Politecnico di Milano

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
Corso di Laurea Magistrale in Ingegneria Aeronautica

Tesi di Laurea Magistrale

Design of a redundant, skewed MEMS sensor-based Attitude and Heading Reference System

Advisor
Prof. Alberto Luigi Michele Rolando

Co-Advisor
Alberto Brandl, PhD

Candidate
Marcello Parigi
Matr.905350

Academic Year 2020 – 2021

To my parents and my grandparents

Acknowledgments

La presente tesi segna la fine del mio percorso di studi. Un capitolo della mia vita che si chiude, lasciando spazio a nuove avventure. Si tratta dell'occasione perfetta per ripercorrere con il pensiero i momenti passati e ringraziare tutte le persone che mi sono state affianco in questo cammino, ricco di esperienze e soddisfazioni.

In primo luogo, ringrazio il Prof. Alberto Rolando per avermi proposto questa tesi molto interessante (la quale si è rivelata una preziosa fonte di apprendimento) e per tutto il tempo dedicatomi. La sua presenza, anche se virtuale, è stata infatti costante; questo, sfortunatamente, a causa alla pandemia di COVID-19 che ha fatto da sfondo a tutto il progetto. Un grazie sentito ad Alberto Brandl che mi ha seguito in modo impeccabile, sempre pronto a dispensare consigli utili. Senza il loro supporto, il lavoro sarebbe stato decisamente più arduo.

La seconda serie di ringraziamenti è destinata invece agli amici che mi hanno accompagnato durante il viaggio. In primis, ringrazio Michele, amico di una vita, che mi ha sempre ascoltato e supportato, anche nei momenti più complicati. È ora il turno di Paolo, conosciuto il primo anno tra i banchi del Politecnico e rimasto poi stabilmente al mio fianco. Un grazie ad Andrea, fedele compagno di avventure e chiacchierate stimolanti. Ulteriori ringraziamenti ai fratelli Giuseppe e Pietro, conosciuti sin dalla tenera età. Non posso certo dimenticare Dalila, amica classicista. Mille grazie infine a Simone, Gianluca, Marco, Matteo, Giacomo e Paolo per aver allietato le lezioni ed aver reso il percorso meno pesante. Tantissimi i momenti condivisi insieme all'interno ed al di fuori delle "mura" universitarie.

In conclusione, il più importante e sentito dei ringraziamenti è rivolto ai miei genitori Marzia e Piercarlo ed ai nonni Adriana e Marcelliano. Ai primi per avermi costantemente stimolato, consigliato e sostenuto in qualsiasi mia scelta. Ai secondi per la compagnia, gli insegnamenti ed il perenne interesse dimostrato. Un ultimo pensiero è rivolto al caro zio Antonio, esperto di laser; un uomo di scienza e di fede. Sei stato un grande esempio. Grazie di cuore a tutti!

Milano, Luglio 2021 M. P.

Abstract

Since the dawn of aviation, the accurate estimation of attitude and heading has been a crucial challenge; to this end, several technologies have been exploited over the years. Recently, an increasing interest in Unmanned Aerial Vehicles (UAVs) created the need for low-cost, lightweight and low-power consumption Attitude and Heading Reference Systems (AHRS), based on MEMS sensors. This is enabled by the remarkable advancement achieved in the field of the aforementioned devices.

The present thesis presents a viable approach to the design of a redundant AHRS, based on four 9-axis MEMS sensors placed on the faces of a tetrahedron. The choice of such configuration has two possible significant advantages: working with multiple skewed sensors (which implies a redundancy of measurements) and the potential capability to detect faults and isolate malfunctioning sensors. Attention is focused on the first benefit, while the latter is proposed for future developments. Finally, the feasibility is justified by the low cost of the four devices, which is consistent with the aim of the project.

Two major achievements can be highlighted: the design of a simulation environment that generates consistent sensor measurements and the successful implementation of an Unscented Kalman Filter (UKF) which provides stable and accurate estimates of the attitude. The non-linear filter delivers satisfactory performance in all the different test conditions, as long as the limitations given by Euler angles are met. Moreover, on the basis of the sensor model that is developed, the configuration that integrates four sensors proves to be more accurate than the equivalent single-sensor solution. In conclusion, size reduction and the adoption of less noisy sensors have beneficial effects on the quality of estimates.

Sommario

Sin dagli albori dell'aviazione, stimare l'assetto e la rotta di un velivolo in modo accurato ha costituito una sfida rilevante; a tal fine, sono state impiegate diverse tecnologie nel corso degli anni. Recentemente, un crescente interesse nei confronti dei velivoli a pilotaggio remoto ha creato la necessità di avvalersi di Attitude and Heading Reference Systems (AHRS), basati su sensori MEMS, leggeri, a basso costo e caratterizzati da un ridotto consumo energetico; tutto ciò è reso possibile grazie ai notevoli progressi compiuti nel campo di tali dispositivi.

La presente tesi ha l'obiettivo di proporre un possibile approccio alla progettazione di un AHRS, dotato di quattro sensori MEMS a 9 assi posizionati sulle facce di un tetraedro. La scelta di tale configurazione presenta due vantaggi fondamentali: il fatto di poter lavorare con più sensori non allineati (il che garantisce una ridondanza di misure) e la potenziale capacità di identificare malfunzionamenti, con la conseguente possibilità di isolare i sensori interessati. Viene posta particolare attenzione al primo aspetto, mentre il secondo viene lasciato agli sviluppi futuri. Infine, la fattibilità è giustificata dal basso costo dei quattro dispositivi, in pieno accordo con le finalità del progetto.

Sono stati raggiunti due traguardi significativi: la creazione di un ambiente simulativo in grado di generare misure coerenti dei sensori e l'implementazione (avvenuta con successo) di un Unscented Kalman Filter (UKF), che produce stime stabili e accurate dell'assetto. Il filtro non lineare fornisce delle prestazioni soddisfacenti in tutte le condizioni in cui è stato testato, a patto che vengano rispettate le limitazioni derivate dagli angoli di Eulero. Inoltre, basandosi sul modello dei sensori che è stato elaborato, la configurazione che integra i quattro sensori si rivela più accurata rispetto allo stesso filtro dotato di un singolo sensore. In conclusione, la riduzione delle dimensioni del sistema e l'adozione di sensori meno rumorosi apporta dei benefici sulla qualità delle stime.

Contents

List of Figures

List of Tables

Introduction

Since the origin of the aerospace industry, attitude and heading estimation has been an essential subject. Multiple applications, including airplanes, spacecrafts, satellites and even tactical missiles, required aforementioned data. Attitude is in fact crucial for control and monitoring purposes. At first, mechanical gyroscopes and accelerometers were used. Later, the two components were integrated into Inertial Measurement Units (IMU). The addition of a magnetometer and a processing system led to a further development, giving birth to Attitude and Heading Reference Systems (AHRS). A traditional AHRS is composed of inertial and magnetic sensors, namely, three accelerometers, three rate gyros and three magnetometers.

The main problem is that rate gyros are prone to bias and random-walk errors, resulting from the integration of wide-band noise. Hence, in the past, reliable AHRS required very expensive sensors; the only ones that could guarantee long term bias stability. The resulting cost limited such instruments to expensive applications [11]. However, in the last decade, the diffusion and improvements in Micro-Electro-Mechanical-Systems (MEMS) have opened up new frontiers in both civil and military applications. In particular, the reduced cost of these devices allowed them to be used in smartphones, robotics, navigation, virtual reality, bio-mechanics and personal health (wearable devices).

But it was the ever increasing interest in Unmanned Aerial Vehicles (UAVs) that primarily stimulated the development of low-cost, lightweight and low-power consumption AHRS, based on MEMS sensors [14]. In fact, in cost-sensitive applications, filtering techniques can be applied to bound the attitude error growth.

Figure 1: Example of a MEMS-based AHRS: Watson Industries' AHRS-S305®.

Rate gyros measure angular rates around their local reference axes. Thus, through appropriate rotations and by integration, Euler angles can be computed. Meanwhile, accelerometers and magnetometers also jointly provide Euler angle solutions according to gravity and magnetic measurements [44]. On these grounds, the main

idea behind the thesis project is to combine data from the different sensors with a non-linear Kalman filter, trying to improve the results.

It is worth specifying that a rate gyro is a type of gyroscope, which rather than indicating direction, provides the rate of change of angles with time (around its reference axes). In the literature and especially in the industry, the two terms are often used as synonyms. Also in the present thesis, it will be referred to rate gyros as gyroscopes, since the latter include the former.

Typically, MEMS sensors are mounted orthogonally such that the sensor frame axes coincide with the body frame. An example of a MEMS-based AHRS is presented in Figure 1. However, other redundant configurations can be explored.

Thesis objective

The goal of this thesis is to lay down the foundations for the development of a low-cost redundant MEMS sensor-based AHRS, with a possible upgrade to an Inertial Navigation System (INS). This product could meet the increasing demand in the market of UAVs, especially consumer ones. Instead of using a single 9-axis MEMS sensor, as in most available solutions, the idea is to implement a redundant system based on four (9-axis) devices. The feasibility is justified by the low cost of the four sensors, that are placed on the faces of a tetrahedron, as illustrated in Figure 2. For this reason, it will be referred to as *TetrAHRS*.

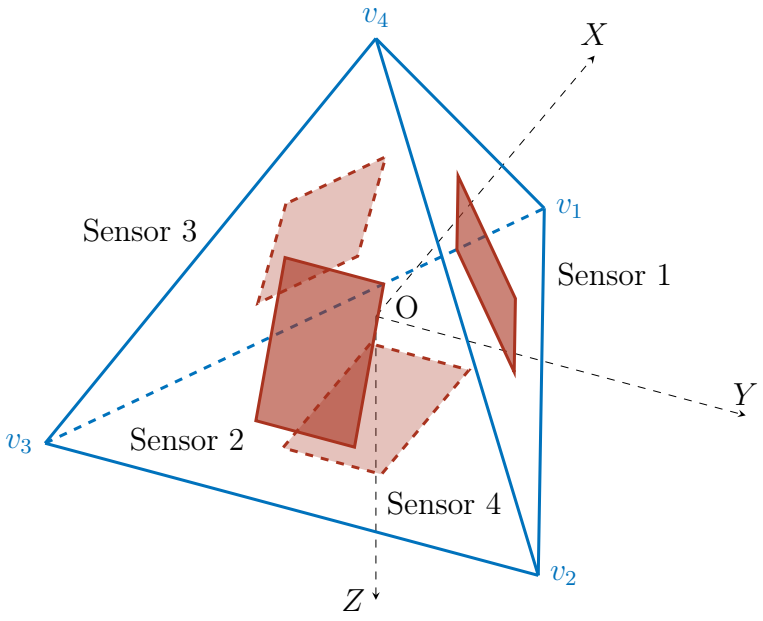


Figure 2: Positions of sensors on the tetrahedron.

The choice of a tetrahedron allows to work with skewed sensors since the axes are not aligned. Hence, the aim of the research is to find out if this configuration, together with the redundancy of measurements, could improve the quality of results.

Another significant advantage is the potential capability to *detect faults* and *isolate* them. In fact, having more than one sensor means that multiple solutions (relative to the same time instant) can be computed, considering different subsets

of MEMS. An algorithm could be implemented to compare the results and look for inconsistencies. The analysis of the solutions, correlated to the corresponding subset, should allow to determine the faulty sensor and isolate it. The AHRS should indeed work properly with a reduced number of MEMS devices. The idea could be taken into consideration for possible future developments of the project.

A mathematical model is created to simulate measurements obtained from *TetrAHRS*, as it moves along an arbitrary path. The second task is to develop a non-linear Kalman filter that allows to combine outputs coming from different sensors. Finally, simulated data are used as inputs for the filter in order to test its behavior.

Regarding the experimental studies, an analysis of measurements obtained from a real-world sensor is performed. This allows to assess parameters for the simulation platform, in particular related to noise. A thorough sensor characterization is required to refine the model of the sensors, which can be improved by adding more parameters. Unfortunately, experimental tests were severely limited by the COVID-19 pandemic that kept on raging during the whole course of the project.

In conclusion, a suggested strategy could be to build a "large scale" structure (with an approximated size of 10-20 cm) in order to perform some initial tests and get accustomed to the configuration. Subsequently, the path to miniaturization and integration with other systems is pursued. This way, the device could become suitable for many applications (especially UAVs) where small sizes are desirable.

Review of past literature

The idea of exploiting MEMS sensors in order to lower the price and the size of AHRS and INS is not new. Over the years, a large number of products have been developed and countless papers have been written on the subject.

One of the latest article that was found in the literature offers a new perspective on low-cost MEMS-Based AHRS [27]. It proposes a solution to integrate sensor data with a complementary adaptive Kalman filter, adding an innovative initialization method to improve the accuracy of the initial attitude estimates.

Similarly, Yangzhu Wang and his colleagues (in Reference [44]) suggest the implementation of complementary filtering (CF) in order to combine sensor measurements. Information fusion is performed with Euler angles directly, since they are more straightforward for understanding and visualizing. Moreover, an effective discontinuity avoiding strategy is proposed to refine the estimation. To make up for shortcomings of MEMS sensors, such as multiple errors, drifts and bad accuracy, some effective calibration is performed and filtering algorithms are presented to guarantee agreeable AHRS performance.

On the contrary, the introduction of sensor redundancy is a more unconventional path, in particular considering a tetrahedral arrangement. In year 2000, the idea of placing MEMS sensors on a truncated tetrahedron was advanced by Salah Sukkarieh and his co-authors in Reference [41]. Four ceramic vibrating structure gyroscopes and four QLC 400 accelerometers were set on the support. This configuration

provides for a theoretical 33 % increase in information and also allows for fault detection, which is required in many autonomous applications.

Another article (Reference [31]) took into consideration a tetrahedron for holding gyroscopes into position. It presents a calibration procedure applied to an inertial measurement unit composed of a triad of accelerometers put in the center of the structure and four gyros in a tetrad configuration.

Remarkable advances in MEMS technology allow to extend the concept by utilizing four 9-axis sensors. As a result, an higher number of measurements is available with better accuracy and stability, lower size and cost, compared to the past. This makes the solution even more interesting.

State of the art

Several typologies of AHRS are available on the market. They are based on different technologies, specifically for what concerns the gyroscopes. As explained in Section 1.1, the rate gyro is the most important component within the device and as such it makes a substantial difference in terms of performance. Mechanical gyros, Ring Laser Gyroscopes (RLG), Fiber Optic Gyroscopes (FOG) and MEMS gyros are mainly utilized. Mechanical sensors were progressively replaced by FOG, which are now competing with the MEMS-based AHRS to prevail in the market [28].

High-performance devices guarantee enhanced accuracy and stability but, due to their high price, they are not suitable for consumer products, e.g., UAVs. For this reason, the analysis focuses on MEMS-based solutions. The objective is not to rank the instruments in terms of performance or price but to get an overview of the main characteristics offered by some products. Out of the wide variety, two AHRS are reported as examples.

Watson Industries' AHRS-S305® ([19]) exploits three solid state MEMS angular rate gyros, three accelerometers and a triaxial fluxgate magnetometer. This choice makes the AHRS-S305 more economical and half the size of other products. The signals from three solid state gyros are coordinate transformed and then integrated to produce attitude and heading outputs. Data are compared to the accelerometers and the fluxgate magnetometer to derive the gyro drift error, which is used to adjust system biases.

Figure 3: POLAR-300 from UAV Navigation®.

The POLAR-300 from UAV Navigation® ([29]) is a high-end, MEMS-based Air Data and Attitude and Heading Reference System (ADAHRS) and Inertial Navigation

System (INS). It is perfect for system integration in avionics packages or other attitude sensing applications. Redundancy built into the POLAR-300 software allows it to survive individual sensor failures while maintaining accurate estimates of attitude and position. Its high-performance MEMS-based IMU is calibrated and compensated over the full industrial temperature range.

Structure of the thesis

The present thesis is organized in the following way:

- Chapter 1 provides an overview of the topics that will be touched in the treatise. These include the working principles of an AHRS and the components that are integrated in them, with a particular focus on MEMS sensors. The last part is dedicated to mathematical tools that allow to represent and transform spatial quantities (in particular, Direction Cosine Matrices and Euler angles).

- Chapter 2 describes the simulation environment that allows to generate consistent sensor measurements, based on the motion of *TetrAHRS*. An arbitrary trajectory is prescribed to the system and sensor outputs are computed accordingly. Subsequently, they are fed into the Kalman filter.

- Chapter 3 is dedicated to Kalman filtering. The concepts behind it are explained, beginning with the linear case and then moving to non-linear filters. The analysis of the framework, that was developed to tackle the problem, leads to the choice of an Unscented Kalman filter (UKF), which is described in details. At last, alternatives to the UKF are proposed.

- Chapter 4 outlines the procedures that were followed in order to perform static data acquisitions with a 9-axis MEMS sensor. Statistical properties are calculated and examined.

- Chapter 5 collects the outcomes of the simulation process. The performance of the filter is assessed in different test conditions, showing satisfying results. Finally, sensitivity analyses and a comparison with alternative non-linear filters are carried out.

