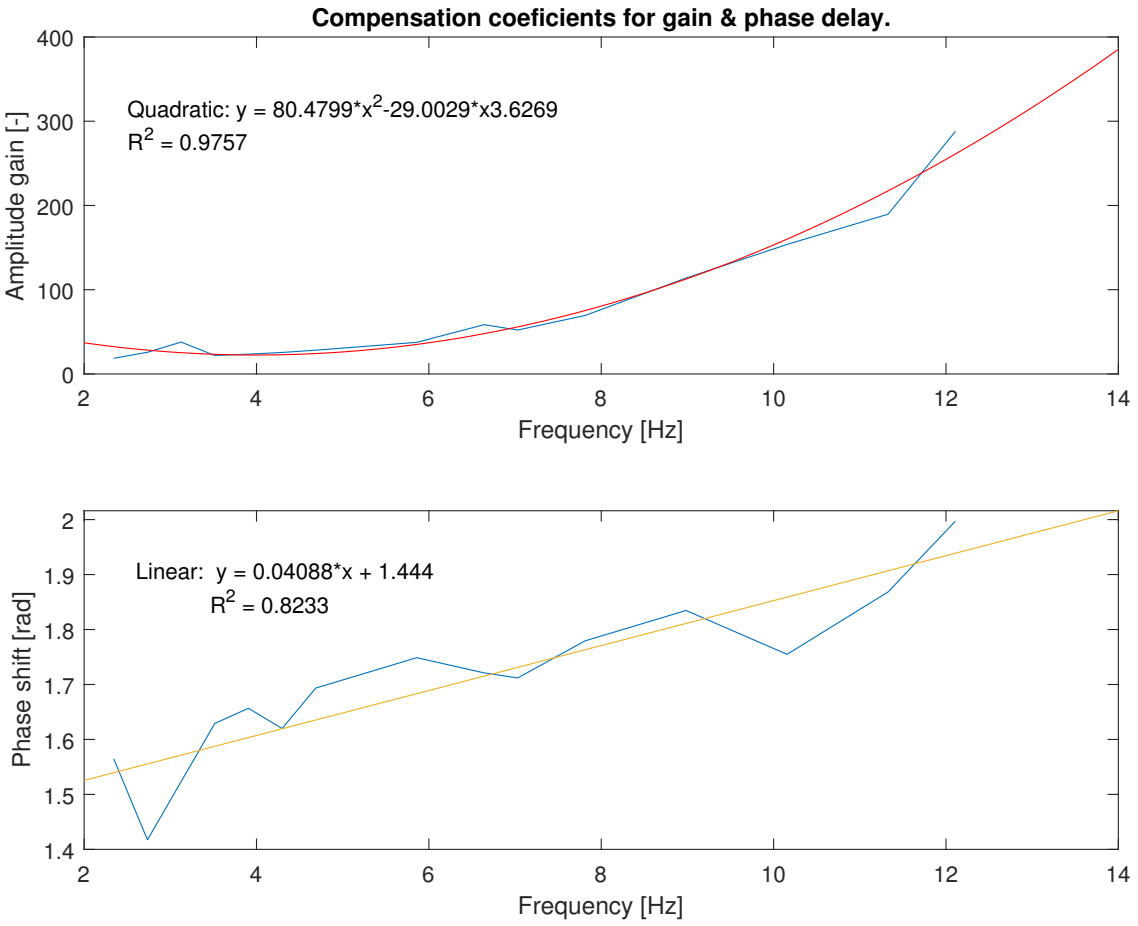

Figure 5.7: Evolution of compensator parameters for 5 ms^{-2} acceleration amplitude.

1. Initiate the search with a single-hidden-layer neural network containing 10 neurons.
2. Incrementally increase the neuron count in that layer by 10, up to a maximum of 30 neurons.
3. Introduce feedback connections from the last layer back to the first layer and evaluate the performance of these recurrent architectures.
4. Add an additional hidden layer and execute a combinatorial search across various neuron counts in each layer, both with and without feedback connections.
5. Continue this iterative process until the architecture reaches a complexity of four hidden layers, each containing up to 30 neurons, complete with feedback loops where specified.

Each candidate architecture generated by this automated grid search was subjected to a rigorous training process as in Figure 5.9, followed by validation using a separate dataset. This comprehensive approach ensures that the most effective neural network structure can be revealed. As an evaluation criterion for selecting

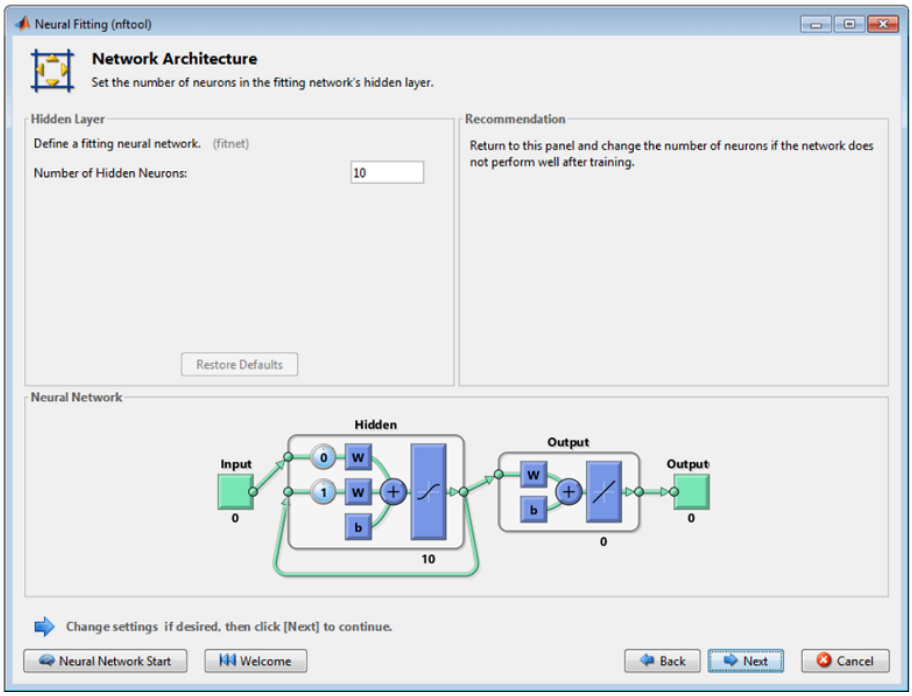

Figure 5.8: Initial definition of the ANN structure.

the best-behaving artificial neural site we have selected the standard deviation of compensated measurement.

The implementation of our automated grid search algorithm yielded a plethora of artificial neural network configurations. However, many of these generated architectures exhibited suboptimal performance when assessed against our predefined constraints, which chiefly pertained to the maximal number of interconnections—that is, the number of DSP slices utilized. To elucidate the performance landscape of the architectures generated, we focused our analytical attention on evaluating the standard deviation, like in the previous chapter, of key performance metrics at the extremities of our search space.

An analysis of the standard deviation of the compensated signal, when compared to the reference signal in the context of a single-layer artificial neural network (ANN), reveals no marked improvement over the baseline, unprocessed signal, in Figure 5.10. Additionally, the single-layer ANN introduces further complications. Specifically, it may inject a time delay into the output signal, an effect that becomes progressively pronounced with longer feedback delays.

This finding has critical implications for real-time systems that require immediate or near-instantaneous response, as any introduced time delay may negatively impact the system's performance. Moreover, it calls into question the viability of utilizing a simplistic, single-layer ANN for this particular application, thereby warranting an exploration into more complex, multi-layered architectures or alternative computational strategies that can accomplish effective signal compensation without the accompanying drawbacks.

At the end of our exploratory grid search spectrum, we encounter artificial neural

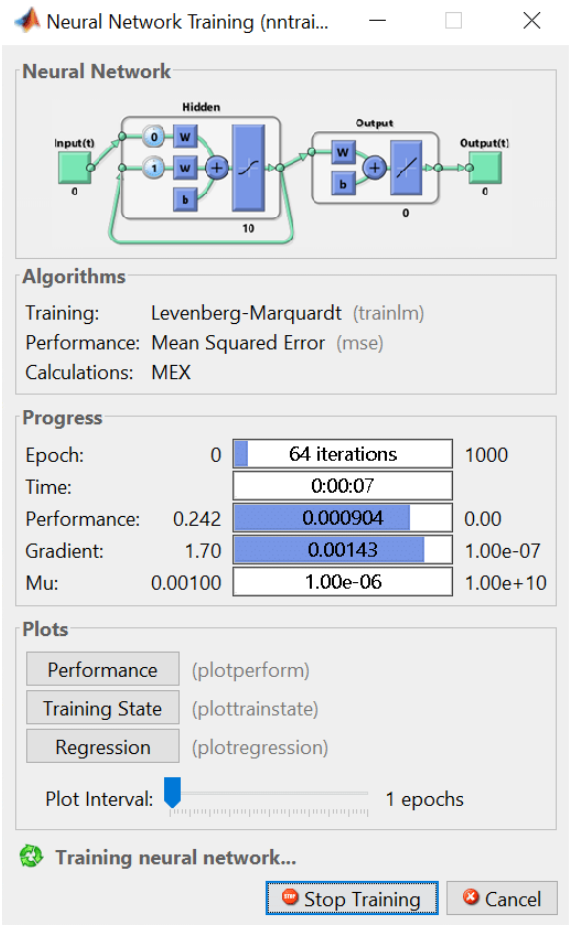

Figure 5.9: Matlab Deep Learning Toolbox, training ANN of the visualized structure, 10 neurons 1 layer with feedback.

networks composed of four hidden layers, both with and without feedback loops. The introduction of feedback mechanisms can negatively impact system performance, primarily due to induced time delays, and did not reveal any improvement in compensator performance. In light of this, our preference leans towards employing ANNs devoid of such feedback layers.

Our empirical evaluations substantiate this preference: the four-hidden-layer ANN without feedback, in Figure 5.11 not only mitigated the issue of time delays but also outperformed its feedback-enabled counterpart in key performance metrics. Given these results, it becomes clear that focusing on an ANN architecture with feedback loops would be counterproductive for our specific application, as it offers no tangible benefits while introducing additional complexities and inefficiencies.

The artificial neural network did exhibit some efficacy in mitigating errors, particularly within lower frequency bands. A representative example of this performance enhancement can be observed in Figure 5.12. In this illustration, the key performance indicator under scrutiny is the standard deviation of the error, which is readily discernible from the associated histogram located at the bottom of the Figure.

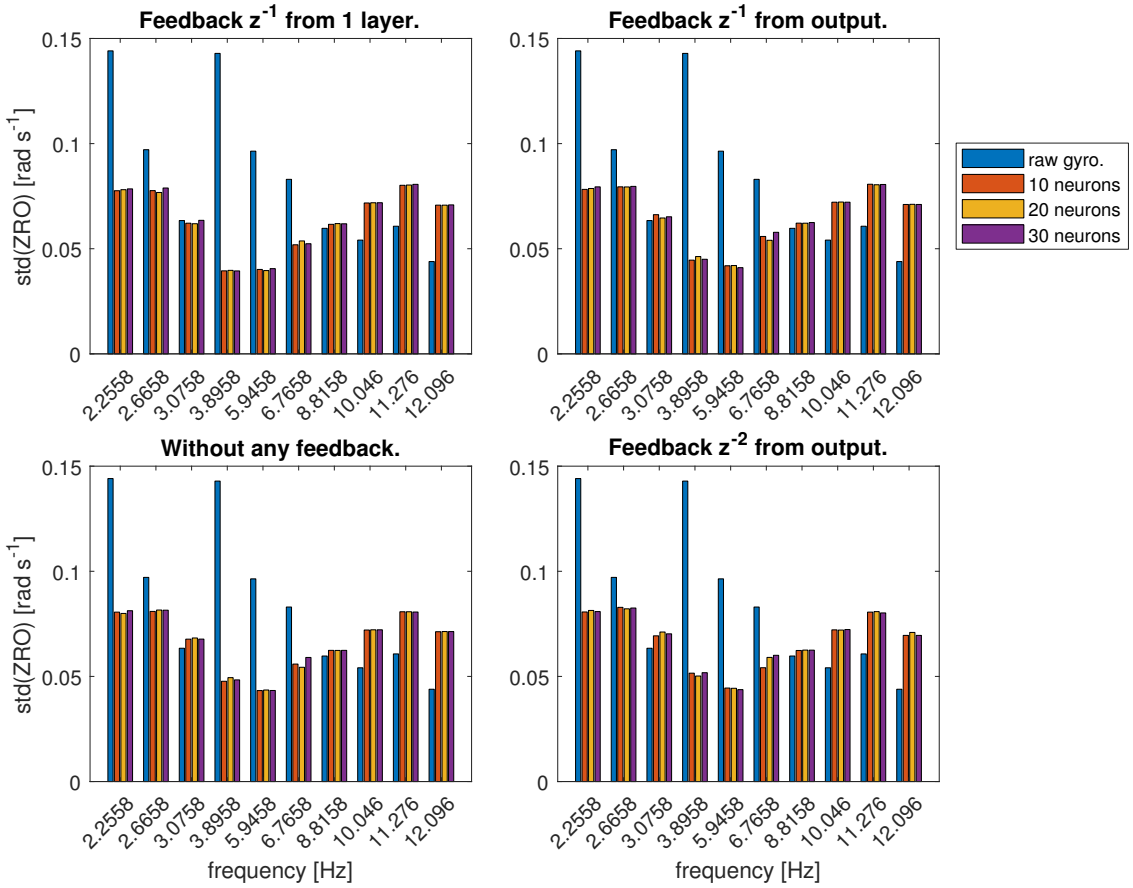

Figure 5.10: Performance of single layer compensator.

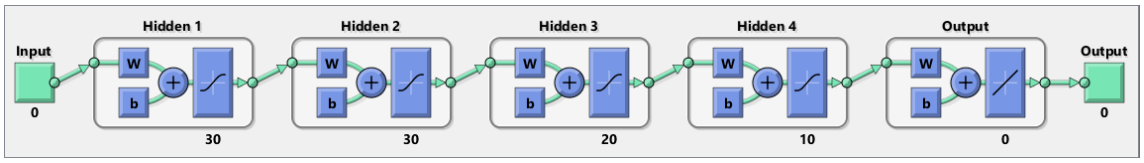

Figure 5.11: Illustrative neural network model without feedback.

The histogram of standard deviations serves as an empirical validation of the model's capability to suppress zero rate output, thereby underscoring the network's partial success in refining the gyroscope's Zero Rate Output within specific frequency ranges. While this doesn't negate the challenges we've outlined with respect to broader frequency ranges, it does suggest that the ANN approach holds promise and merits further investigation for targeted frequency suppression.

However, as illustrated in Figure 5.13, even the most intricate artificial neural network architecture within our search purview failed to comprehensively compensate for the zero rate output of the gyroscope across the entire spectrum of dynamic disturbances. Intriguingly, at certain frequencies, the standard deviation of the compensated signal deteriorated, performing worse than the original, uncompensated signal.