Chapter 1

Background

This Chapter aims at providing a detailed overview of relevant topics that will be integral part of the dissertation. Firstly, a description of an AHRS is proposed, specifying the difference from IMU and INS. It follows a review of their component parts: accelerometers, gyroscopes and magnetometers. A particular attention is dedicated to MEMS sensors. Finally, mathematical tools that allow to represent and transform spatial quantities are presented.

1.1 Attitude and Heading Reference System

Before introducing the Attitude and Heading Reference System (AHRS), the Inertial Measurement Unit (IMU) shall be defined. It is an electronic device that measures and reports accelerations (including the gravitational one) and angular rates. Generally, it is composed of three accelerometers and three gyroscopes. Most of IMUs, in particular recent ones, also include three magnetometers, so they are able to detect the orientation of Earth's magnetic field with respect to sensor axes. They are essential components in the guidance and control of unmanned systems, such as UAVs.

An AHRS adds a central processing unit (CPU) to the IMU. Thus, the main difference is that an IMU delivers sensor data while an AHRS embeds a processing system (a Kalman Filter for instance) that provides attitude and heading information. They are sometimes referred to as MARG (Magnetic, Angular Rate, and Gravity) and consist of either solid-state or MEMS sensors [25]. They are designed to replace traditional mechanical gyroscopic flight instruments.

Lastly, Inertial Navigation Systems (INS) are composed of an IMU and additionally embed a GPS/GNSS receiver. An INS fuses inertial, navigation, and aiding data (odometer, DVL, etc.) thanks to an algorithm, such as the Extended Kalman Filter. This GNSS and IMU sensor fusion provides roll, pitch, heading, position, and velocity [42].

Some AHRS can be connected to a GNSS receiver or other external aids to improve their performances; however, this thesis considers a device that is comprised of accelerometers, gyroscopes and magnetometers only. The gyroscope is the most important sensor within the instrument. In an ideal world, it could be sufficient for a

fully-functional AHRS, with no need of other sensors. That is due to the fact that a gyro provides angular rates around its sensitivity axes; hence, under the assumption that the initial attitude is known, its measurements can be integrated over time in order to obtain the attitude in the desired instant. Unfortunately, gyroscopes are subjected to measurements errors, that include noise, biases and other behaviors associated with the type of sensor. The integration of aforementioned errors means that the orientation will drift over time.

There arises the need to compensate for the drift of the solution using alternative measurements of orientation. Accelerometers and magnetometers come into play but they have drawbacks too. An accelerometer is able to provide the direction of the gravitational acceleration vector but the measurement is corrupted by motion and vibrations. Similarly, a magnetometer can sense Earth's magnetic field but it may become unreliable in particular conditions [24].

Therefore, the goal of the AHRS fusion algorithm is to compensate for sensor faults, trying to improve the quality of results. Focusing on the problem being studied, four 9-axis sensors are involved. Hence, the implementation of a Kalman filter should be a viable and efficient way.

1.2 Accelerometers, gyroscopes and magnetometers

As explained in the previous Section, AHRS are based on accelerometers, gyroscopes and magnetometers. The three devices will be examined individually.

1.2.1 Accelerometers

An accelerometer is an electronic sensor that provides measurements of acceleration, including the contribution of gravity (more details are reported in Section 2.1.5). As described in Reference [32], there are mainly three types of accelerometers:

- *Piezoelectric*: take advantage of the properties of homonymous materials to produce electricity when put under physical stress. They include a sensing crystal to which a seismic weight is attached. When the sensor experiences acceleration, the weight exerts a force on the crystal. The piezoelectric crystal converts the force acting upon it to electrical signals, which can be measured to find the acceleration. This type of accelerometer is most commonly used in vibration and shock measurement.

- *Piezoresistive*: exploit the increase in their resistance based on the acceleration they experience. They are less sensitive than piezoelectric accelerometers and they are better suited to vehicle crash testing.

- *Capacitive*: use the change in electrical capacitance to determine an object's acceleration. They are comprised of two capacitive plates and a diaphragm. As the device experiences acceleration, the diaphragm moves, changing the distance between the capacitive plates. MEMS accelerometers are based on this architecture, built on a micrometer scale.

The thesis project involves the use of three-axis MEMS accelerometers; this means that the sensing mass has to be able to translate vertically. This requires a more complex mechanical structure and as a result, the z performance on triple-axis accelerometer is often worse than that of x and y [24].

MEMS accelerometers suffer from measurements errors, which have to be examined and quantified. An error model is proposed in an article from Sandia National Laboratories [22]. It takes into account several contributions, including: bias (zero offset), scale factor (gain), non-linearity, axes misalignment, cross-axis sensitivity and effects related to temperature.

The values of each component can be assessed through an appropriate characterization of the sensor. This procedure will be part of future developments.

1.2.2 Gyroscopes

Gyroscopes are devices mounted on a frame that are able to sense an angular velocity (if the frame is rotating). Reference [33] offers a detailed analysis of different types of gyroscopes:

1. *Mechanical*: consist of a spinning mass that rotates around its axis. According to the gyroscopic effect, it tends to remain parallel to itself and oppose any attempt to change its orientation. If a gyroscope is installed on gimbals that allow the mass to navigate freely in the free directions of space, its spinning axis will remain oriented in the same direction. When an angular speed is imposed to the gimbal (that supports the spinning mass), a proportional torque appears along a direction that is perpendicular to the direction of the imposed angular speed. If a spring system of known stiffness is opposing and balancing the output torque, it is possible to compute the imposed angular quantity. Thus, a mechanical gyroscope requires: a spinning wheel mounted on two gimbals, a rigid frame with rotating bearings (specifically designed to minimize friction) and a sensing system (to transduce the angular displacements into electric signals).

2. *Optical*: operate on the basis of the Sagnac effect. They are in fact able to sense the difference in propagation time between counter-propagating beams traveling in opposite directions in closed or open optical paths. The basic operating principle states that a rotation-induced change in the path lengths generates a phase difference between the counter-propagating light beams. Currently, two main typologies are used; they are based on two different types of architecture:

   – Ring Laser Gyroscopes (RLG) present an active configuration. This means that the closed-loop optical path (i.e., the ring cavity) contains the optical source, forming a ring laser.

   – Interferometric Fiber Optic Gyroscopes (IFOG) are characterized by a passive architecture. Such definition is due to the fact that the optical source is external to the closed loop (i.e., a fiber coil).

The two categories differ in terms of size, weight, power requirements, performance and cost. RLGs cover the high-performance market but their limiting factors are size, weight and high power consumption. On the other hand, different solutions have been proposed to improve FOGs performance, in particular Resonant Micro Optic Gyroscopes (RMOG). They allow to obtain good performance with smaller sizes, compared to FOGs. However, for all kinds of gyroscopes, bias drift is one of the most limiting factors.

3. *MEMS*: generally use a vibrating mechanical element to detect the angular velocity. They do not have rotating parts that require bearings; this allows an easy miniaturization. When rotated, the Coriolis effect causes a secondary vibration to the sensing element in a direction that is orthogonal to the imposed vibration. The active vibration means that gyroscopes will require more power than accelerometers [24]. Anyway, alternative technologies to improve performances are under development.

Five critical parameters involving gyroscopes can be pointed out:

1. Angle Random Walk (ARW): describes the error resulting from a broadband white noise element that characterizes the output of a gyro. Active elements are the major contributors to random noise. Noise is in fact one of the most important differences between optical and MEMS gyroscopes.

2. Bias Offset Error: when input rotation is null, the output of the gyro could be non-zero. Since it is a fixed error, it can be easily corrected.

3. Bias Instability: the initial zero reading of the gyroscope will cause drift over time due to integration of inherent imperfections and noise within the device. Integrating constant bias instability will cause angular errors. These errors will accumulate, so gyroscope-based rotations and angle estimates drift over the long-term [1]. For this reason, it is one of the most critical factors in the gyro selection process for applications that requires excellent accuracy over long time.

4. Temperature Sensitivity: performance changes over temperature. A characterization of parameters such as noise, bias offset and scale factor over temperature is necessary to verify that a gyro meets the targets of the system.

5. Shock and Vibration Sensitivity: noise and bias offset of gyros degrade under vibration and shock input. Considerable attention must be paid to vibrations, which can be substantial in some applications.

The gyroscope drift is mainly due to the integration of two components: bias instability (slow changing) and ARW (higher frequency). These parameters are measured in degrees of rotation per unit of time (typically [deg/h]). The yaw axis is generally the most sensitive to drift [1]. Table 1.1, taken from Reference [33], allows to compare the different types of gyros in terms of bias stability. The low-cost of MEMS sensors justifies their applications into consumer electronics. However, advances in the near future will probably make them erode part of the market that is now dominated by optical and mechanical instruments.

Table 1.1: Comparison among different types of gyroscopes in terms of bias stability and applications.

Performance Grade	Bias Stability	Applications	Technology
Consumer	30-1000 deg/h	Motion interface	MEMS
Industrial and Low-end Tactical	1-30 deg/h	Ammunitions and rockets guidance	MEMS
Tactical	0.1-30 deg/h	Platform stabilization	FOG/RLG
High-end Tactical	0.1-1 deg/h	Missile navigation	RLG/FOG
Navigation	0.01-0.1 deg/h	Aeronautics navigation	RLG/FOG
Strategic	0.0001-0.01 deg/h	Submarine navigation	RLG/FOG

1.2.3 Magnetometers

Magnetometers, or magnetic field sensors, are devices for detecting and measuring magnetic fields. The techniques used to produce magnetic sensors involve many aspects of physics and electronics. Reference [16] provides an overview of different technologies that are utilized:

1. *Superconducting Quantum Interference Devices* (SQUID): are among the most sensitive sensors, achieving a magnetic field resolution (minimum detectable magnetic field) on the order of several fT. They are used mainly in neuromagnetism, magnetic resonance and geology applications. Negative aspects are: high price, low working-temperature and high sensitivity to electromagnetic interference.

2. *Hall-effect* based sensors: allow to measure linear position, angular position, velocity and rotational speed. They are low cost but have intrinsic limits that affect their sensitivity and resolution due to the use of silicon.

3. *Fluxgate* sensors: measure the static or low frequency magnetic field. They are the most widely used sensors for compass navigation systems. However, their drawbacks include a complex fabrication of the magnetic core and the coils, as well as high mass and power consumption. Any reduction of their mass and power negatively affects both their sensitivity and stability; this opposes to miniaturization.

4. *Anisotropic Magnetoresistive* (AMR) sensors: are based on the anisotropic magnetoresistive effect that occurs in ferromagnetic transition metals; their electrical resistance depends on the angle between the electrical current and the direction of magnetization. They have low sensitivity to mechanical stress and a low power consumption.

5. *Giant magnetoresistive* (GMR) sensors have a large shift in the electrical resistance when their thin layers (a few nanometers) of ferromagnetic and non-magnetic materials are exposed to a magnetic field. They are characterized by higher offset and sensitivity temperature dependence compared to AMR sensors.

6. *Fiber optic* sensors: exploit the magnetostrictive effect, whereby the dimensions of the magnetostrictive material change when it is placed inside an external magnetic field. The material is bonded over a piece of optical fiber that, combined with an interferometer, allows to measure the strain of the fiber. Both temperature and pressure shifts affect the operation of this type of sensors.

Recently, improvements in MEMS technology has led to the development of sensors that present several advantages, such as small size, light weight, low-power consumption, minimum cost, high functionality, better sensitivity and resolution.

Most MEMS magnetic field sensors exploit the Lorentz force principle, according to which a force (generated by the magnetic field) causes the displacement of a resonant structure. The latter can be measured with optical, piezoresistive and capacitive sensing techniques. To this end, they use structures that are excited at their resonant frequencies by electrostatic forces or Lorentz forces. Consider, for instance, a clamped-clamped beam with an aluminum loop placed on it. The setup is illustrated in Figure 1.1, taken from Reference [16].

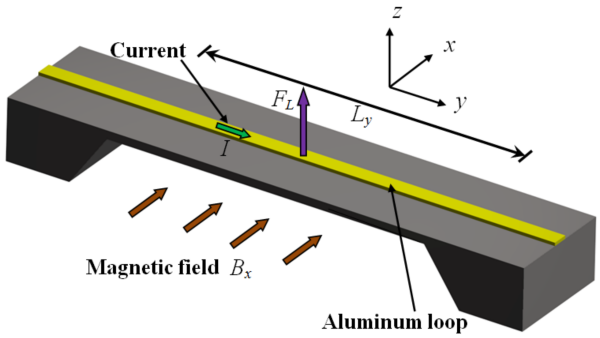


Figure 1.1: Schematic view of the Lorentz force acting on a clamped-clamped beam.

An excitation current (I) flows inside the loop with a frequency equal to the first beam-resonant frequency. When the structure is exposed to an external magnetic field (B_x) in the x-direction, then a Lorentz force (F_L) is generated. This force can be determined as:

$$F_L = I B_x L_y \tag{1.1}$$

where L_y is the length of aluminum loop perpendicular to the magnetic field. The force causes an amplified deflection on the midpoint, whose magnitude is directly proportional to I and B_x.

An important parameter in the performance of resonant structures is the *damping effect*, which limits the maximum amplitude of the structure. The damping level is determined using a quality factor Q, which measures the amount of losses during the operation of the resonant structure:

$$Q = 2\pi \frac{E_M}{E_c} \tag{1.2}$$

where E_M is the total energy stored in the structure and E_C is the energy factor lost per cycle due to the damping effect. An high quality factor can improve the

performance and resolution of the resonator. It also implies a low sensitivity to mechanical disturbances from its surroundings (e.g., mechanical vibrations).

Generally, the quality factor depends on the energy dissipated to a surrounding fluid, the energy coupled through the structure's support to a surrounding solid and the energy lost internally within the structure's material.

Temperature influences the performance of the sensor since it affects material properties (e.g., the elastic modulus, thermal expansion coefficient and thermal conductivity) and originates stresses inside of the resonant structure. Thus, the temperature stability is a critical issue during the operation and it must be controlled through compensation electronic circuits.

The displacement of the resonant structure can be measured in three ways:

- *Piezoresistive Sensing*: is based on active and passive piezoresistors (often integrated with a Wheatstone bridge). This architecture involves simple readout circuits while presenting high sensitivity and low manufacturing cost. Nevertheless, sensors need a careful packaging, trimming and compensation circuits for reducing the effect of thermal fluctuations on their performance.
- *Optical Sensing*: exploits the ability to measure the structural deflection on the basis of light reflections or through optical fibers. Sensors benefit from an immunity to EMI as well as a reduction in their electronic circuitry and weight. However, they present some problems due to the intrinsic losses of the structural imperfections of sensors and they can require complex fabrication processes.
- *Capacitive Sensing*: involves a readout system based on capacitance variations. Devices present a little temperature dependence but they suffers from parasitic capacitances in the connecting leads. The parasitic capacitances can be decreased whether electronic circuits are fabricated on the same substrate from the magnetic field sensor. In addition, sensors require complicated electronics and vacuum packagings.

MEMS devices need more reliability research in order to predict their performance in different environments and operation conditions. Future works should consider the reduction of damping and electronic noise, as well as the integration of different sensors on a single chip for monitoring several physical parameters. More details can be found in References [16] and [17].

1.3 Representation and transformation of spacial quantities

The description of orientation and transformations of spacial quantities are crucial to this study. Direction Cosine Matrices (DCM) are introduced at first. They allow to transform coordinates of a vector from a reference system into another. Properties of aforementioned matrices are reported. Secondly, the difference between *passive* and *active* rotations is enucleated. Finally, Euler angles are described and analyzed.

1.3.1 Direction Cosine Matrix (DCM)

A Direction Cosine Matrix (DCM), or simply a rotation matrix, is 3×3 matrix that transforms the coordinates of an arbitrary vector $\mathbf{v}$ from a reference system A into a rotated system B. Reference [37] provides an insight into the working principles of these matrices.

Consider an orthogonal right-handed coordinate reference system (O, x, y, z), denoted by letter A. Vector $\mathbf{v}$ is expressed by the following coordinates:

$$\mathbf{v}_A = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} \tag{1.3}$$

A right-handed rotation ϑ of the reference frame A is performed around z axis; as a result, a new rotated coordinate frame B (O, x', y', z) is created. The rotation is represented in Figure 1.2. The same vector $\mathbf{v}$ can be expressed in terms of the reference frame B. This is made possible by using a rotation matrix $\mathbf{R}^z_{A\to B}$:

$$\mathbf{v}_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} \cos\vartheta & \sin\vartheta & 0 \\ -\sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \mathbf{R}^z_{A\to B}\,\mathbf{v}_A \tag{1.4}$$

Note that throughout the whole treatise left-multiplying matrices acting on column vectors will be taken into consideration.

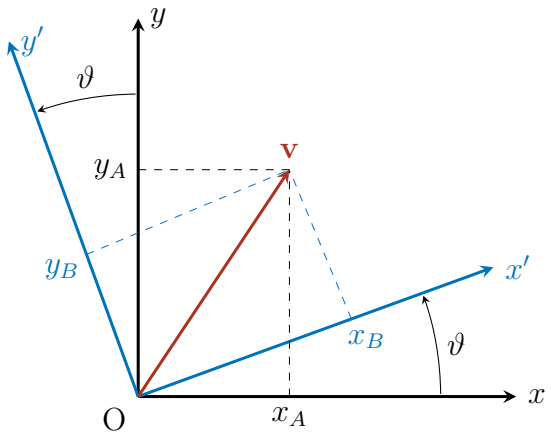


Figure 1.2: Right-handed rotation ϑ of the reference frame A around z axis.