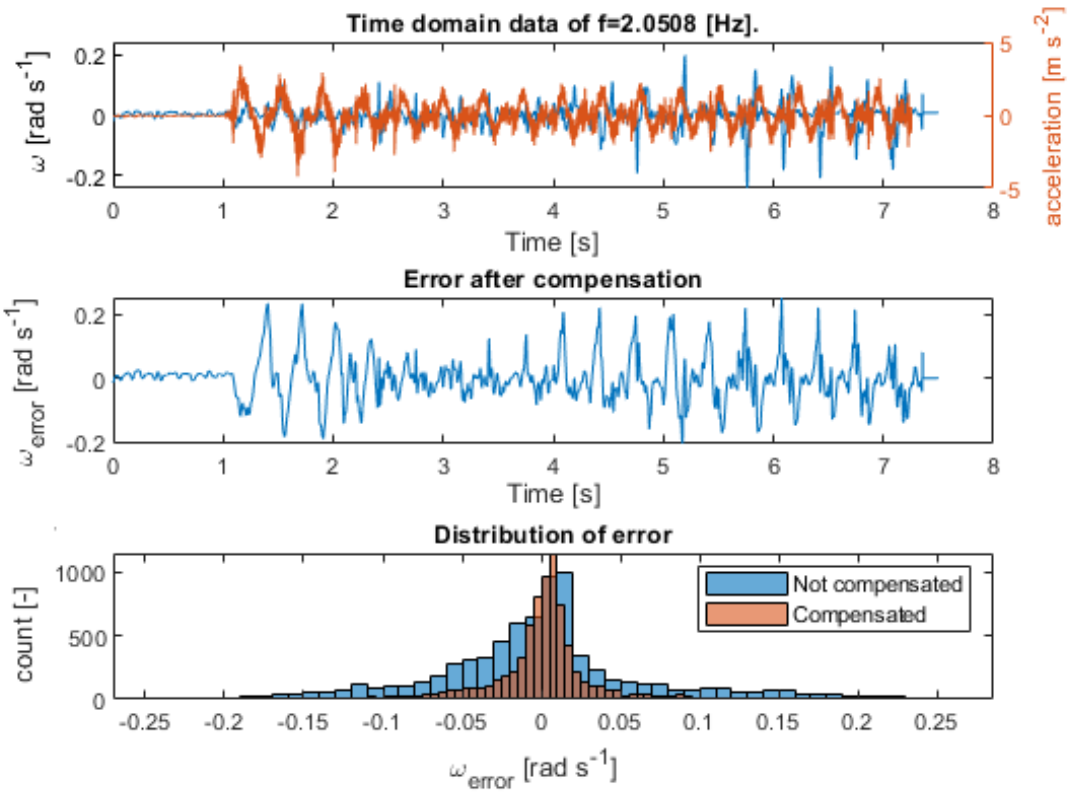

Figure 5.12: Exemplary data of 4-layer, 120 neuron mesh.

This counterintuitive behavior can be attributed to a multitude of potential factors. One possibility is that the neural network's architecture is suboptimal for this particular task, perhaps due to an insufficient number of neurons or layers, leading to an underrated model that is incapable of capturing the underlying complexities of the system. Alternatively, issues such as underfitting could be at play, where the model fails to generalize well to new or unseen data. Another plausible explanation lies in the inadequacy of the training dataset, especially if it lacks sufficient representation in the problematic frequency regions.

These observations underscore the need for a more nuanced approach to ANN design and training, specifically tailored to tackle the challenges presented by the dynamic disturbances affecting gyroscope measurements.

5.3 Summary

In this part, we have introduced two distinct methodologies aimed at mitigating the dynamic disturbances affecting gyroscope measurements, i.e. zero rate output. The empirical data underpinning our analyses were collected from actual hardware and not through simulations.

Two specific single-mass MEMS gyroscopes, namely the MPU 6050 and the

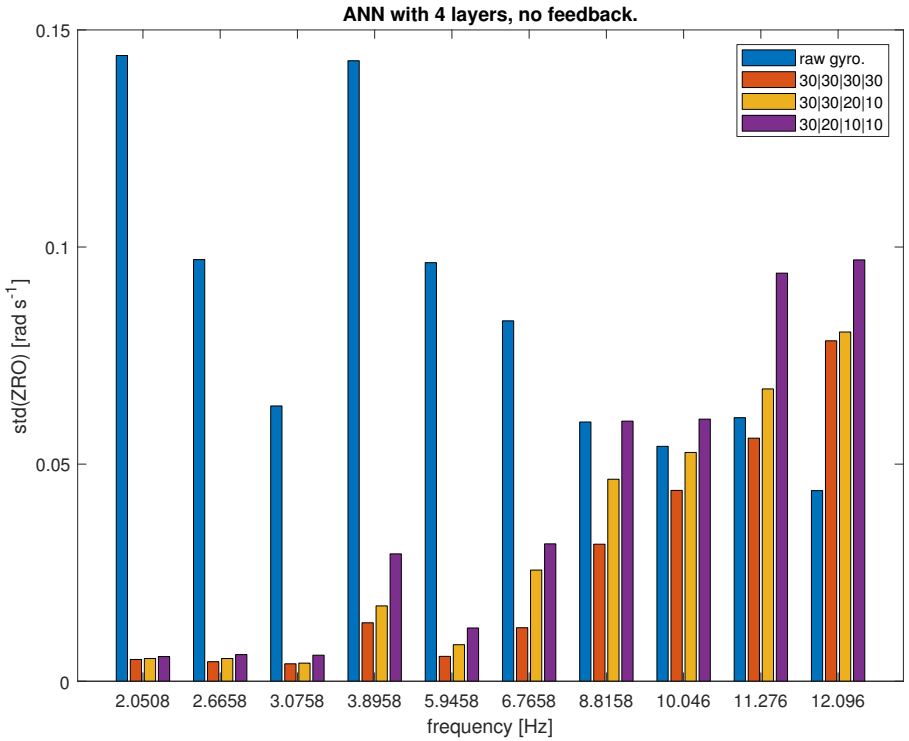

Figure 5.13: Comparative performance of 4 layers meshes, without feedback.

ST L3G4, emerged as the most viable candidates for further investigation based on their comparable response characteristics under perturbation conditions. Each method was rigorously developed and validated using independent datasets, which were procured from real-world measurements and subsequently post-processed on a personal computer.

For the purpose of this study, we focused on periodic signals with acceleration amplitudes up to 5 ms^{-2} and within a frequency range spanning from 2 Hz to 12 Hz. We employed standard deviation as our principal metric to assess the efficacy of the compensatory algorithms, as in an ideal state of no rotation, the standard deviation of the compensated gyroscope under dynamic disturbance should stay 0 $rads^{-1}$.

While neither compensatory algorithm could comprehensively address disturbances across the entire frequency spectrum, they did exhibit complementary strengths on opposite sides of the frequency region of interest. Potential synergic interplay suggests the utility of integrating both methods into a single, robust compensatory framework. However, it's worth noting that the ANN-based approach poses significant computational challenges, especially when considering its implementation on conventional FPGA platforms, where resource constraints might limit the complexity of the neural network that can be deployed.

6 Implementation on RT HW

In our research setup, we employed an automated code generation process through a cohesive toolchain that integrates several software and hardware components. On the software end, the toolchain encompasses the MathWorks suite, including Matlab, Simulink, Matlab Coder, Simulink Coder, and Embedded Coder. Additionally, we utilized MPLAB Device Blocks for Simulink, to facilitate code generation compatible with Microchip hardware. From the Microchip set of tools we used the compiler MPLAB XC, MPLAB X IDE, microcontroller programmer PicKit and microcontroller itself. The microcontroller applied is a dsPIC33FJ128MC804, part of Mechatronic laboratory education kits for students. These kits offer a range of basic peripherals aimed at enhancing the hands-on learning experience. It consists of several basic peripherals for interactive user experience, like LED, potentiometers, and buttons, but also auxiliary inputs, outputs, and busses. Such a bus, I2C, was used to read out values from the measurement device. Protocol needed to be specifically implemented as it is dependent on specific vendor and usage.

Given that I2C protocol implementation is often vendor-specific and tailored to particular use cases, we implemented a protocol, thereby ensuring seamless communication with our specific measurement apparatus. This intricate blend of software and hardware tools and custom implementations forms the cornerstone of our experimental infrastructure.

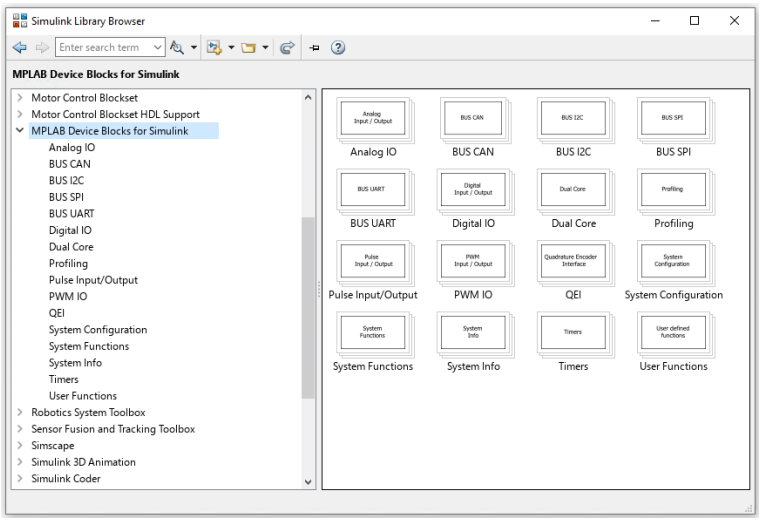

Figure 6.1: Prepared support tools in MPLAB Simulink toolbox.

In addition to the measurement subsystem of our experiment, it was essential

to develop a mechanical excitation subsystem responsible for actuating the device under test (DUT), specifically the gyroscope. This was achieved by employing a DC motor mechanically coupled to a trolley that carries the DUT. Control signals for the motor's excitation were generated through a low-level driver governed by Pulse Width Modulation (PWM), Enable, and Direction (DIR) signals.

These actuation commands were synthesized using an MF624 card and were integrated into a closed-loop control scheme. This scheme featured a current controller and firmware-based interlocks to limit the mechanical travel distance of the test apparatus. To facilitate real-time control at a 1 kHz frequency, we incorporated Matlab's Simulink Real-Time Toolbox. This toolbox ran on a Windows PC and interfaced with the MF624 card, thereby ensuring seamless and robust control of the mechanical excitation subsystem within predetermined operational boundaries.

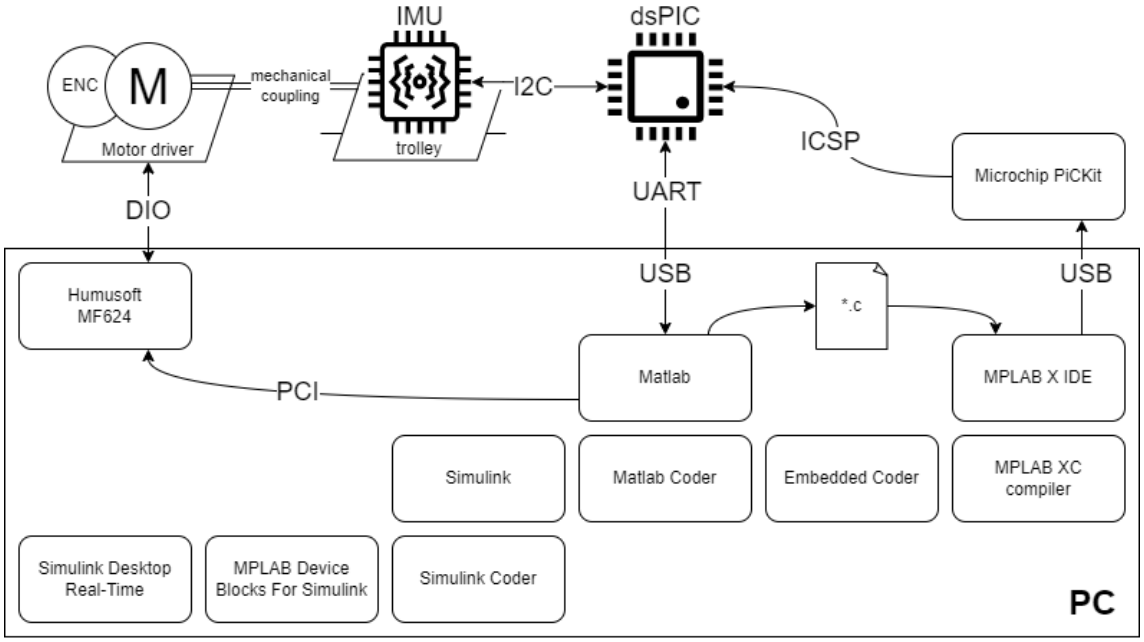

Figure 6.2: Schematic of the system used for prototyping and research.

6.1 Data handling implementation

The core of the auto-generated code is illustrated in Figure 6.4. This schematic representation provides a comprehensive view of the underlying logic and sequence of operations. It serves as a blueprint for understanding the algorithm's inner workings.

In our research setup, we employ a gyroscope for measurements that communicates via an I2C bus. The establishment of an effective communication protocol is crucial for ensuring reliable data transfer in such configurations, especially with the aim for 800 Hz sampling frequency. We leveraged the MPLAB Block for Simulink for the implementation of this protocol, owing to its ease of use and adaptability.

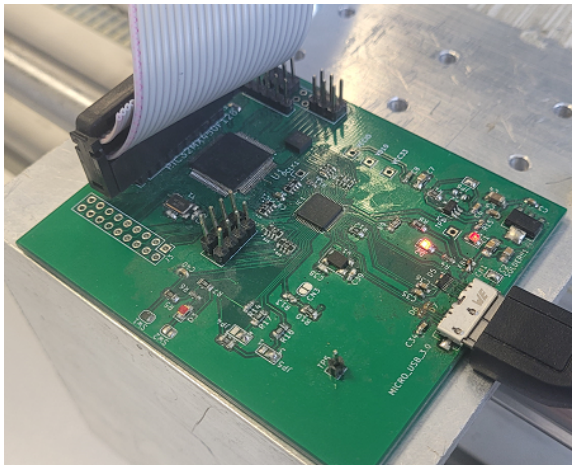

Figure 6.3: Experimental measurement HW.

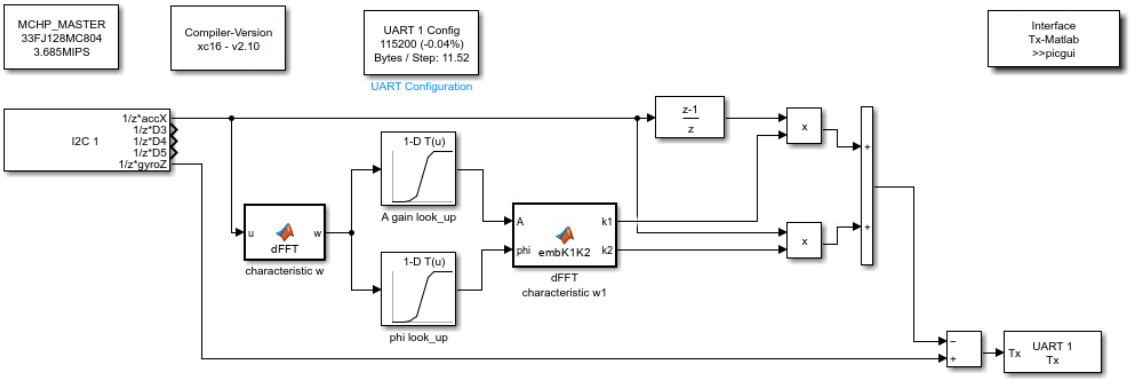

Figure 6.4: Example of Simulink Implementation.

This choice aligns well with rapid prototyping methodologies, facilitating quick iterations and modifications to the communication layer. To elucidate the practical implementation of this I2C-based communication scheme, Figure 6.5 offers a detailed depiction of the actual application. This visual aid serves not only to demonstrate the architecture but also to underscore the efficiencies and capabilities brought forth through the use of MPLAB Block in a Simulink environment. To extract the processed values from the microcontroller, we utilized a UART readout service. To ensure reliable data transfer at high frequencies, it implements a buffering mechanism on the microcontroller that organizes the data into throughput-efficient blocks before transmission. It is important to note that this approach inherently introduces some delay in the real-time data readout. However, given the focus of our investigation, which is the verification of system performance rather than real-time monitoring and operation, this is not a concern. The principal control actions are executed directly on the microcontroller, negating the need for flawless real-time data transfer for the objectives of this study.