The DCM is so called because its elements are direction cosines between corresponding axes of the new and old coordinate system. Let $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$, with appropriate subscripts, be unit vectors defining the axes of reference frames A and B. The x_B-component of the arbitrary vector $\mathbf{v}$ can be written as:

$$\begin{aligned} x_B &= (\mathbf{v}\cdot\mathbf{i}_B)^B = (\mathbf{v}\cdot\mathbf{i}_B)^A = \\ &= x_A(\mathbf{i}_A\cdot\mathbf{i}_B) + y_A(\mathbf{j}_A\cdot\mathbf{i}_B) + z_A(\mathbf{k}_A\cdot\mathbf{i}_B) \end{aligned} \tag{1.5}$$

This equation identifies the first row of the DCM. Similarly, the other components of $\mathbf{v}_B$ can be computed following the same logic. Thus, the rotation matrix elements are defined in the following way:

$$\mathbf{R}_{A\to B} = \begin{bmatrix} \mathbf{i}_A \cdot \mathbf{i}_B & \mathbf{j}_A \cdot \mathbf{i}_B & \mathbf{k}_A \cdot \mathbf{i}_B \\ \mathbf{i}_A \cdot \mathbf{j}_B & \mathbf{j}_A \cdot \mathbf{j}_B & \mathbf{k}_A \cdot \mathbf{j}_B \\ \mathbf{i}_A \cdot \mathbf{k}_B & \mathbf{j}_A \cdot \mathbf{k}_B & \mathbf{k}_A \cdot \mathbf{k}_B \end{bmatrix} \tag{1.6}$$

recalling that dot products of unit vectors are equivalent to direction cosines.

Eq. (1.6) allows to obtain the DCMs that represent right-handed rotations ϑ around the other two axes, respectively x-axis and y-axis:

$$\mathbf{R}^x_{A\to B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\vartheta & \sin\vartheta \\ 0 & -\sin\vartheta & \cos\vartheta \end{bmatrix} \tag{1.7}$$

$$\mathbf{R}^y_{A\to B} = \begin{bmatrix} \cos\vartheta & 0 & -\sin\vartheta \\ 0 & 1 & 0 \\ \sin\vartheta & 0 & \cos\vartheta \end{bmatrix} \tag{1.8}$$

Properties of DCMs (Rotation matrices)

A coordinate rotation must leave the length of a vector unchanged. The change of length under the rotation that was previously considered is given by:

$$\begin{aligned} \|\mathbf{v}\|^2 &= (\mathbf{v}_B)^T(\mathbf{v}_B) = (\mathbf{R}_{A\to B}\mathbf{v}_A)^T(\mathbf{R}_{A\to B}\mathbf{v}_A) \\ &= (\mathbf{v}_A)^T\,\mathbf{R}^T_{A\to B}\,\mathbf{R}_{A\to B}\,(\mathbf{v}_A) \end{aligned} \tag{1.9}$$

Hence, in order to preserve the aforementioned quantity, the following condition shall be verified:

$$\mathbf{R}^T_{A\to B}\,\mathbf{R}_{A\to B} = \mathbf{I} = \mathbf{R}_{A\to B}\,\mathbf{R}^T_{A\to B} \tag{1.10}$$

This last expression is the definition of an *orthogonal* matrix. Such property makes the inverse matrix easier to compute:

$$\mathbf{R}^{-1}_{A\to B} = \mathbf{R}^T_{A\to B} \tag{1.11}$$

Orthogonality also implies that the columns and the rows of the rotation matrix form an orthonormal set of vectors.

A vector can be expressed in a new coordinate reference system that is obtained from a sequence of rotations:

$$\mathbf{v}_D = \mathbf{R}_{C\to D}\,\mathbf{R}_{B\to C}\,\mathbf{R}_{A\to B}\,\mathbf{v}_A = \mathbf{R}_{A\to D}\,\mathbf{v}_A \tag{1.12}$$

Exploiting Eq. (1.11), the inverse operation is defined as:

$$\begin{aligned} \mathbf{v}_A &= \mathbf{R}^{-1}_{A\to D}\,\mathbf{v}_D = (\mathbf{R}_{C\to D}\,\mathbf{R}_{B\to C}\,\mathbf{R}_{A\to B})^{-1}\,\mathbf{v}_D \\ &= \mathbf{R}^{-1}_{C\to D}\,\mathbf{R}^{-1}_{B\to C}\,\mathbf{R}^{-1}_{A\to B}\,\mathbf{v}_D = \mathbf{R}^T_{C\to D}\,\mathbf{R}^T_{B\to C}\,\mathbf{R}^T_{A\to B}\,\mathbf{v}_D \\ &= (\mathbf{R}_{C\to D}\,\mathbf{R}_{B\to C}\,\mathbf{R}_{A\to B})^T\,\mathbf{v}_D = \mathbf{R}^T_{A\to D}\,\mathbf{v}_D \end{aligned} \tag{1.13}$$

Particular attention must be paid to the order of multiplication since rotation matrices are not commutative:

$$\mathbf{R}_{B\to C}\,\mathbf{R}_{A\to B} \neq \mathbf{R}_{A\to B}\,\mathbf{R}_{B\to C} \tag{1.14}$$

The last property states that the determinant of a DCM is unity and a non-trivial DCM has one, and only one, eigenvalue equal to unity. More details are reported in Reference [37].

1.3.2 Difference between passive and active rotations

The rotation matrices introduced in the previous Section represent *passive* rotations. This definition allows to distinguish them from *active* ones. These two types of rotations are profusely invoked in countless scientific disciplines and it is important not to confuse them. Reference [35] sheds light on the topic.

Passive rotations

A *passive* rotation indicates a positive counter-clockwise (according to the sign convention that was chosen) rotation around any of the coordinate axes of the frame (i.e., frame rotation) while the vectors referring to this frame remain fixed in space. In short, the axes rotates while vectors remain static. Consequently, once the frame has been rotated, an arbitrary vector will be identified by two sets of coordinates: one set referring to the original frame (before the rotation took place) and a second set associated with the frame that resulted after the rotation was performed.

The DCM matrices presented in Eq. (1.4), Eq. (1.7) and Eq. (1.8) allow to move from one set of coordinates to the other. The definition of *passive* rotation is coherent with the fact that the vector in question is not rotated; in fact, a coordinate transformation of the vector (in terms of the two reference frames) is accomplished.

When dealing with a sequence of rotations, a passive rotation matrix multiplies the rotation matrix sequentially on the left-side of the previous rotation [23]. Thus, the order of multiplications is right to left. This means that the last rotation will be accounted by the leftmost matrix and so on.

Active rotations

An *active* rotation implies a positive counter-clockwise rotation (following the same sign convention) around an arbitrary axis that "actively" rotates vectors while the frame remains fixed during the rotation. Therefore, the situation is inverted with respect to *passive* rotations. These classes of rotations are also called body or vector rotations. This is due to the fact that a new vector (identified by new components) is obtained. The new coordinates are expressed in the same reference frame.

Active rotation matrices differ from *passive* ones by the sign inversion of the sine functions:

$$\mathbf{R}_a^x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\vartheta & -\sin\vartheta \\ 0 & \sin\vartheta & \cos\vartheta \end{bmatrix} \tag{1.15}$$

$$\mathbf{R}_a^y = \begin{bmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{bmatrix} \tag{1.16}$$

$$\mathbf{R}_a^z = \begin{bmatrix} \cos\vartheta & -\sin\vartheta & 0 \\ \sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.17}$$

Active rotation matrices multiplies on the right-side of the previous ones [23]. Hence, as opposed to *passive* matrices, the order of multiplications is left to right.

In conclusion, a counter-clockwise rotation of axes (*passive* rotation) is equal to a clockwise rotation of vectors (*active* rotation). This equality can be formulated as:

$$\mathbf{R}(\vartheta) = \mathbf{R}_a(-\vartheta) \tag{1.18}$$

and vice versa. The proof is reported in Reference [35].

The idea is illustrated in Figure 1.3 by exploiting a simple example. An arbitrary vector $\mathbf{v}$ is considered. In terms of the x, y, z coordinate frame, its components are $[1, 1, 0]^T$. If the reference frame is rotated around the z-axis by a value of $\vartheta = 45°$ (counter-clockwise *passive* rotation), the new coordinates that identify the vector $\mathbf{v}$ will be $[\sqrt{2}, 0, 0]^T$. The same result is obtained by rotating vector $\mathbf{v}$ by $\vartheta = -45°$(clockwise *active* rotation) while axes x, y and z remain fixed in space. This example (taken from Reference [35]) verifies the equality.

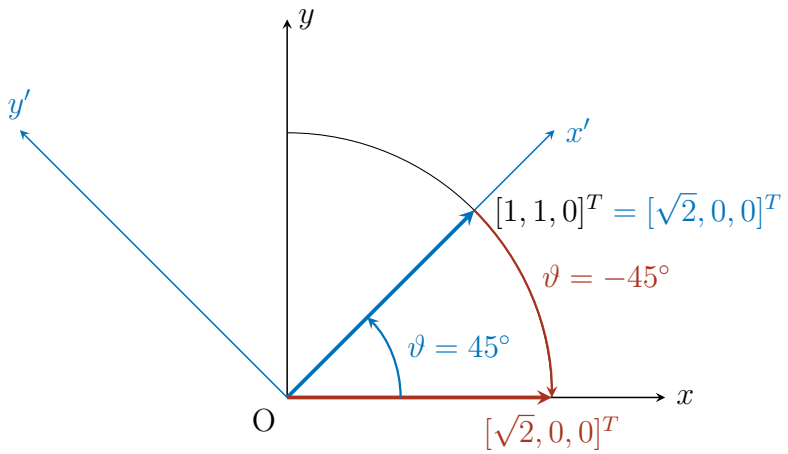


Figure 1.3: Example of equality between passive and active rotation matrices.

1.3.3 Euler angles

The orientation of one Cartesian coordinate system with respect to another can be described by three successive rotations around the orthogonal coordinate axes; the angles of rotation are called *Euler angles* (or Eulerian angles). The axes and the order of the rotations are chosen in various ways depending on the field of science.

In the aerospace sector, two orders are predominant: x, y, z and z, y, x. In particular, standard practice is to describe aircraft orientation by the z, y, x (also called 3, 2, 1) right-handed Euler rotation sequence. It allows to get from a reference system on the surface on Earth (NED frame) into alignment with an aircraft body-fixed coordinate system (Global frame). The procedure described in Reference [37] involves:

1. A positive rotation ψ around D axis. N axis is rotated into x' while E axis becomes y':

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N \\ E \\ D \end{bmatrix} \tag{1.19}$$

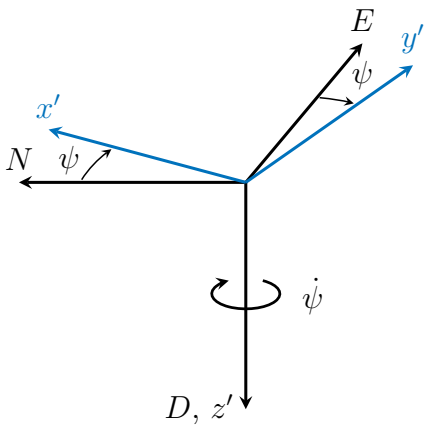


Figure 1.4: Euler angles: rotation ψ around D axis.

2. A positive rotation ϑ around y' axis. x' axis is rotated into x'' while z' axis becomes z'':

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \cos\vartheta & 0 & -\sin\vartheta \\ 0 & 1 & 0 \\ \sin\vartheta & 0 & \cos\vartheta \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{1.20}$$

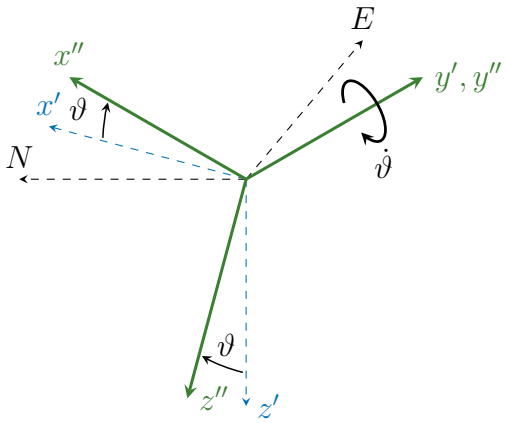


Figure 1.5: Euler angles: rotation ϑ around y' axis.

3. A positive rotation φ around x'' axis. y'' axis is rotated into Y while z'' axis becomes Z:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \tag{1.21}$$

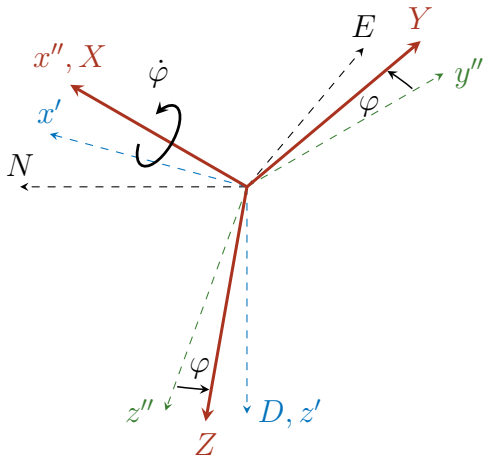


Figure 1.6: Euler angles: rotation φ around x'' axis.

Since the goal is to transform coordinates of an arbitrary vector from one reference system to a rotated one, *passive* rotation matrices are exploited. It is worth noting that the zeros and particularly the "1" are placed so that the component of the vector along the rotation axis is left unchanged; cosines are collocated on the main diagonal while sines in the remaining off-diagonal positions. This implies that the matrix reverts to the identity matrix when the rotation angle is zero. Finally, the negative sine term is put on the row above the "1" when a positive angle corresponds to a right-handed rotation around the current axis [37].

In summary, the transformation that allows to move from the NED to the G (XYZ) reference frame is given by the multiplication of the rotation matrices:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} \cos\vartheta & 0 & -\sin\vartheta \\ 0 & 1 & 0 \\ \sin\vartheta & 0 & \cos\vartheta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N \\ E \\ D \end{bmatrix} \tag{1.22}$$

In a more compact form:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{R}_\varphi \, \mathbf{R}_\vartheta \, \mathbf{R}_\psi \begin{bmatrix} N \\ E \\ D \end{bmatrix} \tag{1.23}$$

By performing the matrix multiplications, the following matrix is obtained:

$$\mathbf{R}_{N\to G} = \begin{bmatrix} \cos\vartheta\cos\psi & \cos\vartheta\sin\psi & -\sin\vartheta \\ -\cos\varphi\sin\psi + \sin\varphi\sin\vartheta\cos\psi & \cos\varphi\cos\psi + \sin\varphi\sin\vartheta\sin\psi & \sin\varphi\cos\vartheta \\ \sin\varphi\sin\psi + \cos\varphi\sin\vartheta\cos\psi & -\sin\varphi\cos\psi + \cos\varphi\sin\vartheta\sin\psi & \cos\varphi\cos\vartheta \end{bmatrix}$$

The defined ranges for Euler angles are:

$$-\pi < \varphi \leq \pi \tag{1.24}$$

$$-\frac{\pi}{2} \leq \vartheta \leq \frac{\pi}{2} \tag{1.25}$$

$$-\pi < \psi \leq \pi \tag{1.26}$$

If the pitch angle ϑ had been allowed to have a $(-\pi, \pi]$ range, the airplane (or UAV) could fly inverted and heading South with the roll and heading angles reading zero. This condition is not desirable from a human factors viewpoint [37].

Chapter 2

Simulation

The present Chapter outlines the procedures that lead to the development of a model that simulate *TetrAHRS*, as it moves in the three dimensional space. Starting from the geometry of the system and its dynamics, measurements obtained from the four MEMS sensors are generated. These data are subsequently used as inputs for the Kalman filter, which requires consistent information in order to work properly.

The simulation framework involves three subsequent phases:

1. **Generation of "ideal" sensor measurements**, due to the motion and rotation of *TetrAHRS* on a prescribed arbitrary trajectory.
2. **Generation of noisy sensor measurements**, through the simulation of sensors, based on a model that includes their own dynamics, real effects and noise. Ideal measurements, computed in the previous phase, serve as input to the sensor model. Whereas, the output consists in noisy measurements that effectively reproduce data obtained from real-world sensors.
3. **Validation of the model**, including both ideal and noisy measurements. The whole process that led to measurements is now inverted: starting from sensor outputs, the goal is to compute accelerations, angular rates and magnetic quantities that were initially imposed to *TetrAHRS*. The procedure is checked for a sufficient number of relevant trajectories, in order to cover the whole range of motion.

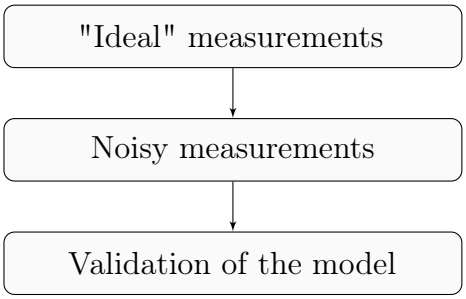


Figure 2.1: Phases of the simulation framework.