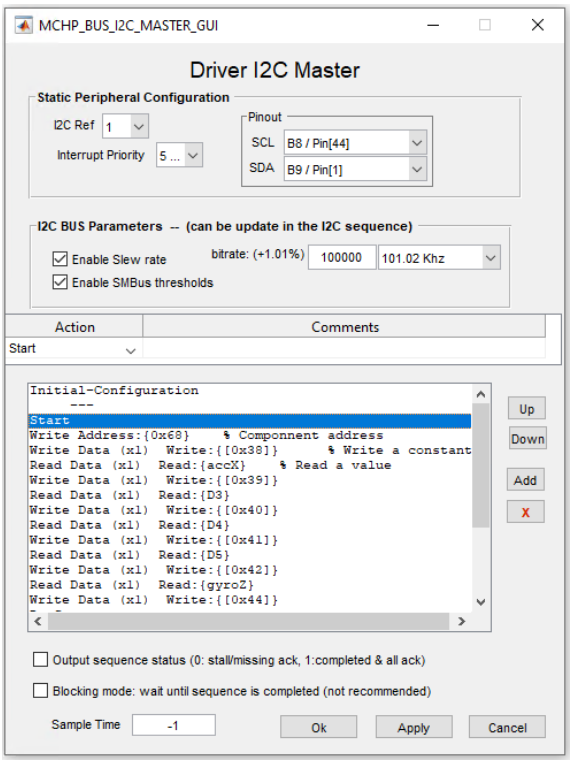

Figure 6.5: Setting of I2C bus protocol.

6.2 Artificial neural network algorithm

In the case of the compensational algorithm based on an artificial neural network, we can evaluate the potential for embedded systems and potentially microcontrollers by following assumption. In the case of the linear weight of neurons, we can assume that the whole network can be converged to the set of multiple matrix operations such for the forward propagation we follow:

- Input Layer: If the network input is a vector $\bar{x}$ of size $N \times 1$, it can be treated as a matrix where N is the number of input features. In our case 1
- Hidden Layers: The neurons in the hidden layers can be grouped into a matrix $\bar{H}$. The weighted sum of inputs to each neuron in the hidden layers can be calculated as $\bar{Z} = \bar{W} \times \bar{X} + \bar{b}$, where $\bar{W}$ is the weight matrix, $\bar{X}$ is the input matrix, and $\bar{b}$ is the bias vector. The activation function (e.g., ReLU, Sigmoid, etc.) is then applied element-wise to $\bar{Z}$ to obtain the output $\bar{H}$ of the hidden layer.
- Output Layer: Similarly, the output $\bar{O}$ can be calculated as $\bar{O} = \bar{W}_{out} \times \bar{H} + \bar{b}_{out}$ followed by the application of an activation function if necessary.

In the proposed design, we consider implementing an artificial neural network comprising four layers with respective neuron counts of 30, 30, 20, and 10. The weight matrices $\bar{W}$ are dimensioned as 30×30, 30×20, and 20×10, necessitating

the storage of 1700 weight parameters. The target operational frequency for this network is 800 Hz.

A computationally demanding aspect of this network is the matrix-vector multiplication. To elaborate, a single dot product operation for one row in a 30×30 matrix requires 30 multiplications and 29 additions, summing up to 59 computational operations. When this is scaled for the entire matrix, we arrive at 1770 operations. On the dsPIC33FJ128MC804 microcontroller, each of these operations involves 2 instructions for loading the numbers, 1 for multiplication, and 1 for storing the result—a total of 4 instructions per operation.

Extrapolating these requirements to an 800 Hz operational frequency, the computational load for just the first layer alone reaches approximately 5.6 Million Instructions Per Second (MIPS), which already exceeds the microcontroller's processing capabilities, which is 3.6 MIPS in standard. Additionally, the 16KB of SRAM on this microcontroller is insufficient to store the requisite amount of data, given that the weight parameters alone would consume a substantial portion of this memory.

Consequently, the implementation of such an ANN architecture on this specific microcontroller is infeasible. Notably, any reduction in the network size to fit the computational and memory constraints would compromise the ANN's performance to a point where it would no longer offer significant improvements in compensating for the gyroscope errors.

6.3 NLS-based compensator implementation and performance

For the successful real-time deployment of our proposed algorithm on hardware, it is imperative to carefully consider the computational constraints intrinsic to embedded systems. These constraints predominantly encompass memory capacity and the computational latency associated with the execution of complex, non-linear functions. To circumvent these limitations, we advocate for the adoption of a compensation algorithm founded on a look-up table (LUT) infrastructure. This LUT will house precomputed values that have been empirically determined through system identification techniques, as delineated in the section 5.1. The integration of phase shift calculations into our compensatory algorithm presents a particularly intricate challenge, especially when computational efficiency is paramount. As a part of an innovative solution to this complex issue, we suggest implementing the phase shift using a linear combination of weighted sine and cosine functions. A general sinusoidal function-in our case phase shifted response of the gyroscope:

$$A\sin(2\pi ft + \varphi) = y_{gyroZRO}(t) \tag{6.1}$$

can be rewritten using trigonometric identities into the form, which is a sum of weighted sine and cosine terms:

$$A(\sin(2\pi ft)\cos(\varphi) + \cos(2\pi ft)\sin(\varphi)) \tag{6.2}$$

where, $A\cos(\varphi)$ and $A\sin(\varphi)$ are the weights for the $\sin(2\pi ft)$ being measure and $\cos(2\pi ft)$ numerically derived. By adjusting the weights $A\cos(\varphi)$ and $A\sin(\varphi)$, you can effectively implement a phase shift φ in the original sinusoidal function. This form is particularly useful in signal processing and control systems, as it enables easier analysis and manipulation of sinusoidal signals.

In the section 5.1, we identified harmonic acceleration signals as a significant source of disturbance that necessitates robust calculation or measurement of the corresponding jerk signal. This involves applying an appropriate handling of the source signal and integrating it into the gyroscope compensatory mechanism.

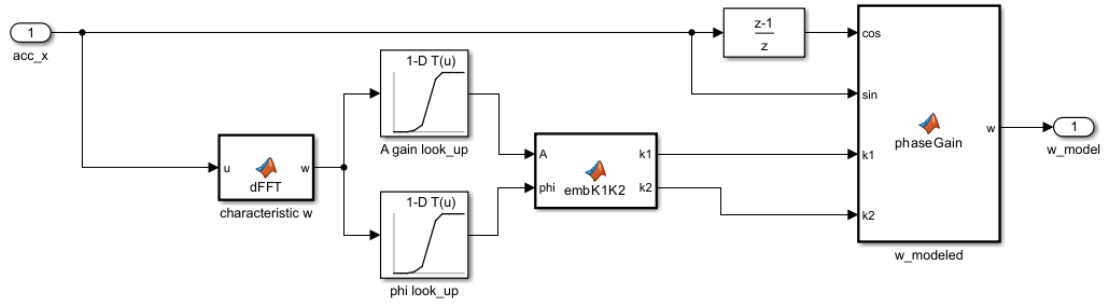

Figure 6.6: Basis of embedded Simulink model.

According to our previous work [33] we have extended the proposed algorithm to facilitate our latest findings, summarized in Figure 5.7 regarding the dependency of amplitude gain and phase shift to the dynamic disturbances to the model as described on the illustration 6.6.

To obtain the desired compensator functionality following pseudocode can be applied:

1. Determine the gain and phase shift from a look-up table, to apply.
2. Multiply original $\sin(2\pi ft)$ from accelerometer by A and $\cos(\varphi)$.
3. Numerically derive the original acceleration signal, preferably applying Savitzky-Golay filter [39].
4. Multiply the result of point 3. by A and $sin(\varphi)$.
5. Sum the results of steps 2 and 4, to get phase shifted signal as stated in Equation 6.2.

This way allows us to define a phase shift and amplitude by the sum of two complementary functions.

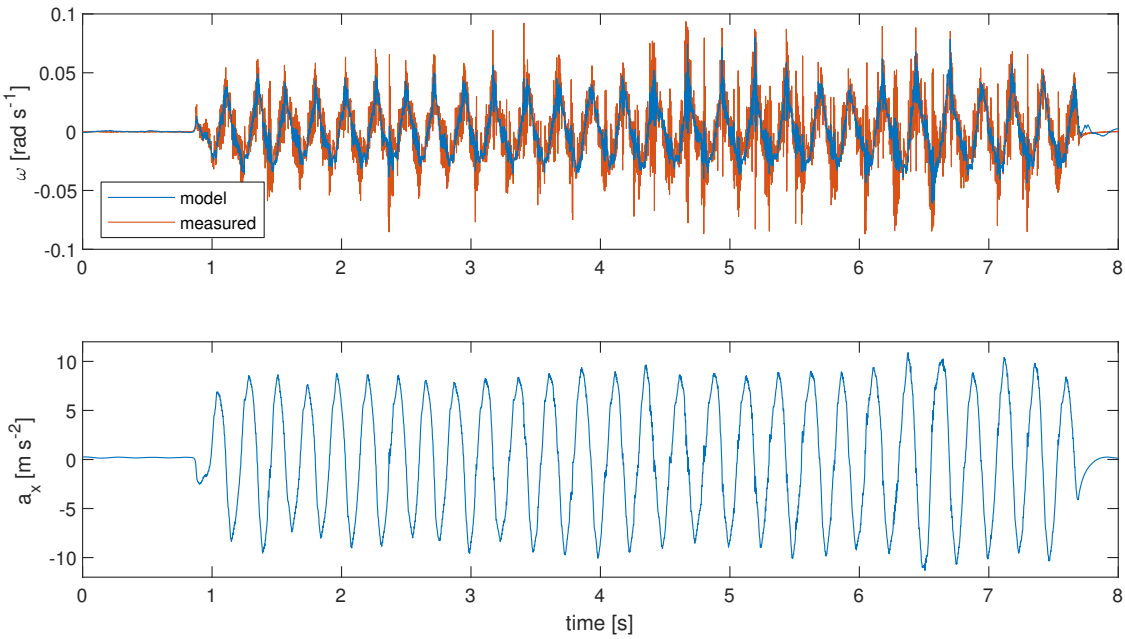

Figure 6.7: Measured and modelled gyroscope error.

Modeled ZRO could be used to compensate for the actual raw measurement of the gyroscope as defined:

$$\omega_{\text{compensated}}(t) = \omega_{\text{gyro}}(t) - y_{gyroZRO}(t) \tag{6.3}$$

The compensated measurement was read out from the microcontroller after processing with the following result. Illustrative performance can be seen in Figure 6.7 with compensator performance visible in Figure 6.8. Read-out values were processed by the microcontroller onboard in real-time.

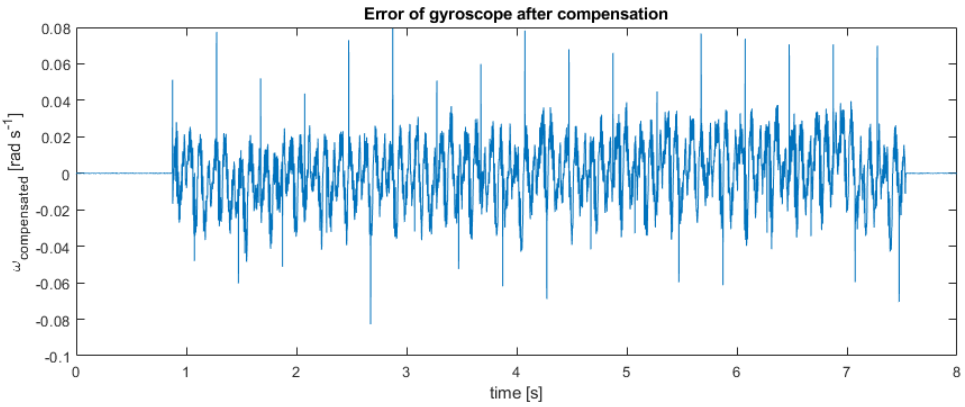

Figure 6.8: Time domain variation of gyroscope error.

Post-processing the data has identified the following performance of the algorithm summarized in Table 6.1.

	ω_{gyro}	$\omega_{compensated}$
mean $[rad/s]$	$1.4056e-16$	$1.566e-14$
standard deviation $[rad/s]$	0.1328	0.0151

Table 6.1: Compensator error comparison.

Upon analysis, we can ascertain that the mean value of the output remains zero so during the measurement no gyroscope drift was present. The computed values of the mean are markedly below the threshold of significance and can primarily be attributed to numerical errors stemming from post-processing, data handling, or acquisition methods. Of greater consequence is the performance of our algorithm in relation to the standard deviation of error, particularly under harmonic disturbances. Our observations demonstrate that the algorithm has achieved a substantial reduction in standard deviation—on the order of one decade. This result is indicative of the algorithm's efficacy in mitigating noise and enhancing the precision of the system when subjected to harmonic perturbations. These findings offer compelling evidence of the algorithm's robustness and constitute a significant advancement. The application of this type of compensator for non-harmonical movements shall be further evaluated.

6.4 Summary

The algorithms we have developed underwent an evaluation to assess their feasibility for real-time hardware implementation. Specifically, the non-linear least squares based algorithm was optimized to reduce computational load. By replacing non-linear functions with precomputed values stored in look-up tables, we significantly reduced the computational power required. This optimization enables the algorithm to be effectively deployed on low-cost, real-time hardware without compromising precision. On the other hand, the deployment of an efficient artificial neural network based compensator presents substantial resource demands. Consequently, its implementation on either a microcontroller or an FPGA is economically infeasible when contrasted with the cost of low-cost MEMS gyroscopes. To practically assess the performance of these compensators, we utilized a dsPIC microcontroller as our target hardware in Figure 6.3. The microcontroller was programmed using code automatically generated from the Matlab/Simulink toolchain. A detailed evaluation of the compensator's performance has been discussed in the preceding chapter.

7 Conclusion

The following objectives were set for this work:

1. Analysis of the effect of linear accelerations and jerks on MEMS gyroscopic sensors and their quantification.
2. Design of new models for linear acceleration compensation.
3. Implementation of the proposed method on real-time HW and experimental measurements.

The objectives have been met, i.e., to analyze the effect of acceleration and jerk, to design compensation models, and to implement the selected model on RT HW. Despite the specific functionality of the compensator, the work has many indirect results that have been achieved contextually, which is evident from the range of published papers.

Objectives 1 and 2 are very closely related in terms of application. The motivation for their solution was both engineering and pedagogical practice, which was independently initiated by the development of an unstable education model in the Mechatronics laboratory at the Brno University of Technology and from the perspective of the development of unstable electromechanical, dynamic systems requiring inertial sensors in cooperation with Hochschule Esslingen. The actual work carried out is described in Chapters 4, 5, and 6, while an introduction to the subject can be found in the research section 2. The first two objectives were met, but the applicability in practice is limited to the disturbance coming from harmonic signals and the delay associated with the dominant frequency extraction from discrete-time fast Fourier transformation. Still, this method can be considered as a founding stone and can be extended further or optimized. Objective 3 (Implementation of the proposed method on real-time HW and experimental measurements) proved to be more powerful than originally intended and with the potential for wider application beyond the field of gyroscopic sensors. Automatic code generation during the time of realization of this work has significantly advanced and proved to be a reliable workflow suitable not only for research purposes but also as a reliable industrial tool.

7.1 Thesis achievements

1. Based on the dynamic analysis, experimental methods were developed to evaluate the effect of linear accelerations and jerks on MEMS gyroscopic sensors. The evaluation includes an algorithm based on non-linear least squares, which

makes it possible to quantify the aforementioned influence in an exact way. Using the data thus obtained, it is possible to unambiguously compare all types of MEMS gyroscopic sensors based on the criterion of sensitivity to acceleration and jerk disturbances. The results are described in Chapter 4. The initial part of this method was published at [11].

2. The algorithm derived from the non-linear least square method has been simplified to create a parametric compensation model of a single-mass MEMS gyroscopic sensor with an emphasis on low computational requirements. This model can be applied on a microcontroller. As a result, it is possible to cost-optimize the computational HW and achieve a reduction of error in standard deviation by 10 times relative to the measurement without a compensator. This model is described in Section 6.3. This result was presented at the Mechatronics 2019, an international conference held in Warsaw, Poland [33], and cited in [40].

3. Development and validation of a non-linear compensation model based on an artificial neural network. Compared to the previous parametric model, this is a more computationally demanding model. Based on the results from our work [11], this type of compensator achieved a reduction of error standard deviation by 4 times relative to the measurement without a compensator. This model is described in section 5.2. The result was published at [11].

7.2 Further research possibilities

The method presented here to quantify the effect of linear acceleration on a gyroscopic sensor was defined with the intention of using it in the widest possible range of applications. However, its functionality was tested on sets of harmonic signals and an extension to more general random noise would be beneficial. In the literature [41], [42], or [43] we often come across the issue of the influence of constant normal and tangential acceleration. We consider it beneficial to develop and test the methodology and compensation algorithms for the case of varying rotational accelerations (or non-constant normal and tangential acceleration).