2.1 Generation of "ideal" sensor measurements

The first step is to create a mathematical model that describe the geometry of *TetrAHRS*. Since the object is built with a sufficiently stiff material and it is not subjected to relevant stresses, it can be modeled as a rigid-body. The system is in fact assumed to be fixed to the structure of the aircraft, ensuring that is does not receive structural loads.

Once all geometric quantities are computed, the focus moves to the dynamics of the rigid-body. An arbitrary trajectory is prescribed, choosing the final desired position of the tetrahedron and its final attitude (expressed in terms of Euler angles). Thanks to a fifth order polynomial, position, velocity and acceleration of the center of mass (CM) are determined, along with attitude, angular rates and angular accelerations of the rigid-body. All values are computed at each instant of time, during the simulation.

The final aim is to obtain sensor measurements. Hence, all quantities that are at first referred to the CM of the tetrahedron are then related to positions of the sensors, considering their local reference frames.

2.1.1 Reference frames

The whole system, including the four MEMS sensors, can be suitably modeled introducing three reference frames:

- NED reference frame: $NED(O, N, E, D)$. It stands for North-East-Down and it is assumed to be an inertial coordinate system. The trajectory along which the tetrahedron moves is expressed in terms of these coordinates. Therefore, position, velocity and acceleration of the CM are referred to this reference frame. The same applies to attitude (in terms of Euler Angles).

- Global (body) reference frame: $G(\text{CM}, X, Y, Z)$. It is carried by the tetrahedron. The position of its origin coincides with the CM of *TetrAHRS* and the axes rotate with it. The rotations with respect to the NED reference frame can be expressed in terms of Euler angles. Angular rates and angular acceleration of the rigid-body refer to this frame (after being handled with appropriate rotations).

- Local reference frames: $L_i(s_i, x_i, y_i, z_i)$. They are referred to each *i-th* sensor; therefore, four different local reference frames are being considered. Since the tetrahedron is assumed to be rigid, their position is fixed with respect to the Global reference frame (G); they also rotate rigidly with it.

All reference frames are assumed to be orthogonal and right-handed. They are shown in Figure 2.2, represented using different colors.

2.1.2 Geometry of the tetrahedron and sensor positions

The starting point of the geometric model is a tetrahedron that has a settable edge length L. As reported in Reference [6], geometric quantities can be computed on

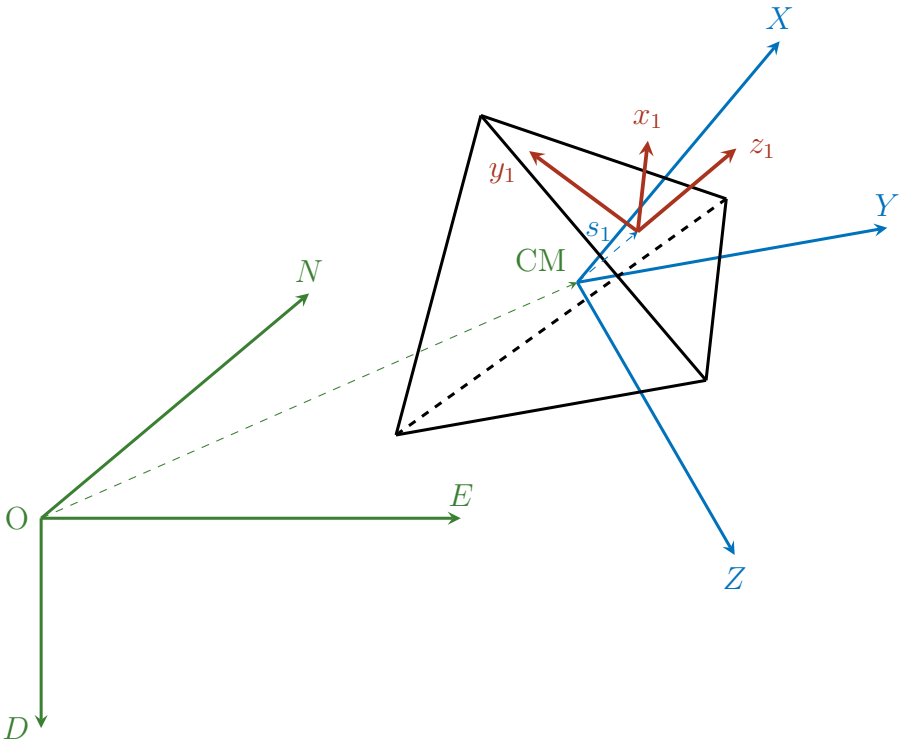


Figure 2.2: Graphic representation of the reference frames.

the basis of the edge, beginning with the height of the polyhedron:

$$h = \sqrt{\frac{2}{3}}L \tag{2.1}$$

The four faces are equilateral triangles. Therefore, the radius of the circumference inscribed into each face is:

$$r = \frac{L}{2\sqrt{3}} \tag{2.2}$$

It is worth mentioning that, in an equilateral triangle, altitudes, angle bisectors, perpendicular bisectors and medians to each side coincide. So, each median is divided by the others in two parts: the length of one segment is r while the length of the other is $2r$.

The origin of the Global reference frame (G) is located in the CM of the tetrahedron. One way to obtain its position consists in finding the radius of the in-sphere that is tangent to faces:

$$h_{CM} = \frac{L}{\sqrt{24}} \tag{2.3}$$

Or, equivalently, exploiting the tangent function, related to with r:

$$h_{CM} = (2r)\tan(\frac{\pi}{2} - \arctan(2\sqrt{2})) \tag{2.4}$$

where $\arctan(2\sqrt{2})$ represents the face-edge-face angle of the tetrahedron, i.e., "dihedral angle". It is approximately 70.5288°.

The position of vertex corners can be determined accordingly:

$$^{G}\mathbf{v}_1 = [2r, 0, h_{CM}] \tag{2.5}$$

$$^{G}\mathbf{v}_2 = [-r, \frac{L}{2}, h_{CM}] \tag{2.6}$$

$$^{G}\mathbf{v}_3 = [-r, -\frac{L}{2}, h_{CM}] \tag{2.7}$$

$$^{G}\mathbf{v}_4 = [0, 0, h_{CM} - h] \tag{2.8}$$

Once the geometry of the tetrahedron is fully defined, the four MEMS sensor are introduced. For simplicity, they are placed in the geometric center of each face. In addition, to a first approximation, they are assumed to be dimensionless; this hypothesis is compatible with their extremely small size. In order to better model the real system, their position takes into account the fact that they are located in designated cavities. So, a distance a from the face is considered.

Focusing on the segment that connects one vertex corner to the geometric center of the opposite face (which is perpendicular to the face and has length h), the actual distance between the sensor and the vertex becomes:

$$d = h - a \tag{2.9}$$

The projection of the distance d on the face at the base of the tetrahedron leads to:

$$d_p = d\cos(\frac{\pi}{2} - \arctan(2\sqrt{2})) \tag{2.10}$$

Subsequently, the distance between the sensor and the origin is computed (considering the properties of equilateral triangles):

$$d_s = d_p - 2r \tag{2.11}$$

The segment d_s has to be projected on the X and Y axes:

$$d_{s,X} = d_s\sin(\frac{\pi}{6}) \tag{2.12}$$

$$d_{s,Y} = d_s\cos(\frac{\pi}{6}) \tag{2.13}$$

The last quantity that is needed to determine the position of each sensor (in the G frame) is the height of sensors from the plane at the base:

$$h_s = d\sin(\frac{\pi}{2} - \arctan(2\sqrt{2})) \tag{2.14}$$

Sensor positions are finally obtained:

$$^{G}\mathbf{s}_1 = [d_{s,X}, d_{s,Y}, h_{CM} - h_s] \tag{2.15}$$

$$^{G}\mathbf{s}_2 = [-d_s, 0, h_{CM} - h_s] \tag{2.16}$$

$$^{G}\mathbf{s}_3 = [d_{s,X}, -d_{s,Y}, h_{CM} - h_s] \tag{2.17}$$

$$^{G}\mathbf{s}_4 = [0, 0, h_{CM} - a] \tag{2.18}$$

2.1.3 Generation of trajectory and angular quantities

The generation of the trajectory followed by the tetrahedron and angular quantities is accomplished through a simulation in time, based on fifth-order polynomials.

Trajectory

Considering the early stage of simulation, a straight trajectory has been chosen. At first, it is considered the motion from an assigned starting point to a final one. Then, the basic concept is extended in order to create a more elaborate path among desired waypoints.

The trajectory between two successive points is computed through a fifth order polynomial, following the procedure outlined in Reference [36]. The polynomial requires the following parameters as inputs:

- Initial position of the CM (starting waypoint)
- Final position of the CM (ending waypoint)

The two quantities are expressed in terms of the NED reference frame. As a consequence, all quantities computed afterwards are referred to the same frame.

Initial and final values of both velocity and acceleration also need to be defined, in order to make the polynomial fully constrained. Since the main interest lies in the consistency of measurements (rather than the study of a particular trajectory), they are set to zero. However, their value can be arbitrarily changed according to the needs of the specific application.

The polynomial is defined in the following way:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{2.19}$$

Introducing initial time t_0 and final time t_f, the constraints at points $q(t_0)$ and $q(t_f)$ can be defined as follows:

$$q(t_0) = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5 \tag{2.20a}$$
$$v(t_0) = a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4 \tag{2.20b}$$
$$a(t_0) = 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3 \tag{2.20c}$$
$$q(t_f) = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \tag{2.20d}$$
$$v(t_f) = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \tag{2.20e}$$
$$a(t_f) = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \tag{2.20f}$$

These equations can be organized into matrix form in order to determine the six coefficients:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 1 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 1 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q(t_0) \\ v(t_0) \\ a(t_0) \\ q(t_f) \\ q(t_f) \\ a(t_f) \end{bmatrix} \tag{2.21}$$

Once the coefficients have been computed, position, velocity and acceleration of the CM are known at each instant of time.

The choice of a fifth-order polynomial is more suitable than a cubic one. A third-order polynomial would in fact lead to discontinuities in the acceleration at boundaries (or via points, in case of more complex trajectories); as a result, the derivative of acceleration (jerk) would be infinite at those points. This problem can be avoided by using a quintic polynomial [36].

Attitude and angular quantities

The same polynomial can also be applied to angular quantities, setting other two parameters (both expressed in terms of Euler angles):

- Initial attitude of the body (starting waypoint)
- Final attitude of the body (ending waypoint)

First and second (time) derivatives of Euler angles at boundaries are set to zero for the same reason that was explained in the previous paragraph. The initial and final attitude of the body can be chosen arbitrarily, but keeping in mind that a singularity occurs for $\vartheta = \pm\frac{\pi}{2}$ (due to the nature of Euler angles). This problem is commonly known as "Gimbal Lock" and it consists in the inability to uniquely identify the attitude. For this reason, a check of the value of ϑ has been implemented in the code. If that particular attitude is approached, a warning message appears.

The choice of the parametrization by means of Euler angles over quaternions was justified by two considerations. The first one is that they lead to a simpler formulation of the Kalman Filter, as explained in Chapter 3. Whereas, the second reason is related to the fact that Euler angles are compatible with most of the applications for which *TetraAHRS* is thought. In fact, exceptions aside, the majority of airplanes and drones have a very low probability of exceeding the limits imposed by Euler angles. The parametrization with quaternions would solve the problem of the singularity but their implementation requires more effort, especially concerning the Kalman Filter. Thus, they will be considered in future developments.

When dealing with a parametrization in terms of Euler angles (and their derivatives), it is important to remember that three subsequent rotations are performed. They are described in Section 1.3.3.

Two out of the three rotations are performed around axes that do not coincide with the ones of the G reference frame:

- φ around $X = x''$ axis
- ϑ around $y' = y''$ axis
- ψ around $D = z'$

It follows that Euler angles derivatives ($\dot{\varphi}$, $\dot{\vartheta}$ and $\dot{\psi}$) are not directly components along the G reference axes. Since the angular velocity vector $\boldsymbol{\omega}_G$ is associated with

the rigid-body, it has to be expressed in the G frame. Hence, its components have to be projected on the latter according to the following relation:

$$ {}^{G}_{N}\boldsymbol{\omega}_{G} = \mathbf{R}_{\varphi}\,\mathbf{R}_{\vartheta}\,\dot{\boldsymbol{\psi}} + \mathbf{R}_{\varphi}\,\dot{\boldsymbol{\vartheta}} + \dot{\boldsymbol{\varphi}} \tag{2.22} $$

where:

$$ \dot{\boldsymbol{\varphi}} = \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} \quad \dot{\boldsymbol{\vartheta}} = \begin{bmatrix} 0 \\ \dot{\vartheta} \\ 0 \end{bmatrix} \quad \dot{\boldsymbol{\psi}} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{2.23} $$

Performing the multiplications, it follows that:

$$ \begin{bmatrix} \omega_{G_X} \\ \omega_{G_Y} \\ \omega_{G_Z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\vartheta \\ 0 & \cos\varphi & \sin\varphi\cos\vartheta \\ 0 & -\sin\varphi & \cos\varphi\cos\vartheta \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix} \tag{2.24} $$

Thus, the three components of $\boldsymbol{\omega}_G$ are:

$$ \omega_{G_X} = \dot{\varphi} - \dot{\psi}\sin\vartheta \tag{2.25a} $$

$$ \omega_{G_Y} = \dot{\vartheta}\cos\varphi + \dot{\psi}\sin\varphi\cos\vartheta \tag{2.25b} $$

$$ \omega_{G_Z} = \dot{\psi}\cos\varphi\cos\vartheta - \dot{\vartheta}\sin\varphi \tag{2.25c} $$

The same procedure can be applied to angular accelerations. The angular acceleration vector $\boldsymbol{\alpha}_G$ is also projected in the G frame, following Eq. (2.22):

$$ {}^{G}_{N}\boldsymbol{\alpha}_{G} = \mathbf{R}_{\varphi}\,\mathbf{R}_{\vartheta}\,\ddot{\boldsymbol{\psi}} + \mathbf{R}_{\varphi}\,\ddot{\boldsymbol{\vartheta}} + \ddot{\boldsymbol{\varphi}} \tag{2.26} $$

Finally, making the components explicit:

$$ \alpha_{G_X} = \ddot{\varphi} - \ddot{\psi}\sin\vartheta \tag{2.27a} $$

$$ \alpha_{G_Y} = \ddot{\vartheta}\cos\varphi + \ddot{\psi}\sin\varphi\cos\vartheta \tag{2.27b} $$

$$ \alpha_{G_Z} = \ddot{\psi}\cos\varphi\cos\vartheta - \ddot{\vartheta}\sin\varphi \tag{2.27c} $$

Comprehensive simulation between two waypoints

Once all values at the two waypoints are defined, the polynomial allow to determine the quantities listed below:

- Position vector of the CM
- Velocity vector of the CM
- Acceleration vector of the CM
- Attitude of the body (expressed in terms of Euler angles)
- First (time) derivative of Euler angles, then transformed into $\boldsymbol{\omega}_G$
- Second (time) derivative of Euler angles, then transformed into $\boldsymbol{\alpha}_G$

They are all computed at each instant of time.

2.1.4 Kinematics

The kinematics associated with *TetrAHRS* can be formulated by introducing two vectors. The first one is $\mathbf{r}_{CM}$. It identifies the CM of the system (origin of the G frame) and it is expressed in terms of NED coordinates. The second vector is $\mathbf{s}_i$, which locates the position of the *i-th* sensor. Unlike $\mathbf{r}_{CM}$, it is expressed in the G frame. Since sensors are fixed on the tetrahedron, $\mathbf{s}_i$ is constant in magnitude and rotates together with the body. So, its components are constant with respect to G. The quantities just mentioned are represented in Figure 2.3.

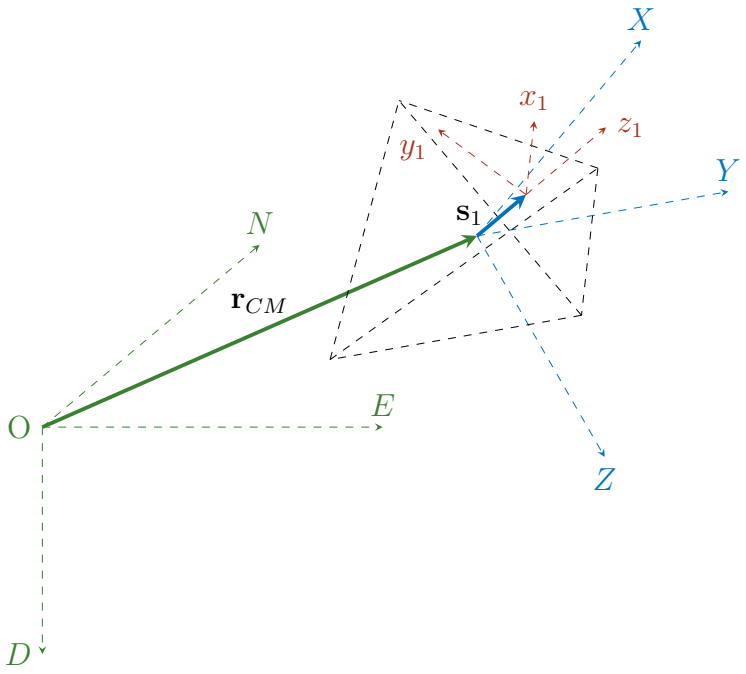


Figure 2.3: Representations of geometric quantities used in the kinematics.

The model involves the three reference frames that were previously introduced:

- Inertial NED reference frame (abbreviated as "N")
- Global reference frame (G), integral with *TetrAHRS*
- Local reference frame (L_i), referred to the *i-th* sensor

The goal of this section is to compute the acceleration experienced by the sensor, referred to the G frame (Global-acceleration). Given that the sensors are fixed with respect to the Global triad, appropriate rotations of reference frames (performed later on) will be sufficient to accomplish the task.

Acceleration of the TetrAHRS center of mass (CM)

The procedure outlined in Reference [21] offers a guideline for achieving the desired result. It begins with the determination of the velocity with respect to the G frame

but expressed in the NED one. Recalling the *simple derivative transformation formula* and the mixed derivatives:

$$ {}^{N}_{G}\mathbf{v}_{CM} = {}^{G}\frac{d}{dt}{}^{N}\mathbf{r}_{CM} = {}^{N}\frac{d}{dt}{}^{N}\mathbf{r}_{CM} - {}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{r}_{CM} \tag{2.28} $$

where ${}_{N}\boldsymbol{\omega}_{G}$ represents the angular velocity vector associated with the G frame (which is integral to *TetrAHRS*), expressed in the NED system.

Hence,

$$ {}^{N}_{G}\mathbf{v}_{CM} = {}^{N}\mathbf{v}_{CM} - {}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{r}_{CM} \tag{2.29} $$

Left superscripts indicate the frame in which the quantity is expressed while left subscripts define the frame in which that derivative is taken. If the left super and subscripts of a velocity/acceleration vector are the same, it is a simple velocity/acceleration vector and only the superscript is kept.

The acceleration experienced by the CM can be computed by differentiating the velocity with respect to G:

$$ \begin{aligned} {}^{N}_{G}\mathbf{a}_{CM} &= {}^{G}\frac{d}{dt}{}^{N}_{G}\mathbf{v}_{CM} = {}^{G}\frac{d}{dt}({}^{N}\mathbf{v}_{CM} - {}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{r}_{CM}) = \\ &= {}^{N}\mathbf{a}_{CM} - {}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{v}_{CM} - ({}_{N}\boldsymbol{\alpha}_{G} - {}_{N}\boldsymbol{\omega}_{G} \times {}_{N}\boldsymbol{\omega}_{G}) \times {}^{N}\mathbf{r}_{CM} \\ &- {}_{N}\boldsymbol{\omega}_{G} \times ({}^{N}\mathbf{v}_{CM} - {}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{r}_{CM}) = \\ &= {}^{N}\mathbf{a}_{CM} - {}_{N}\boldsymbol{\alpha}_{G} \times {}^{N}\mathbf{r}_{CM} - 2\,{}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{v}_{CM} \\ &+ {}_{N}\boldsymbol{\omega}_{G} \times ({}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{r}_{CM}) \end{aligned} \tag{2.30} $$

Therefore, the N-expression of the G-acceleration is:

$$ \begin{aligned} {}^{N}_{G}\mathbf{a}_{CM} = {}^{N}\mathbf{a}_{CM} &- {}_{N}\boldsymbol{\alpha}_{G} \times {}^{N}\mathbf{r}_{CM} - 2\,{}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{v}_{CM} \\ &+ {}_{N}\boldsymbol{\omega}_{G} \times ({}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{r}_{CM}) \end{aligned} \tag{2.31} $$

where:

- ${}_{N}\boldsymbol{\alpha}_{G} \times {}^{N}\mathbf{r}_{CM}$ is the *tangential acceleration*,
- $2\,{}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{v}_{CM}$ is the *Coriolis acceleration* and
- ${}_{N}\boldsymbol{\omega}_{G} \times ({}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{r}_{CM})$ is the *centripetal acceleration*.

Because the sensors are integral with the G frame, it is natural to express the acceleration vector ${}^{N}_{G}\mathbf{a}_{CM}$ (currently in NED coordinates) in the Global one. The matrix reported in Eq. (1.23) allows to accomplish this task:

$$ \mathbf{R}_{N\to G} = \mathbf{R}_{\varphi}\,\mathbf{R}_{\vartheta}\,\mathbf{R}_{\psi} \tag{2.32} $$

It corresponds to three subsequent rotations of the reference axes (according to the corresponding Euler angles) while the acceleration vector stands still. As stated in Reference [37], it represents a change of coordinates.

Hence,

$$ {}^{G}\mathbf{a}_{CM} = \mathbf{R}_{N\to G}\,{}^{N}_{G}\mathbf{a}_{CM} \tag{2.33} $$

Acceleration experienced by accelerometers

Once ${}^{G}\mathbf{a}_{CM}$ is determined, the linear acceleration $\mathbf{a}_i$ experienced by the *i-th* sensor can be computed considering its position $\mathbf{s}_i$ (with respect to the CM) and the angular quantities related to the rigid-body: ${}^{G}_{N}\boldsymbol{\omega}_G$ and ${}^{G}_{N}\boldsymbol{\alpha}_G$.

$$ {}^{G}\mathbf{a}_i = {}^{G}\mathbf{a}_{CM} + {}^{G}_{N}\boldsymbol{\alpha}_G \times {}^{G}\mathbf{s}_i + {}^{G}_{N}\boldsymbol{\omega}_G \times ({}^{G}_{N}\boldsymbol{\omega}_G \times {}^{G}\mathbf{s}_i) \tag{2.34} $$

This last expression represents the sum of *linear*, *tangential* and *centripetal* accelerations. Moreover, it already takes into account that

$$ {}^{G}\frac{d}{dt}{}^{G}\mathbf{s}_i = 0 \tag{2.35} $$

given the fact that sensors are fixed on the tetrahedron.

2.1.5 Generation of measurements data

TetrAHRS is comprised of four MEMS sensors, located in the geometric center of each face of the tetrahedron. Local reference systems (L_i) are assumed to be centered in those points and their axes are oriented in the following way:

- Axis x_i belongs to the corresponding face plane and it is parallel to the ground when looking at the tetrahedron from its side, in its initial configuration (aligned with NED reference frame). It points towards the right direction.
- Axis y_i also belongs to the face plane but it is orthogonal to x_i, pointing towards the upper vertex $\mathbf{v}_4$.
- Axis z_i is orthogonal to the face plane, pointing outwards (in the opposite direction of the CM of the tetrahedron).

Axes are shown in Figure 2.4. The sensor located at the base of *TetrAHRS* (no. 4) follows the same rules, with x_4 parallel to $\overline{v_2v_3}$, y_4 pointing towards $\mathbf{v}_1$ and z_4 towards the ground.

Each sensor includes:

1. Triaxial accelerometer → Linear accelerations
2. Triaxial gyroscope → Angular rates
3. Triaxial magnetometer → Components of Earth's magnetic field vector

All measurements coming from each sensor have to be expressed in terms of the corresponding L_i frame. So, the transformation that allows to move from reference frame G to L_i, has to be introduced. It involves two subsequent rotations:

1. A positive rotation β_i around Z axis, represented by:

$$ \mathbf{R}_Z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta_i & \sin\beta_i \\ 0 & -\sin\beta_i & \cos\beta_i \end{bmatrix} \tag{2.36} $$

   where β_i is characteristics of each sensor.

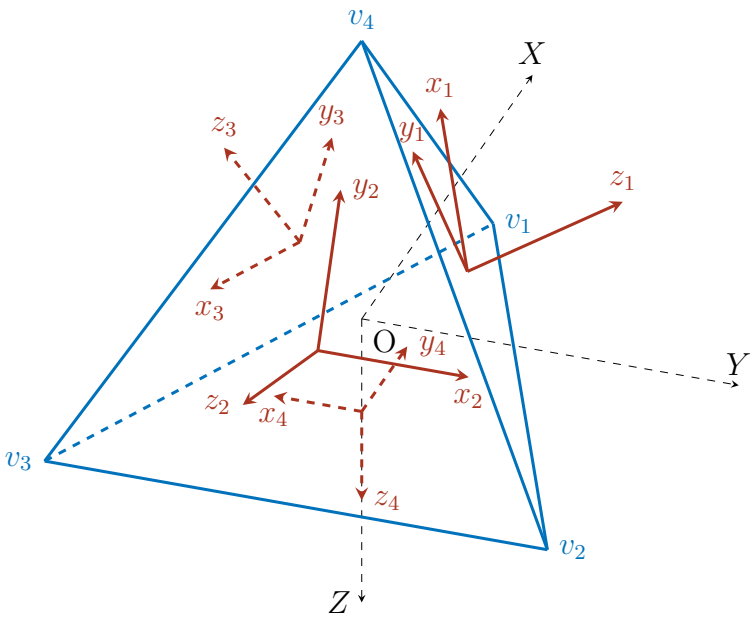


Figure 2.4: Representation of local reference frames L_i.

2. A positive rotation γ_i around the new x_i axis, denoted by:

$$\mathbf{R}_{x_i} = \begin{bmatrix} \cos\gamma_i & \sin\gamma_i & 0 \\ -\sin\gamma_i & \cos\gamma_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.37}$$

with γ_i that varies according to the sensor.

Hence, the *passive rotation* matrix is given by:

$$\mathbf{R}_{G\to L_i} = \mathbf{R}_{x_i}\,\mathbf{R}_Z \tag{2.38}$$

Rotations are not commutative, so, a considerable care must be taken in the choice of the order of multiplication. Further explanation can be found in 1.3.2.

Table 2.1: Values of angles β_i and γ_i for each sensor.

Sensor	β_i	γ_i
1	$-\frac{\pi}{6}$	$-\pi + \arctan(2\sqrt{2})$
2	$\frac{\pi}{2}$	$-\pi + \arctan(2\sqrt{2})$
3	$\frac{7}{6}\pi$	$-\pi + \arctan(2\sqrt{2})$
4	$-\frac{\pi}{2}$	/

As reported in Table 2.1, values of β_i and γ_i are constant. This means that $\mathbf{R}_{G\to L_i}$ is a consant matrix, in contrast to $\mathbf{R}_{N\to G}$, which depends on the attitude. In conclusion, it can be observed that Sensor no. 4 (which is located at the base of the tetrahedron) only requires a rotation β_i around Z axis.

Linear accelerations

An accelerometer is a device that measure *proper acceleration*. *Proper acceleration* is the acceleration (rate of change of velocity) of a body in its own instantaneous rest frame [34]; this is different from *coordinate acceleration*, which is acceleration in a fixed coordinate system. For example, an accelerometer at rest on the surface of the Earth measures an acceleration due to Earth's gravity, straight "upwards" of $g \simeq 9.81$ m/s^2. By contrast, accelerometers in free fall (falling toward the center of the Earth at a rate of about g would measure zero [5].

Therefore, acceleration data should take into account the contribution of the acceleration ${}^{G}\mathbf{a}_i$, coming from the motion of the tetrahedron (that was computed in Section 2.1.4) and the gravity contribution. Both terms have to be properly rotated in the *i-th* Local reference frame (L_i):

$$\mathbf{n}_i = \mathbf{R}_{G\to L_i}\,{}^{G}\mathbf{a}_i - \mathbf{R}_{G\to L_i}\,\mathbf{R}_{N\to G}\,{}^{N}\mathbf{g} \tag{2.39}$$

where $\mathbf{R}_{N\to G}$ and $\mathbf{R}_{G\to L_i}$ were defined, respectively, in Eq. (2.32) and Eq. (2.38). The negative sign that precede the gravity contribution is necessary to model the vector pointing "upwards". In fact, gravity vector ${}^{N}\mathbf{g}$ is expressed in the NED frame:

$${}^{N}\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{2.40}$$

with $g \simeq 9.80665$ m/s^2.

In conclusion, the *proper acceleration* has to be measured in units that are multiples of g, so the following division is performed:

$$\mathbf{n}_{i,\,g} = \frac{\mathbf{n}_i}{g} \tag{2.41}$$

Angular rates

Gyroscopes measure angular rates along their three axes (which constitute the L_i frame). Angular velocity vector ${}^{G}_{N}\boldsymbol{\omega}_G$ was computed in Eq. (2.22); the left superscript denotes that it is expressed in the G reference frame. Hence, in order to simulate ideal gyroscope readings, the vector has to be projected in the L_i frame:

$$\boldsymbol{\omega}_i = \mathbf{R}_{G\to L_i}\,{}^{G}_{N}\boldsymbol{\omega}_G \tag{2.42}$$

where $\mathbf{R}_{G\to L_i}$ is the *passive rotation* matrix that was introduced in Eq. (2.38).

Components of Earth's magnetic field vector

Magnetometers are devices that measure magnetic fields. In particular, for the purpose of the project, the object of interest is Earth's magnetic field.

At any location, the latter can be represented by a three-dimensional vector. A typical procedure for measuring its orientation is to use a compass to determine

the direction of magnetic North. Its angle relative to true North is the *declination* (D) or *variation*. While, facing magnetic North, the angle that the field makes with the horizontal is called *inclination* (I) or *magnetic dip*. The two quantities are represented in Figure 2.5. Finally, the *intensity* (F) of the field is proportional to the force it exerts on a magnet [26]. Thus, it corresponds to the magnitude of the magnetic field vector.

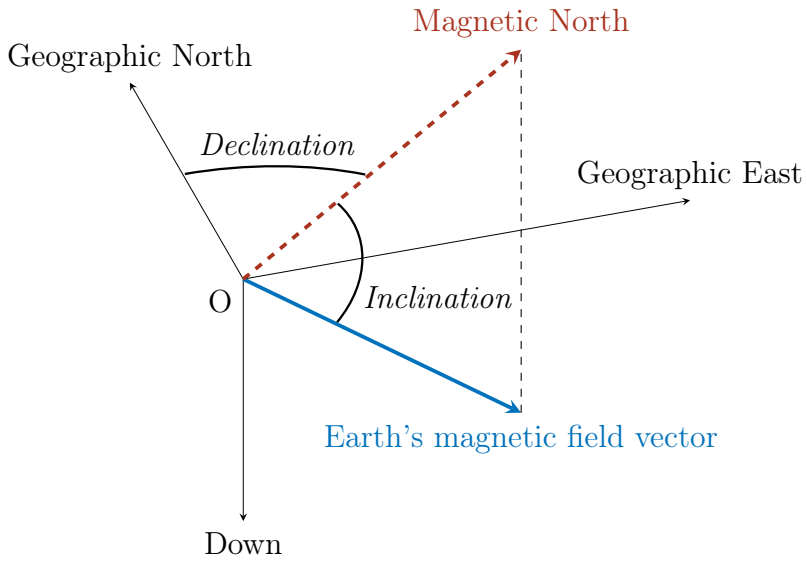


Figure 2.5: Earth's magnetic field vector.

It follows a more accurate description of the parameters:

1. *Declination* is positive for an eastward deviation of the field relative to true north. Information on declination for a region can be represented by a chart with isogonic lines.

2. *Inclination* is given by an angle that can assume values between -90° (up) to 90° (down). In the northern hemisphere, the field points downwards. It is straight down at the North Magnetic Pole and rotates upwards as the latitude decreases until it is horizontal (0°) at the magnetic equator. It continues to rotate upwards until it is straight up at the South Magnetic Pole.

3. *Intensity* of Earth's field ranges approximately between 0.25 G and 0.65 G, according to Reference [30].

Depending on the geographic location where the field is measured, the three quantities have different values. Thus, once the position is set, they can be obtained from dedicated charts and databases.

In view of the early stage of simulation, the following assumptions can be made:

1. It is not relevant that the magnetic field vector is exact. It is however important that the vector remains constant (in both magnitude and orientation) during the whole simulation process. This assumption is valid if the trajectories

involved are on a scale (up to $10^1 \div 10^2$ m) that is lower than the one that would make the magnetic vector change significantly. On the other hand, if the application implies greater distances, this aspect should be taken into consideration and dealt with appropriate updates of the magnetic vector.

2. To a first approximation, it is plausible to assume that the magnetometer is only subjected to Earth's magnetic field, excluding possible distortions due to other magnetic sources nearby. In real applications, this condition can be verified through the analysis of the operating environment and measurements; once it is estimated, it can be compensated accordingly.

Therefore, a magnetic field vector is introduced:

$$ {}^{N}\mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \tag{2.43} $$

The components are expressed in the NED frame. They are computed considering a generic geographic location and they are assumed to remain constant during the entire duration of the simulation.

Similarly to what was done with accelerations and angular rates, the magnetic field vector is projected in the local reference frame of the sensors (L_i). This procedure retraces the same path that has been followed in Eq. (2.39):

$$ \mathbf{m}_i = \mathbf{R}_{G \to L_i} \, \mathbf{R}_{N \to G} \, {}^{N}\mathbf{m} \tag{2.44} $$

The two *passive rotation* matrices $\mathbf{R}_{N \to G}$ and $\mathbf{R}_{G \to L_i}$ were defined, respectively, in Eq. (2.32) and Eq. (2.38).