The proposed compensation algorithms have rather specific functionality directly aimed at application in the single-axis two-wheel self-balancing personal transporter. It would be advisable to focus on the generalization of each method in more detail. In the analysis of the individual gyroscopic sensors, we observed frequencies where the sensitivity to external excitation deviated significantly from the general trend, explicitly around 3.5 Hz. We suspect that these may be harmonics of the resonant frequency associated with the internal mechanical structures of the MEMS gyroscope. We recommend further investigation of this occurrence, incorporating the insights into an enhanced compensator.

List of Figures

List of Tables

References

[1] *1.Famous Foucault's Pendulum [2] — Download Scientific Diagram*. [N.d.]. Available also from: https://www.researchgate.net/figure/Famous-Foucaults-Pendulum-2_fig1_282152044/actions#reference.

[2] GREPL, R. Model Based Design of a Self-balancing Vehicle: A Mechatronic System Design Case Study. In: *Mechatronics 2013*. Cham: Springer International Publishing, 2014, pp. 869–876. Available from DOI: 10.1007/978-3-319-02294-9{_}110.

[3] *Návrh konstrukce, řízení a elektroniky pro nestabilní balancující vozidlo; Ing. František Zouhar (2010 - 33817) – VUT*. [N.d.]. Available also from: https://www.vut.cz/studenti/zav-prace/detail/33817.

[4] MAJID DADAFSHAR. APPLICATION NOTE 5830. *APPLICATION NOTE 5830*. 2014. Available also from: http://www.maximintegrated.com/en/an5830.

[5] ACAR, Cenk; SHKEL, Andrei. MEMS Vibratory Gyroscopes. 2009. ISBN 978-0-387-09535-6. Available from DOI: 10.1007/978-0-387-09536-3.

[6] WEINBERG, Marc S; BERNSTEIN, Jonathan J; CHO, Steve T; KING, A Thomas; KOUREPENIS, Anthony S; MACIEL, Paul. A micromachined comb-drive tuning fork rate gyroscope. *[1993] Proceedings IEEE Micro Electro Mechanical Systems*. 1993, pp. 143–148. Available also from: https://api.semanticscholar.org/CorpusID:111237742.

[7] *File:Fibre-optic-interferometer.svg - Wikipedia*. [N.d.]. Available also from: https://en.m.wikipedia.org/wiki/File:Fibre-optic-interferometer.svg.

[8] CHEN, Yun Jhih; HANSEN, Azure; HOTH, Gregory W.; IVANOV, Eugene; PELLE, Bruno; KITCHING, John; DONLEY, Elizabeth A. NIST's Compact Atomic Gyroscope Displays New Twists — NIST. *Physical Review Applied*. 2019, vol. 12, no. 1. ISSN 23317019. Available from DOI: 10.1103/PHYSREVAPPLIED.12.014019.

[9] LITTLE, Bethany J.; HOTH, Gregory W.; CHRISTENSEN, Justin; WALKER, Chuck; DE SMET, Dennis J.; BIEDERMANN, Grant W.; LEE, Jongmin; SCHWINDT, Peter D.D. A passively pumped vacuum package sustaining cold atoms for more than 200 days. *AVS Quantum Science*. 2021, vol. 3, no. 3. ISSN 26390213. Available from DOI: 10.1116/5.0053885/570595.

[10] *Estimation Techniques for Low-Cost Inertial Navigation – ScienceOpen*. [N.d.]. Available also from: https://www.scienceopen.com/document?vid=a7e1b6d6-2679-4891-ba64-86d1793bedab.

[11] SPACIL, T.; RAJCHL, M. Compensation of Linear Acceleration in Single-Mass MEMS Gyroscope. *Proceedings of the 2018 18th International Conference on Mechatronics - Mechatronika, ME 2018*. 2019, pp. 1–6. ISBN 9788021455443.

[12] LEE, Feng Yu; LIANG, Kai Chih; CHENG, Emerson; FANG, Weileun. Design and implementation of a fully-decoupled tuning fork (FDTF) MEMS vibratory gyroscope for robustness improvement. *2015 Transducers - 2015 18th International Conference on Solid-State Sensors, Actuators and Microsystems, TRANSDUCERS 2015*. 2015, pp. 1160–1163. ISBN 9781479989553. Available from DOI: 10.1109/TRANSDUCERS.2015.7181134.

[13] WEINBERG, H. Gyro Mechanical Performance: The Most Important Parameter. *Analog Devices Inc., Technical Article MS-2158. http:// ...* 2011, no. September, pp. 1–5. Available also from: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Gyro+Mechanical+Performance:+The+Most+Important+Parameter#0.

[14] INDEITSEV, D. A.; BELYAEV, Ya V.; LUKIN, A. V.; POPOV, I. A.; IGUMNOVA, V. S.; MOZHGOVA, N. V. Analysis of imperfections sensitivity and vibration immunity of MEMS vibrating wheel gyroscope. *Nonlinear Dynamics*. 2021, vol. 105, no. 2, pp. 1273–1296. ISSN 1573269X. Available from DOI: 10.1007/S11071-021-06664-0/FIGURES/34.

[15] FEI, Juntao; DAI, Weili; HUA, Mingang; XUE, Yuncan. System Dynamics and Adaptive Control of MEMS Gyroscope Sensor. *IFAC Proceedings Volumes*. 2011, vol. 44, no. 1, pp. 3551–3556. ISBN 9783902661937. ISSN 1474-6670. Available from DOI: 10.3182/20110828-6-IT-1002.00034.

[16] GAVCAR, Hasan Doğan. Compensation methos for quasi-static acceleration sensitivity of MEMS gyroscopes /. 2014. Available also from: https://open.metu.edu.tr/handle/11511/24200.

[17] SONMEZOGLU, S.; GAVCAR, H. D.; AZGIN, K.; ALPER, S. E.; AKIN, T. Simultaneous detection of linear and coriolis accelerations on a mode-matched MEMS gyroscope. In: *2014 IEEE 27th International Conference on Micro Electro Mechanical Systems (MEMS)*. IEEE, 2014, pp. 32–35. ISBN 978-1-4799-3509-3. Available from DOI: 10.1109/MEMSYS.2014.6765566.

[18] SONMEZOGLU, S.; GAVCAR, H. D.; AZGIN, K.; ALPER, S. E.; AKIN, T. Simultaneous detection of linear and coriolis accelerations on a mode-matched MEMS gyroscope. *Proceedings of the IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*. 2014, pp. 32–35. ISBN 9781479935086. ISSN 10846999. Available from DOI: 10.1109/MEMSYS.2014.6765566.

[19] BODNICKI, Maciej; LUCZAK, Sergiusz. Comments on "Delay Compensation of Tilt Sensors Based on MEMS Accelerometer Using Data Fusion Technique". *IEEE Sensors Journal*. 2018, vol. 18, no. 3, pp. 1333–1335. ISSN 1530-437X. Available from DOI: 10.1109/JSEN.2017.2767102.

[20] ŁUCZAK, Sergiusz; GREPL, Robert; BODNICKI, Maciej. Selection of MEMS Accelerometers for Tilt Measurements. *Journal of Sensors*. 2017, vol. 2017, pp. 1–13. ISSN 1687-725X. Available from DOI: 10.1155/2017/9796146.

[21] CRASSIDIS, John L.; LANDIS MARKLEY, F.; CHENG, Yang. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*. 2007, vol. 30, no. 1, pp. 12–28. ISSN 15333884. Available from DOI: 10.2514/1.22452.

[22] STADEN, Frederick van; NUTA, Roxana-Georgiana; TATU -, Georgiana-Luiza; NEMETH MACAMBIRA, Christian; GHEDINI DER AGOPIAN, Paula; ANTONIO MARTINO, Joao. Analysis of compensation for a g-sensitivity scale-factor error for a MEMS vibratory gyroscope. *Journal of Micromechanics and Microengineering*. 2015, vol. 25, no. 11, p. 115006. ISSN 0960-1317. Available from DOI: 10.1088/0960-1317/25/11/115006.

[23] *Least-Squares (Model Fitting) Algorithms - MATLAB & Simulink*. [N.d.]. Available also from: https://www.mathworks.com/help/optim/ug/least-squares-model-fitting-algorithms.html#f204.

[24] *Choosing the Algorithm - MATLAB & Simulink*. [N.d.]. Available also from: https://www.mathworks.com/help/optim/ug/choosing-the-algorithm.html.

[25] *Newton's method - Wikipedia*. [N.d.]. Available also from: https://en.wikipedia.org/wiki/Newton%27s_method.

[26] MORÉ, Jorge J. The Levenberg-Marquardt algorithm: Implementation and theory BT - Numerical Analysis. *Numerical Analysis*. 1978, vol. 630, no. Chapter 10, pp. 105–116. ISBN 978-3-540-08538-6. Available also from: http://www.springerlink.com/index/10.1007/BFb0067700%20file://localhost/Users/DongwonShin/Dropbox/Library.papers3/Books/1978/Mor%C3%A9/Numerical%20Analysis%201978%20Mor%C3%A9.pdf%20papers3://publication/doi/10.1007/BFb0067700.

[27] *Find global minimum - MATLAB*. [N.d.]. Available also from: https://www.mathworks.com/help/gads/globalsearch.html.

[28] *Find multiple local minima - MATLAB*. [N.d.]. Available also from: https://www.mathworks.com/help/gads/multistart.html?s_tid=doc_ta.

[29] NELLES, Oliver. *Book*. Nonlinear system identification: from classical approaches to neural networks and fuzzy models. 2001. ISBN 978-3-642-08674-8. ISSN 0196-4313. Available from DOI: 10.1007/978-3-662-04323-3.

[30] *Neuron Model - MATLAB & Simulink*. [N.d.]. Available also from: https://www.mathworks.com/help/deeplearning/ug/neuron-model.html.

[31] BRE, Facundo; GIMENEZ, Juan M.; FACHINOTTI, Víctor D. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*. 2018, vol. 158, pp. 1429–1441. ISSN 0378-7788. Available from DOI: 10.1016/J.ENBUILD.2017.11.045.

[32] *Code Generation: Run MATLAB Code and Simulink Models Anywhere! » Student Lounge - MATLAB & Simulink*. [N.d.]. Available also from: https://blogs.mathworks.com/student-lounge/2019/01/02/code-generation-online-tutorial/.

[33] SPACIL, T.; RAJCHL, M.; BASTL, M.; NAJMAN, J.; APPEL, M. *Advances in Intelligent Systems and Computing*. Vol. 1044, Design of deterministic model for compensation of acceleration sensitivity in MEMS gyroscope. 2020. ISBN 9783030299927. ISSN 21945365. Available from DOI: 10.1007/978-3-030-29993-4{_}35.

[34] LIU, Tong; HE, Chunhua; ZHAO, Qiancheng; YANG, Zhenchuan; YAN, Guizhen; JIN, Yufeng. Research on the Zero-Rate Output compensation for MEMS vibratory gyroscopes. *2017 IEEE 12th International Conference on Nano/Micro Engineered and Molecular Systems, NEMS 2017*. 2017, pp. 804–807. ISBN 9781509030590. Available from DOI: 10.1109/NEMS.2017.8017140.

[35] *MPU-6050 — TDK InvenSense*. [N.d.]. Available also from: https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/.

[36] *L3G4IS - MEMS motion sensor: three-axis digital output gyroscope for gaming and OIS - STMicroelectronics*. [N.d.]. Available also from: https://www.st.com/en/mems-and-sensors/l3g4is.html#documentation.

[37] *SCC2000 Series Combined Gyro Sensor and Accelerometer — Gyro Sensors — Murata Manufacturing Co., Ltd.* [N.d.]. Available also from: https://www.murata.com/en-global/products/sensor/gyro/overview/lineup/scc2000.

[38] maxon DC motor Permanent magnet DC motor with coreless winding. 2012. Available also from: www.maxonmotor.com/academy.

[39] SOVA, Václav; BRABLC, Martin; GREPL, Robert. FPGA Implementation of Multiplierless Low-Pass FIR Differentiator - IEEE Conference Publication. In: *PROCEEDINGS OF THE 2018 18TH INTERNATIONAL CONFERENCE ON MECHATRONICS - MECHATRONIKA*. Brno, 2018, pp. 382–386. Available also from: https://ieeexplore.ieee.org/document/8624664.

[40] INDEITSEV, D. A.; BELYAEV, Ya V.; LUKIN, A. V.; POPOV, I. A.; IGUMNOVA, V. S.; MOZHGOVA, N. V. Analysis of imperfections sensitivity and vibration immunity of MEMS vibrating wheel gyroscope. *Nonlinear Dynamics*. 2021, vol. 105, no. 2, pp. 1273–1296. ISSN 1573269X. Available from DOI: 10.1007/S11071-021-06664-0.

[41] SAUKOSKI, Mikko; AALTONEN, Lasse; HALONEN, Kari A.I. Zero-rate output and quadrature compensation in vibratory MEMS gyroscopes. *IEEE Sensors Journal*. 2007. ISBN 1530-437X. ISSN 1530437X. Available from DOI: 10.1109/JSEN.2007.908921.

[42] BANCROFT, Jared B.; LACHAPELLE, Gerard. Estimating MEMS gyroscope g-sensitivity errors in foot mounted navigation. In: *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service, UPINLBS 2012*. 2012. ISBN 9781467319096. Available from DOI: 10.1109/UPINLBS.2012.6409753.

[43] FAN, Chen; HU, Xiaoping; HE, Xiaofeng; TANG, Kanghua; LUO, Bing. Observability analysis of a MEMS INS/GPS integration system with gyroscope G-sensitivity errors. *Sensors (Switzerland)*. 2014. ISBN 8673184576. ISSN 14248220. Available from DOI: 10.3390/s140916003.

List of author's publications

[1] SPACIL, Tomas; RAJCHL, Matej. Compensation of Linear Acceleration in Single-Mass MEMS Gyroscope. In: *2018 18th International Conference on Mechatronics - Mechatronika (ME)*. 2018, pp. 1–6.

[2] SPACIL, Tomas; RAJCHL, Matej; BASTL, Michal; NAJMAN, Jan; APPEL, Martin. Design of Deterministic Model for Compensation of Acceleration Sensitivity in MEMS Gyroscope. *Advances in Intelligent Systems and Computing*. 2020, vol. 1044, pp. 285–291. ISSN 9783030299927. Available from DOI: 10.1007/978-3-030-29993-4{_}35.

[3] NAJMAN, Jan; BRABLC, Martin; RAJCHL, Matej; BASTL, Michal; SPÁČIL, Tomáš; APPEL, Martin. Monte Carlo Based Detection of Parameter Correlation in Simulation Models. *Advances in Intelligent Systems and Computing*. 2020, vol. 1044, pp. 54–61. ISSN 9783030299927. Available from DOI: 10.1007/978-3-030-29993-4{_}7.

[4] BASTL, Michal; SPACIL, Tomas; NAJMAN, Jan; CELIK, Mustafa; HANCIOĞLU, Oğuz Kaan; GREPL, Robert. Estimation of Maximum Signal Strength for Satellite Tracking Based on the Extended Kalman Filter. *Advances in Military Technology*. 2023, vol. 18, no. 1, pp. 87–99. Available from DOI: 10.3849/aimt.01725.

[5] BASTL, Michal; NAJMAN, Jan; SPÁČIL, Tomáš. Design of an Antenna Pedestal Stabilization Controller Based on Cascade Topology. In: 2020, pp. 469–475. ISBN 978-3-030-29992-7. Available from DOI: 10.1007/978-3-030-29993-4_58.

[6] DOBOSSY, Barnabás; FORMÁNEK, Martin; STASTNY, Petr; SPÁČIL, Tomáš. Fault Detection and Identification on Pneumatic Production Machine. In: *Modelling and Simulation for Autonomous Systems*. Cham: Springer International Publishing, 2022, pp. 39–60. ISBN 978-3-030-98259-1. Available from DOI: 10.1007/978-3-030-98260-7_3.