Magnetic vector $\mathbf{m}_i$ represents the readings of each magnetometer. In the computation process, the attitude of *TetrAHRS* is considered through matrix $\mathbf{R}_{N \to G}$. This concept plays a key role in the formulation of the Kalman filter, outlined in Chapter 3.

2.2 Generation of noisy sensor measurements

Data computed in the previous Section were defined "ideal" because they are direct result of dynamical analysis and mathematical transformations. In order to simulate measurements obtained from real-world MEMS sensors, several aspects should be introduced. The most relevant ones are:

- Sensor dynamics
- Real effects
- Noise and biases

A Simulink model containing different types of aeronautic sensors was developed by Alberto Brandl (more details can be found in Reference [4]). It served as starting point for the design of the model that simulates the present configuration.

Tri-axial sensors were adapted to model accelerometers, gyroscopes and magnetometers. Since the project involves four identical MEMS sensors, the same layout (shown in Figure 2.6) was repeated four times in order to feature the complete system.

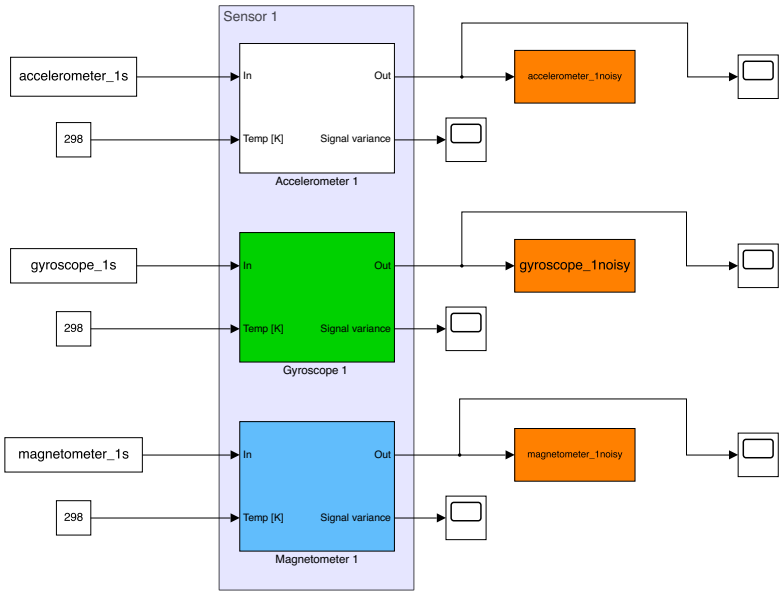


Figure 2.6: Simulink model of a single sensor.

Numerous options were implemented in the original model, including real effects associated with the order of the sensor and temperature, biases (fixed, turn-on and associated with temperature), white noises and auto-regressive (AR) process parameters on different axes.

The main idea at this stage is to proceed by steps, progressively increasing the complexity of the model. Thus, at first, a band-limited Gaussian White Noise (with zero mean and given variance σ^2) is added to the "ideal" signal; this provides a first approximation of the dispersion of sensor measurements. Values of variance σ^2 related to the different types of sensors are taken from experimental tests carried out on a real-world device. Procedures and details are described in Chapter 4. Moreover, in order to reach an higher level of generality, several tests will be performed considering a wider range of variances. Results will be discussed in Section 5.3.2.

A second and final step is the addition of a random turn-on to turn-on bias, which also takes into account other possible sources of biases. The details are described in Section 5.1.3.

However, an accurate sensor model would require further error contributions, including drifts (especially concerning the gyroscope), temperature-dependent biases,

scale factors errors, non-linearities, axes misalignments and more. Dedicated models should be implemented while errors shall be assessed through a sensor characterization process. An example of a viable approach to simulate accelerometers and gyroscopes is offered by Reference [43].

2.3 Validation of the model

A validation procedure has to be carried out, in order to verify that the model performs as expected. The main idea that underlies the approach is an inversion of the scheme that was used in the computation of measurements. In fact, starting from the latter, the aim is to obtain the values of linear accelerations, angular rates and components of Earth's magnetic field vector that were arbitrarily prescribed to *TetrAHRS*. This way, it is possible to check that the "going there" and "coming back" lead to same results.

The method is applied to both "ideal" and noisy data. An identity among values is expected only in the "ideal" case; while, in the second case, an exact correspondence is not possible due to the addition of noise. Nevertheless, the trend of prescribed (initial) data can be compared to the result of the inverse process, allowing to identify possible inconsistencies.

The superscript V, which stands for *validation*, will identify quantities that are computed through the inverted process.

2.3.1 Computation of validation data

The validation process involves the three set of outputs that are coming from the MEMS sensors:

1. Linear accelerations
2. Angular rates
3. Components of Earth's magnetic field vector

Validation data are calculated with both "ideal" and noisy measurements. The same procedure is followed; the only difference lies in the different outputs of sensors that are chosen.

Linear accelerations

Linear accelerations measured by triaxial accelerometers (integral part of the MEMS sensors) have to lead to the acceleration of the CM, that is imposed by prescribing initial and final positions of *TetrAHRS*. The procedure retraces the steps followed in Section 2.1.5, but with an appropriate inversion.

The first step is the computation of linear accelerations sensed by sensors due to the rotation of the body, in the same way that was presented in Eq. (2.34), but

excluding the term related to the CM:

$$ {}^{G}\mathbf{a}_{i,\,no\,CM}^{V} = {}_{N}^{G}\boldsymbol{\alpha}_{G} \times {}^{G}\mathbf{s}_{i} + {}_{N}^{G}\boldsymbol{\omega}_{G} \times ({}_{N}^{G}\boldsymbol{\omega}_{G} \times {}^{G}\mathbf{s}_{i}) \tag{2.45} $$

where ${}_{N}^{G}\boldsymbol{\alpha}_{G}$ and ${}_{N}^{G}\boldsymbol{\omega}_{G}$ are the same angular quantities that were used before, computed for each instant of time during the simulation.

The sensor output vector $\mathbf{n}_{i,g}$ is multiplied by g in order to obtain quantities expressed in $\mathrm{m/s^2}$:

$$ \mathbf{n}_{i}^{V} = \mathbf{n}_{i,\,g} * g \tag{2.46} $$

By manipulating Eq. (2.39), ${}^{G}\mathbf{a}_{i}^{V}$ can be obtained:

$$ {}^{G}\mathbf{a}_{i}^{V} = \mathbf{R}_{G\to L_i}^{-1}(\mathbf{n}_{i}^{V} + \mathbf{R}_{G\to L_i}\,\mathbf{R}_{N\to G}\,{}^{N}\mathbf{g}) \tag{2.47} $$

Rotation matrices (both *active* and *passive*) are orthogonal; so, their inverse coincides with their transpose:

$$ \mathbf{R}_{G\to L_i}^{-1} = \mathbf{R}_{G\to L_i}^{T} = \mathbf{R}_{L_i\to G} \tag{2.48} $$

Subsequently,

$$ {}^{G}\mathbf{a}_{CM,\,i}^{V} = {}^{G}\mathbf{a}_{i}^{V} - {}^{G}\mathbf{a}_{i,\,no\,CM}^{V} \tag{2.49} $$

Since the acceleration of TetrAHRS was expressed in the NED reference frame, an appropriate *passive rotation* is required:

$$ {}^{N}\mathbf{a}_{CM,\,i}^{V} = \mathbf{R}_{G\to N}\,{}^{G}\mathbf{a}_{CM,\,i}^{V} \tag{2.50} $$

with

$$ \mathbf{R}_{G\to N} = \mathbf{R}_{N\to G}^{-1} = \mathbf{R}_{N\to G}^{T} \tag{2.51} $$

Finally, taking advantage of Eq. (2.31):

$$ \begin{aligned} {}^{N}\mathbf{a}_{CM_{Sim},\,i}^{V} = {}^{N}\mathbf{a}_{CM,\,i}^{V} + {}_{N}\boldsymbol{\alpha}_{G} \times {}^{N}\mathbf{r}_{CM} + 2\,{}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{v}_{CM} \\ - {}_{N}\boldsymbol{\omega}_{G} \times ({}_{N}\boldsymbol{\omega}_{G} \times {}^{N}\mathbf{r}_{CM}) \end{aligned} \tag{2.52} $$

This way, four sets of accelerations ${}^{N}\mathbf{a}_{CM_{Sim},\,i}^{V}$ are obtained, each one relative to the corresponding sensor.

Angular rates

The same reasoning is applied to angular rates, measured by triaxial gyroscopes that are integrated in the sensors. Outputs $\boldsymbol{\omega}_i$ represent the starting point of the validation process. Each set of data is projected in the G frame; in fact, this transformation allows to compare them to the (initial) simulated ${}_{N}^{G}\boldsymbol{\omega}_{G}$.

To this end, the inverse of the *passive rotation* matrix presented in Eq. (2.38) shall be considered:

$$ \mathbf{R}_{L_i\to G} = \mathbf{R}_{G\to L_i}^{-1} = \mathbf{R}_{G\to L_i}^{T} \tag{2.53} $$

due to orthogonality.

Therefore,

$$ {}^{G}\boldsymbol{\omega}_{i}^{V} = \mathbf{R}_{L_i\to G}\,\boldsymbol{\omega}_{i} \tag{2.54} $$

Components of Earth's magnetic field vector

Measurements $\mathbf{m}_i$ coming from magnetometers consist in the components of Earth's magnetic field vector, as seen from sensors. The goal of validation is to obtain the magnetic vector (expressed in the NED reference frame) that was initially set in order to compute measurements.

To accomplish this task, the process that led to output data shall be inverted. Hence, the *passive rotation* matrix introduced in Eq. (2.44):

$$\mathbf{R}_{N\to L_i} = \mathbf{R}_{G\to L_i}\,\mathbf{R}_{N\to G} \tag{2.55}$$

has to be inverted, taking into account orthogonality:

$$\mathbf{R}_{L_i\to N} = \mathbf{R}_{N\to L_i}^{-1} = \mathbf{R}_{N\to L_i}^{T} \tag{2.56}$$

Finally, the magnetic vector for validation is obtained:

$$^{N}\mathbf{m}_i^{V} = \mathbf{R}_{L_i\to N}\,\mathbf{m}_i \tag{2.57}$$

2.3.2 Validation through data comparison

The validation procedure is based on the comparison among validation data (that have just been computed) and the (initial) simulated quantities related to TetrAHRS.

The comparison involves the following data:

1. Linear accelerations:

$^{N}\mathbf{a}^{V}_{CM_{Sim},i}$	$\longleftrightarrow$	$^{N}\mathbf{a}_{CM_{Sim}}$
Equation (2.52)		Section 2.1.3

2. Angular rates:

$^{G}\boldsymbol{\omega}_i^{V}$	$\longleftrightarrow$	$^{G}_{N}\boldsymbol{\omega}_G$
Equation (2.54)		Section 2.1.3

3. Earth's magnetic field vector:

$^{N}\mathbf{m}_i^{V}$	$\longleftrightarrow$	$^{N}\mathbf{m}$
Equation (2.57)		Equation (2.43)

One first check is performed taking into consideration an arbitrary trajectory. Then, the process is repeated for a sequence of relevant paths and rotations. They are chosen to test the whole range of motion.

Chapter 3

Kalman Filter

The present thesis aims at combining noisy measures obtained from MEMS sensors and estimate the state of the system. A powerful and flexible method to accomplish this task is offered by the Kalman filter.

This Chapter presents an introduction to linear Kalman filtering, necessary to understand its working principles. Subsequently, the framework at the core of the project is described. The analysis of vectors and matrices justifies the choice of a nonlinear Kalman filter, in particular an Unscented Kalman Filter. Hence, the UKF is described in detail and implemented. Finally, alternatives to the aforementioned filter are proposed.

3.1 Introduction

A Kalman filter is an effective and versatile mathematical procedure for combining noisy sensor outputs to estimate the state of a system with uncertain dynamics (as stated in Reference [13]).

Two main variables can be found in the core of its working mechanism:

1. *Estimated State Vector*, whose components include:

   (a) Variables of interest,

   (b) "Nuisance variables" that are of no intrinsic interest but may be necessary to the estimation process.

   State variables for a specific application must include all those system dynamic variables that are measurable by the sensors. Their components, however, do not have to be those along the sensor input axes.

2. *Covariance Matrix*, which is a measure of estimation uncertainty. The equations used to propagate the covariance matrix (called the *Riccati equation*) model and manage uncertainty. This method allows to take into account how sensor noise and dynamic uncertainty contribute to uncertainty about the estimated system state.

The Kalman filter is an estimator for what is called the *linear-quadratic problem*, which consists in estimating the instantaneous *state* of a linear dynamic system perturbed by *white noise*, by using measurements linearly related to the *state* but corrupted by *white noise*. The resulting estimator is statistically "optimal" with respect to any quadratic function of estimation error [12].

Original applications only included linear dynamics systems with linear sensors; however, later on, the filter was enhanced to be also applied to nonlinear cases. For a better grasp of the idea, the linear formulation will be presented firstly. Reference [13] will be followed in the treatize. Then, the focus will shift to the nonlinear implementation due to the nature of the problem.

The Kalman filter is a two-step process. The two phases are called:

- *Prediction*
- *Correction*

3.1.1 State and covariance correction

The *correction* step makes corrections to an estimate, based on new information obtained from sensor measurements:

$$\underbrace{\hat{\mathbf{x}}(+)}_{\text{corrected}} = \underbrace{\hat{\mathbf{x}}(-)}_{\text{predicted}} + \mathbf{K}_{gain} \left[\underbrace{\mathbf{z}}_{\text{measurements}} - \underbrace{\mathbf{H}\,\hat{\mathbf{x}}(-)}_{\text{predicted measurements}} \right] \tag{3.1}$$

where $\hat{\mathbf{x}}$ is the *estimated state vector*, defined as the maximum likelihood estimate of $\mathbf{x}$. It has two different values:

1. $\hat{\mathbf{x}}(-)$, prior to using the measurements
2. $\hat{\mathbf{x}}(+)$, after using the measurements

While, $\mathbf{z}$ is the *vector of measurements*, which contains measurements coming from sensors. The Kalman gain matrix $\mathbf{K}_{gain}$ is the crown jewel of Kalman filtering. All the effort of solving the matrix Riccati equation is for the sole purpose of computing its "optimal" value, used for correcting the estimate of the state vector.

Finally, the *predicted measurements* are obtained as a linear function of the state vector $\mathbf{x}$ to be estimated plus additive noise with known statistical properties:

$$\mathbf{z}_{predicted} = \mathbf{H}\,\hat{\mathbf{x}}(-) + \underbrace{\mathbf{v}}_{\text{noise}} \tag{3.2}$$

where $\mathbf{H}$ is the measurement sensitivity matrix. In nonlinear application, this expression will be substituted with an appropriate procedure, described later on.

A clarification on noise vector $\mathbf{v}$ has to be made. It is assumed to be additive and to be a zero-mean Gaussian noise with known covariance $\mathbf{R}$:

$$E[\mathbf{z}] = 0 \tag{3.3}$$

$$\mathbf{R} = E[\mathbf{z}\mathbf{z}^T] \tag{3.4}$$

Matrix $\mathbf{R}$ is called *covariance matrix* of sensor noise.

Another covariance matrix $\mathbf{P}$ has to be introduced. It is the *covariance matrix* of *estimation uncertainty*. Similarly to the *estimated state vector*, matrix $\mathbf{P}$ can be distinguished in:

1. $\mathbf{P}(-)$, prior to using the measurements
2. $\mathbf{P}(+)$, after using the measurements

At each iteration, it has to be updated according to the following equation:

$$\mathbf{P}(+) = \mathbf{P}(-) - \mathbf{P}(-)\mathbf{H}^T\left[\mathbf{H}\mathbf{P}(-)\mathbf{H}^T + \mathbf{R}\right]^{-1}\mathbf{H}\mathbf{P}(-) \tag{3.5}$$

From Eq. (3.5), through appropriate manipolations, Eq. (3.1) is obtained, with:

$$\mathbf{K}_{gain} = \mathbf{P}(-)\mathbf{H}^T\left[\mathbf{H}\mathbf{P}(-)\mathbf{H}^T + \mathbf{R}\right]^{-1} \tag{3.6}$$

The detailed derivation is presented in Chapter 7 of Reference [13].

3.1.2 State and covariance prediction

The second phase involved in Kalman filtering is given by the *prediction* step, in which the estimate $\hat{\mathbf{x}}$ and its associated covariance matrix of estimation uncertainty $\mathbf{P}$ are propagated from one time instant to another. This is the part where the dynamics of the underlying physical processes come into play.

The *state* of a dynamic process is a vector of variables that completely specify enough of the initial boundary value conditions for propagating the trajectory of the dynamic process forward in time. The procedure for propagating that solution forward in time is called *state prediction*. The model for propagating the covariance matrix of estimation uncertainty is derived from the model used for propagating the state vector.

State space models

State space models definable as systems of linear first-order differential equations can be represented more compactly in terms of a *state vector* $\mathbf{x}$, *dynamic coeffcient matrix* $\mathbf{F}$, and *dynamic disturbance vector* $\mathbf{w}$:

$$\frac{d}{dt}\mathbf{x} = \mathbf{F}\mathbf{x} + \mathbf{w} \tag{3.7}$$

given as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \tag{3.8}$$

The forward solution in time is:

$$\mathbf{x}(t) = \exp\left(\int_{t_0}^{t} \mathbf{F}(s)ds\right)\left[\mathbf{x}(t_0) + \int_{t_0}^{t} \exp\left(-\int_{t_0}^{s} \mathbf{F}(r)dr\right)\mathbf{w}(s)ds\right] \tag{3.9}$$

where $\mathbf{x}(t_0)$ is the initial value of the state vector $\mathbf{x}$ for $t > t_0$.

Discrete systems

Measurements are the outputs of sensors sampled at discrete times $\cdots < t_{k-1} < t_k < t_{k+1} < \cdots$. The Kalman filter uses these values to correct the estimate of the state at those discrete times.

If we let $\ldots$, $\mathbf{x}_{k-1}$, $\mathbf{x}_k$, $\mathbf{x}_{k+1}$, $\ldots$ be the corresponding state vector values of the linear dynamic system at those discrete times, then each of these values can be determined from the previous value by using Eq. (3.9):

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{3.10}$$

with $\mathbf{\Phi}_{k-1}$ being the *state transition matrix* for the dynamic system defined by $\mathbf{F}$. It represents the known dynamic behavior of the system. Note that $\mathbf{\Phi}$ depends only on $\mathbf{F}$, and not on the dynamic disturbance function $\mathbf{w}(t)$. Whereas, the noise vectors $\mathbf{w}_k$ are the discrete-time analog of the dynamic disturbance function $\mathbf{w}(t)$. They depend upon $\mathbf{F}$ and $\mathbf{w}$.

$$\mathbf{\Phi}_{k-1} = \exp\left(\int_{t_{k-1}}^{t_k} \mathbf{F}(s)ds\right) \tag{3.11}$$

$$\mathbf{w}_{k-1} = \mathbf{\Phi}_k \int_{t_{k-1}}^{t_k} \exp\left(-\int_{t_{k-1}}^{t} \mathbf{F}(s)ds\right)\mathbf{w}(t)dt \tag{3.12}$$

Covariance prediction models

A *stochastic process* is a model for the evolution over time of a probability distribution. In Kalman filtering, this can be viewed as the probability distribution of the true state of a dynamic process. When the underlying probability distribution is Gaussian, the distribution is completely specified by its *mean* and its *covariance*.

The *mean* will be the estimated value of the state vector while the *covariance matrix* represents the mean-squared uncertainty in the estimate. The time update equations implemented in the Kalman filter propagate the estimated value and its associated mean-squared uncertainty forward in time.

A *zero-mean white Gaussian* noise process in discrete time is a sequence of *independent* samples $\ldots$, $\mathbf{w}_{k-1}$, $\mathbf{w}_k$, $\mathbf{w}_{k+1}$, $\ldots$ from a normal probability distribution $\mathcal{N}(0, \mathbf{Q}_k)$ with zero mean and known finite covariances $\mathbf{Q}_k$. *Covariance matrices* $\mathbf{Q}_k$ represent the unknown random disturbances.

Sampling is called *independent* if the expected values of outer products

$$E\left[\mathbf{w}_i \mathbf{w}_j{}^T\right] = \begin{cases} 0 & i \neq j \\ \mathbf{Q}_i & i = j \end{cases} \tag{3.13}$$

for all integer indices i and j of the random process.

Zero-mean white Gaussian noise processes are the fundamental random processes used in Kalman filtering. However, it is *not* necessary that all noise sources in the modeled sensors and dynamic systems are zero-mean white Gaussian noise processes.

The linear model parameters $\mathbf{\Phi}$ and $\mathbf{Q}$ can be used to calculate how the discrete-time process variables $\boldsymbol{\mu}$ (*mean*) and $\mathbf{P}$ (*covariance*) evolve over time:

$$\boldsymbol{\mu}_k = \mathbf{\Phi}_{k-1}\,\boldsymbol{\mu}_{k-1} \tag{3.14}$$

$$\mathbf{P}_k = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1} \tag{3.15}$$

Futher details can be found in Chapter 7 of Reference [13].

3.1.3 Summary of linear Kalman filter equations

The discrete-time equations that are implemented in the linear Kalman filter are summarized in the following Table:

Table 3.1: Summary of linear Kalman filter equations.

Predictor (Time updates)	
Predicted state vector:	$\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_k\,\hat{\mathbf{x}}_{k-1}(+)$
Predicted covariance matrix:	$\mathbf{P}_k(-) = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}(+)\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1}$
Corrector (Measurement updates)	
Kalman gain:	$\overline{\mathbf{K}}_k = \mathbf{P}_k(-)\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1}$
Corrected state estimate:	$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \overline{\mathbf{K}}_k\left[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-)\right]$
Corrected covariance matrix:	$\mathbf{P}_k(+) = \mathbf{P}_k(-) - \overline{\mathbf{K}}_k\mathbf{H}_k\mathbf{P}_k(-)$

3.2 Filter framework

Once the working principles of the filter have been described, the focus shall be shifted to the specific case that is being studied. All the key variables will be introduced and choices on the type of filter will be made accordingly.

3.2.1 State vector

The state vector should contain all the variables of interest. In addition, parameters that are required by the formulation can be inserted. Twelve elements of the state have been chosen:

1. Linear acceleration vector components along G axes

2. Angular velocity vector components around G axes

3. Angular acceleration vector components around G axes

4. Euler angles (expressing the attitude with respect to the NED frame)

The state vector is structured as follows:

$$\mathbf{x} = \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \\ \omega_X \\ \omega_Y \\ \omega_Z \\ \alpha_X \\ \alpha_Y \\ \alpha_Z \\ \varphi \\ \vartheta \\ \psi \end{bmatrix} \tag{3.16}$$

Angular acceleration components α_X, α_Y and α_Z have been introduced to take into account the effect of the rotation of the body. Looking at Eq. (2.34), it can be observed that the angular acceleration vector multiplies the vector that identifies the position of the sensor with respect to the CM.

Since miniaturization is pursued, the size of the system and, at the same rate, the magnitude of position vector $\mathbf{s}_i$ are going to reduce progressively. As a result, the contribution due to rotation becomes less relevant up to a point where it could be neglected. In that case, the three elements corresponding to angular acceleration could be removed. The state transition matrix, reported in Eq. (3.22), and the covariance matrix $\mathbf{Q}$ (introduced in Section 3.2.3) shall be modified accordingly.

3.2.2 State transition matrix

As presented in Section 3.1.2, the state of the system can be predicted (from a time instant to the following one) thanks to a *state transition matrix*:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}(\mathbf{x}_k)\,\mathbf{x}_k + \underbrace{\mathbf{w}_k}_{\text{dynamic disturbance}} \tag{3.17}$$

The *state transition matrix* $\mathbf{\Phi}(\mathbf{x}_k)$ is obtained from the combination of mathematical relationships that exist among the state variables:

– Angular velocity vector components ω_X, ω_Y and ω_Z are integrals of angular acceleration components α_X, α_Y and α_Z.

– Euler angles are related to the components of angular velocity vector through their derivatives [37]:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\varphi\tan\vartheta & \cos\varphi\tan\vartheta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \dfrac{\sin\varphi}{\cos\vartheta} & \dfrac{\cos\varphi}{\cos\vartheta} \end{bmatrix} \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix} \tag{3.18}$$

Assuming that angular velocity components remain constant during the time interval between one time instant and the following, Euler angles can be computed through numerical integration:

$$\varphi_{k+1} = \varphi_k + \dot{\varphi}_k \, \Delta T \tag{3.19}$$

$$\vartheta_{k+1} = \vartheta_k + \dot{\vartheta}_k \, \Delta T \tag{3.20}$$

$$\psi_{k+1} = \psi_k + \dot{\psi}_k \, \Delta T \tag{3.21}$$

Therefore, collecting all equations, the *state transition matrix* becomes:

$$\boldsymbol{\Phi}(\mathbf{x}_k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta T & \sin\varphi\tan\vartheta\Delta T & \cos\varphi\tan\vartheta\Delta T & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos\varphi\Delta T & -\sin\varphi\Delta T & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dfrac{\sin\varphi}{\cos\vartheta}\Delta T & \dfrac{\cos\varphi}{\cos\vartheta}\Delta T & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

Particular attention must be paid to the elements of the matrix that contain the pitch angle ϑ. In fact, when its value gets close to $\pm\frac{\pi}{2}$, problems occur with both:

– $\tan\vartheta$, which characterizes elements $\boldsymbol{\Phi}_{10,5}$ and $\boldsymbol{\Phi}_{10,6}$

– $\cos\vartheta$, at the denominator of terms $\boldsymbol{\Phi}_{12,5}$ and $\boldsymbol{\Phi}_{12,6}$

As a result, values of cited elements become extremely high (compared to the corresponding values associated with lower pitch angles). This phenomenon can cause problems to the filter, as will be shown in Section 5.1.2. Different solutions to the issue can be proposed:

1. Implementing an algorithm that identifies the phenomenon (e.g., monitoring the values of ϑ and the terms that contain it) and acts on the filter in order to minimize the consequences.

2. Replacing the parametrization in terms of Euler angles with quaternions. This choice would lead to the removal of trigonometric functions, solving the root of the problem. On the other hand, it would imply losing the possibility to formulate the *covariance matrix* $\mathbf{Q}$, as described in the following paragraph.

3.2.3 Covariance matrix Q

Uncertainties related to the state are managed by *covariance matrix* **Q**. It is bulit on the basis of a vector of uncertainties, whose values can be arbitrarily set.

$$\mathbf{q}_{state} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{Linear accelerations: } \ddot{X}, \ddot{Y} \text{ and } \ddot{Z} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{Angular accelerations: } \alpha_X, \alpha_Y \text{ and } \alpha_Z \end{matrix} \tag{3.23}$$

Each q_i is related to the process noise covariance expressed in continuous time, associated with the corresponding element of the state. The discretized equivalent will be inserted in the **Q** matrix, following the criterion reported in Reference [12].

Once the uncertainty on angular accelerations is defined, the one on angular velocities and Euler angles can be derived according to their relationships. In fact, angular velocities are integrals of angular accelerations and Euler angles are integrals of the former, minus a rotation matrix. On these grounds, the covariance matrix **Q** is formulated:

$$\mathbf{Q} = \begin{bmatrix} q_1\Delta T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2\Delta T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3\Delta T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4\frac{\Delta T^3}{6} & 0 & 0 & q_4\frac{\Delta T^2}{2} & 0 & 0 & q_4\frac{\Delta T^3}{6} & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5\frac{\Delta T^3}{6} & 0 & 0 & q_5\frac{\Delta T^2}{2} & 0 & 0 & q_5\frac{\Delta T^3}{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6\frac{\Delta T^3}{6} & 0 & 0 & q_6\frac{\Delta T^2}{2} & 0 & 0 & q_6\frac{\Delta T^3}{6} \\ 0 & 0 & 0 & q_4\frac{\Delta T^2}{2} & 0 & 0 & q_4\Delta T & 0 & 0 & q_4\frac{\Delta T^4}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5\frac{\Delta T^2}{2} & 0 & 0 & q_5\Delta T & 0 & 0 & q_5\frac{\Delta T^4}{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6\frac{\Delta T^2}{2} & 0 & 0 & q_6\Delta T & 0 & 0 & q_6\frac{\Delta T^4}{8} \\ 0 & 0 & 0 & q_4\frac{\Delta T^3}{6} & 0 & 0 & q_4\frac{\Delta T^4}{8} & 0 & 0 & q_4\frac{\Delta T^5}{20} & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5\frac{\Delta T^3}{6} & 0 & 0 & q_5\frac{\Delta T^4}{8} & 0 & 0 & q_5\frac{\Delta T^5}{20} & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6\frac{\Delta T^3}{6} & 0 & 0 & q_6\frac{\Delta T^4}{8} & 0 & 0 & q_6\frac{\Delta T^5}{20} \end{bmatrix}$$

The implementation of the parametrization in terms of quaternions would require a completely new approach to the definition of the matrix.

3.2.4 Measurement vector

The *measurement vector* includes the outputs obtained from MEMS sensors. Since each sensor provides 9 measurements and the configuration consists of 4 sensors, the vector will have 36 elements. They are computed according to the *measurement equations* (defined in Section 3.2.5), on the basis of the *estimated state*. The vector is organized in the following way:

$$\mathbf{z} = \begin{bmatrix} n_{x,1} \\ n_{y,1} \\ n_{z,1} \\ \omega_{x,1} \\ \omega_{y,1} \\ \omega_{z,1} \\ m_{x,1} \\ m_{y,1} \\ m_{z,1} \\ \vdots \\ n_{x,4} \\ n_{y,4} \\ n_{z,4} \\ \omega_{x,4} \\ \omega_{y,4} \\ \omega_{z,4} \\ m_{x,4} \\ m_{y,4} \\ m_{z,4} \end{bmatrix} \tag{3.24}$$

3.2.5 Measurement equations

Measurements equations allow to compute *predicted measurements* from the value of the state. They are based on the same logic that was followed in the simulation phase. The only difference lies in the variables that are used to obtain the outputs.

Accelerations

Proper accelerations, measured by tri-axial accelerometers, are calculated from the estimated accelerations in the G frame, properly rotated according to the attitude:

$$\begin{bmatrix} n_{x,i} \\ n_{y,i} \\ n_{z,i} \end{bmatrix}_k = \mathbf{R}_{G \to L_i} \, \hat{\mathbf{a}}_{total_k} - \mathbf{R}_{G \to L_i} \, \mathbf{R}_{N \to G} \, {}^N\mathbf{g} \tag{3.25}$$

where the total acceleration is:

$$\hat{\mathbf{a}}_{total_k} = \left[\hat{\mathbf{a}}_{G_k} + \hat{\boldsymbol{\alpha}}_{G_k} \times {}^G\mathbf{s}_i + \hat{\boldsymbol{\omega}}_{G_k} \times \left(\hat{\boldsymbol{\omega}}_{G_k} \times {}^G\mathbf{s}_i \right) \right] \tag{3.26}$$

with

$$\hat{\mathbf{a}}_{G_k} = \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix}_k \quad \hat{\boldsymbol{\omega}}_{G_k} = \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}_k \quad \hat{\boldsymbol{\alpha}}_{G_k} = \begin{bmatrix} \alpha_X \\ \alpha_Y \\ \alpha_Z \end{bmatrix}_k \tag{3.27}$$

Matrices $\mathbf{R}_{N \to G}$, $\mathbf{R}_{G \to L_i}$ and vectors ${}^N\mathbf{g}$, $\mathbf{s}_i$ were defined, respectively, in Eq. (2.32), Eq. (2.38), Eq. (2.40) and Section 2.1.2.

It is worth mentioning that matrix $\mathbf{R}_{N \to G}$ is evaluated on the basis of the *estimated* Euler angles at the k^{th} time instant. In fact, the attitude of the body is taken in account through this matrix. Differently, $\mathbf{R}_{G \to L_i}$ is constant.

Angular rates

Components of the angular velocity vector along sensor axes are obtained from the *estimated* angular velocity $\hat{\boldsymbol{\omega}}_{G_k}$:

$$\begin{bmatrix} \omega_{x,i} \\ \omega_{y,i} \\ \omega_{z,i} \end{bmatrix}_k = \mathbf{R}_{G \to L_i} \, \hat{\boldsymbol{\omega}}_{G_k} \tag{3.28}$$

which was defined in Eq. (3.27).

Components of Earth's magnetic field vector

The components of Earth's magnetic field vector, expressed in the sensor local frames, are determined with an appropriate rotation of the reference magnetic vector that was introduced in Eq. (2.43):

$$\begin{bmatrix} m_{x,i} \\ m_{y,i} \\ m_{z,i} \end{bmatrix}_k = \mathbf{R}_{G \to L_i} \, \mathbf{R}_{N \to G} \, {}^{N}\mathbf{m} \tag{3.29}$$

The dependence on the *estimated* state vector is included in the $\mathbf{R}_{N \to G}$ matrix.

3.2.6 Covariance matrix R

The *covariance matrix* $\mathbf{R}$ accounts for sensor noise. Following the definition reported in Section 3.1.1, it can be formulated as a 36×36 diagonal matrix. The variances related to each axis of the four sensors will be placed on its diagonal.

$$\mathbf{R} = \begin{bmatrix} \sigma^2_{n_x,1} & & & & & & & & & & \\ & \sigma^2_{n_y,1} & & & & & & & & & \\ & & \sigma^2_{n_z,1} & & & & & & & & \\ & & & \sigma^2_{\omega x,1} & & & & & & & \\ & & & & \sigma^2_{\omega y,1} & & & & & & \\ & & & & & \sigma^2_{\omega z,1} & & & & & \\ & & & & & & \sigma^2_{m_x,1} & & & & \\ & & & & & & & \sigma^2_{m_y,1} & & & \\ & & & & & & & & \sigma^2_{m_z,1} & & \\ & & & & & & & & & \ddots & \end{bmatrix} \tag{3.30}$$

3.3 Unscented Kalman Filter

A brief analysis of the equations that govern the problem shows that there are relevant non-linearities involved. Moreover, the *state transition matrix* is state-dependent; this fact further supports the choice of a non-linear formulation. As described in Section 3.1, the Kalman filter is suitable for linear systems. On the contrary, the Extended Kalman Filter (EKF) was later developed to apply the Kalman filter framework to non-linear systems. However, it is adequate until a certain degree of non-linearity.

At this juncture, the Unscented Kalman Filter (UKF) comes into play. It is a popular estimation technique for so-called "highly non-linear" dynamic systems. According to Reference [2], it is recommended to use the EKF for a "mildly non-linear" system while the UKF performs better (than the EKF) with "highly non-linear" systems. This solution is also convenient for taking into account other sources of non-linearity inherent in MEMS sensors.

Reference [15] offers a detailed insight in the working mechanism of the UKF. It will be followed for both the description and the formulation of the filter.

The Unscented Kalman Filter (UKF) has its origins in 1995. It was presented in Seattle as a new method for implementing a linearized approximation to non-linear state estimation that, unlike the EKF, did not require the explicit calculation of Jacobians. The idea of "approximating a Gaussian distribution efficiently by a discrete distribution (of vector points) allows a non-linear transformation to be applied to each of the points independently."

The method is based on a specified set of vector points that are symmetric about the mean of a multidimensional Gaussian distribution. The points are selected so that, when they are used as inputs to a non-linear transformation, a weighted sum of the resulting output points produces a better estimate of the transformed mean than the one that is obtained with an EKF.

3.3.1 Introduction to Sigma Points Kalman filters

When the Kalman filter is applied to non-linear systems, the assumptions of *linearity* presented in Section 3.1 are no longer valid. Therefore, Eq. (3.2) and Eq. (3.10) shall be replaced by non-linear expressions. To this end, a new definition of the system is required.

Consider a dynamic process of the form:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \tag{3.31}$$

with observations generated from the observation process:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \tag{3.32}$$

Two functions have been introduced:

1. *Deterministic transition function* (matrix) $\mathbf{f}_{k-1}$ that moves the state $\mathbf{x}$ from time t_{k-1} to time t_k;
2. *Deterministic observation function* $\mathbf{h}_k$ that links the state vector with the observation.

They serve the same purpose as the *state transition matrix* $\mathbf{\Phi}$ and matrix $\mathbf{H}$ in the linear case.

By analogy with the linear formulation, $\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to be zero-mean Gaussian random variables independent of both $\mathbf{x}_k$ and $\mathbf{z}_k$. Under these assumptions,

it can be demonstrated that prediction estimation equations reduce to *Gaussian-weighted integrals*, which can be solved numerically. The solutions lead to four classes of Kalman filters. One of them is the *Sigma Point Class*, that includes the Uscented Kalman Filter (UKF). The distinctive feature is that non-linear functions are expanded in more general polynomials such that the integrals reduce to weighted summations over a set of deterministic vector points, called *sigma points*.

Consider the general multidimensional Gaussian-weighted integral:

$$\hat{\mathbf{g}}(\mathbf{x}) = \int \tilde{\mathbf{g}}(\mathbf{c})\, \mathcal{N}(\mathbf{c}; \mathbf{0}, \mathbf{I})\, d\mathbf{c} \tag{3.33}$$

where $\mathcal{N}(\mathbf{c}; \mathbf{0}, \mathbf{I})$ represents the multidimensional extension of a generic scalar stochastic variable x, governed by a *Gaussian probability density function*: $p(x) = \mathcal{N}(x; \hat{x}, \sigma^2)$. $\hat{x}$ is the first moment or *mean value* and σ^2 is the second moment or *variance*. σ is usually called the *standard deviation*.

As demonstrated in [15], the integral can be approximated by a discrete sum:

$$\hat{\mathbf{g}}(\mathbf{x}) \simeq \sum_{j=1}^{n_s} w_j\, \tilde{\mathbf{g}}\left(\mathbf{c}^{(j)}\right) \tag{3.34}$$

where $\mathbf{c}^{(j)}$ represents the set of *sigma points*:

$$\left\{\mathbf{c}^{(j)},\ j = 1, 2, \ldots, 1, \ldots, n_s\right\} \tag{3.35}$$

Weights w_j are computed on the basis of the following relations:

$$\sum_{j=1}^{n_s} w_j = \int_{\mathbb{R}^k} \mathcal{N}(\mathbf{c}; \mathbf{0}, \mathbf{I})\, d\mathbf{c} = 1 \tag{3.36}$$

$$\sum_{j=1}^{n_s} w_j\, \mathbf{c}^{(j)} = \int_{\mathbb{R}^k} \mathbf{c}\, \mathcal{N}(\mathbf{c}; \mathbf{0}, \mathbf{I})\, d\mathbf{c} = \mathbf{0} \tag{3.37}$$

$$\sum_{j=1}^{n_s} w_j\, \mathbf{c}^{(j)}\, \mathbf{c}^{(j)^T} = \int_{\mathbb{R}^k} \mathbf{c}\, \mathbf{c}^T\, \mathcal{N}(\mathbf{c}; \mathbf{0}, \mathbf{I})\, d\mathbf{c} = \mathbf{I} \tag{3.38}$$

3.3.2 UKF: Choice of Sigma Points

The formulation of the UKF considers a set of vector points $\left\{\mathbf{c}^{(j)}\right\}$ that has the same dimension as the state vector. The points form symmetric pairs at a radius of $\pm q$ along each Cartesian axes and a single point at the origin. Hence, the total number of integration points is:

$$n_s = 2n_x + 1 \tag{3.39}$$

where n_x is the dimension of the state vector. The problem that is being studied numbers 12 elements of the state; thus, $n_x = 12$ and $n_s = 25$.

With the exception of the point at the origin, all vector points lie on the same n_x-dimensional hypersphere of radius q. Therefore, the set can be described by:

$$\mathbf{c}^{(j)} = q\,\mathbf{r}^{(j)} \tag{3.40}$$

with the unit vector set $\mathbf{r} = [1] \in \mathbb{R}^{n_x}$ all lying on the axes of a unit hypersphere. Vector set $\mathbf{r}$ can be defined as:

$$\mathbf{r} = \begin{cases} [0] \in \mathbb{R}^{n_x} & j = 0 \\ [1] \in \mathbb{R}^{n_x} & j = 1, \ldots, 2n_x \end{cases} \tag{3.41}$$

Exploiting the set of Eq. (3.36), (3.37) and (3.38), with the addition of Eq. (3.40), values of *sigma points* and weights can be determined:

$$\mathbf{c}^{(j)} = q\,\mathbf{r}^{(j)} = \sqrt{\frac{n_x}{1 - w_0}}\,\mathbf{r}^{(j)},\ j = 0, 1, \ldots, 2n_x \tag{3.42}$$

$$w_j = \begin{cases} w_0 & j = 0 \\ \frac{1-w_0}{2n_x} & j = 1, \ldots, 2n_x \end{cases} \tag{3.43}$$

with w_0 being a free parameter. In order to satisfy Eq. (3.36) and ensure that all weights are positive, the following restriction must be imposed:

$$0 \leq w_0 \leq 1 \tag{3.44}$$

Since the point at the origin was added in an arbitrary way, there is no loss of generality if $w_0 = 0$ is taken. This choice leads to the removal of the *sigma point* at the origin from the integration sum.

3.3.3 UKF: State vector and covariance prediction

The state prediction equation can be rewritten in terms of a *Gaussian-weighted integral*, which, in turn, can be expressed as a weighted sum:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \int \tilde{\mathbf{f}}(\mathbf{c})\,\mathcal{N}(\mathbf{c}; \mathbf{0}, \mathbf{I})\,d\mathbf{c} \\ &= \sum_{j=0}^{2n_x} w_j\,\tilde{\mathbf{f}}\left(\mathbf{c}^{(j)}\right) \end{aligned} \tag{3.45}$$

In order to determine $\tilde{\mathbf{f}}\left(\mathbf{c}^{(j)}\right)$, the concept of *multidimensional affine transformation* shall be introduced. Starting from a generic *one-dimensional* Gaussian case,

$$x = \hat{x} + \sigma c \tag{3.46}$$

the *affine transformation* moves the coordinate origin to $\hat{x}$ and scales the standard deviation so that $\mathcal{N}(x; \hat{x}, \sigma^2) \rightarrow \mathcal{N}(c; 0, 1)/\sigma$. This way, c becomes a zero-mean unit standard deviation random variable.

The general moment integral is transformed accordingly:

$$E[f(x)] = \int_{-\infty}^{\infty} f(x)\,\mathcal{N}(x;\hat{x},\sigma^2)\,dx = \int_{-\infty}^{\infty} \tilde{f}(c)\,\mathcal{N}(c;0,1)\,dc \tag{3.47}$$

letting

$$\tilde{f}(c) = f(\hat{x} + \sigma c) \tag{3.48}$$

The same logic can be extended to the multidimensional case:

$$\mathbf{x} = \hat{\mathbf{x}} + \mathbf{D}\mathbf{c} \tag{3.49}$$

where $\mathbf{D}$ is defined by the matrix square root equation:

$$\mathbf{P^{xx}} = \mathbf{D}\mathbf{D}^T \tag{3.50}$$

Finally leading to the identity:

$$\mathbf{D} = [\mathbf{P^{xx}}]^{\frac{1}{2}} \tag{3.51}$$

Matrix $\mathbf{D}$ can be the Cholesky factor of $\mathbf{P^{xx}}$ or the principal (positive definite) root of $\mathbf{P^{xx}}$. Actually, any matrix that satisfies Eq. (3.50) will suffice.

Similarly to Eq. (3.47):

$$\begin{aligned} E[\mathbf{f}(\mathbf{x})] &= \int_{-\infty}^{\infty} \mathbf{f}(\mathbf{x})\,\mathcal{N}(\mathbf{x};\hat{\mathbf{x}},\mathbf{P^{xx}})\,d\mathbf{x} \\ &= \int_{-\infty}^{\infty} \tilde{\mathbf{f}}(\mathbf{c})\,\mathcal{N}(\mathbf{c};\mathbf{0},\mathbf{I})\,d\mathbf{c} \end{aligned} \tag{3.52}$$

by applying the *multidimensional affine transformation*:

$$\tilde{\mathbf{f}}(\mathbf{c}) = \mathbf{f}(\hat{\mathbf{x}} + \mathbf{D}\mathbf{c}) \tag{3.53}$$

Further explanations can be found in Chapter 2 of Reference [15].

State vector prediction

Defining the UKF *sigma points* as

$$\mathcal{X}^{(j)}_{k-1|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{D}_{k-1|k-1}\,\mathbf{c}^{(j)},\ j = 0, 1, \ldots, 2n_x \tag{3.54}$$

the value of $\tilde{\mathbf{f}}\left(\mathbf{c}^{(j)}\right)$ can be determined:

$$\tilde{\mathbf{f}}\left(\mathbf{c}^{(j)}\right) = \mathbf{f}\left(\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{D}_{k-1|k-1}\,\mathbf{c}^{(j)}\right),\ j = 0, 1, \ldots, 2n_x \tag{3.55}$$

Finally, the state can be predicted by taking advantage of Eq. (3.45):

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{j=0}^{2n_x} w_j\,\tilde{\mathbf{f}}\left(\mathbf{c}^{(j)}\right) = \sum_{j=0}^{2n_x} w_j\,\mathbf{f}\left(\mathcal{X}^{(j)}_{k-1|k-1}\right) \tag{3.56}$$

where $\mathbf{c}^{(j)}$ and w_j were defined, respectively, in Eq. (3.42) and (3.43). While, matrix $\mathbf{D}_{k-1|k-1}$ was introduced in Eq. (3.51):

$$\mathbf{D}_{k-1|k-1} = [\mathbf{P^{xx}}]^{\frac{1}{2}} \tag{3.57}$$

State vector covariance prediction

Prediction of *covariance matrix* $\mathbf{P}^{\mathbf{xx}}_{k|k-1}$ requires the following assumption:

$$\tilde{\mathbf{g}}(\mathbf{c}) = \left[\tilde{\mathbf{f}}(\mathbf{c}) - \hat{\mathbf{x}}_{k|k-1}\right]\left[\tilde{\mathbf{f}}(\mathbf{c}) - \hat{\mathbf{x}}_{k|k-1}\right]^T \tag{3.58}$$

After substituting $\tilde{\mathbf{g}}(\mathbf{c})$ in place of $\tilde{\mathbf{f}}(\mathbf{c})$, Eq. (3.56) becomes the *state covariance* prediction equation:

$$\begin{aligned}\mathbf{P}^{\mathbf{xx}}_{k|k-1} &= \int_{-\infty}^{\infty} \tilde{\mathbf{g}}(\mathbf{c})\,\mathcal{N}(\mathbf{c};\mathbf{0},\mathbf{I})\,d\mathbf{c} + \mathbf{Q} \\ &= \sum_{j=0}^{2n_x} w_j\,\tilde{\mathbf{g}}\left(\mathbf{c}^{(j)}\right) + \mathbf{Q} \\ &= \sum_{j=0}^{2n_x} w_j\,\left[\mathbf{f}\left(\mathcal{X}^{(j)}_{k-1|k-1}\right) - \hat{\mathbf{x}}_{k|k-1}\right]\left[\mathbf{f}\left(\mathcal{X}^{(j)}_{k-1|k-1}\right) - \hat{\mathbf{x}}_{k|k-1}\right]^T + \mathbf{Q}\end{aligned} \tag{3.59}$$

3.3.4 UKF: Observation vector and covariance prediction

The prediction process for the observation vector and its related *covariance matrices* retraces the same steps that were followed with the state vector:

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{j=0}^{2n_x} w_j\,\mathbf{h}\left(\mathcal{X}^{(j)}_{k|k-1}\right) \tag{3.60}$$

$$\mathbf{P}^{\mathbf{zz}}_{k|k-1} = \sum_{j=0}^{2n_x} w_j\,\left[\mathbf{h}\left(\mathcal{X}^{(j)}_{k|k-1}\right) - \hat{\mathbf{z}}_{k|k-1}\right]\left[\mathbf{h}\left(\mathcal{X}^{(j)}_{k|k-1}\right) - \hat{\mathbf{z}}_{k|k-1}\right]^T + \mathbf{R} \tag{3.61}$$

$$\mathbf{P}^{\mathbf{xz}}_{k|k-1} = \sum_{j=0}^{2n_x} w_j\,\left[\mathbf{f}\left(\mathcal{X}^{(j)}_{k-1|k-1}\right) - \hat{\mathbf{x}}_{k|k-1}\right]\left[\mathbf{h}\left(\mathcal{X}^{(j)}_{k|k-1}\right) - \hat{\mathbf{z}}_{k|k-1}\right]^T \tag{3.62}$$

with

$$\mathcal{X}^{(j)}_{k|k-1} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{D}_{k|k-1}\,\mathbf{c}^{(j)},\ j = 0, 1, \ldots, 2n_x \tag{3.63}$$

3.3.5 UKF: Update process

The first step consists in computing the *Kalman gain* matrix:

$$\mathbf{K}_{gain} = \mathbf{P}^{\mathbf{xz}}_{k|k-1}\left(\mathbf{P}^{\mathbf{zz}}_{k|k-1}\right)^{-1} \tag{3.64}$$

Once the gain is determined, the update can be accomplished:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{gain}\left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\right) \tag{3.65}$$

$$\mathbf{P}^{\mathbf{xx}}_{k|k} = \mathbf{P}^{\mathbf{xx}}_{k|k-1} - \mathbf{K}_{gain}\,\mathbf{P}^{\mathbf{zz}}_{k|k-1}\,\mathbf{K}^T_{gain} \tag{3.66}$$

where $\mathbf{z}_k$ collects measurements coming from the sensors, at time k.

3.4 Implementation of the UKF

The Unscented Kalman Filter was implemented on the basis of the work done by Brandl Alberto and De Vivo Francesco (in References [7] and [8]). Their multipurpose Kalman Filter simulation platform was modified and adjusted to the problem being studied.

Firstly, the equations relative to the UKF were taken and cross-checked with the theory that was presented in Section 3.3. Secondly, the main processing code was streamlined and re-adapted to the specific application. To the contrary, the rest of the formulation, including all vectors and matrices described in Section 3.2, were created from scratch. Finally, the structure of the code and the user interface were entirely redesigned.

3.4.1 Summary of the code

Equations presented in Section 3.3 were implemented in the code. By reorganizing them in a schematic structure, the following phases are obtained:

1. **Filter initialization:**

$$\text{Setting } 0 \leq w_0 \leq 1$$
$$\text{Initializing } \hat{\mathbf{x}}_0 \text{ and } \mathbf{P}_0^{\mathbf{xx}}$$

2. **Computation of sigma points and weights:**

$$\mathbf{c}^{(j)} = q\,\mathbf{r}^{(j)} = \sqrt{\tfrac{n_x}{1-w_0}}\,\mathbf{r}^{(j)},\ j = 0, 1, \ldots, 2n_x \tag{3.67}$$

$$w_j = \begin{cases} w_0 & j = 0 \\ \frac{1-w_0}{2n_x} & j = 1, \ldots, 2n_x \end{cases} \tag{3.68}$$

3. **State vector and covariance prediction:**

$$\mathcal{X}_{k-1|k-1}^{(j)} = \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{D}_{k-1|k-1}\,\mathbf{c}^{(j)},\ j = 0, 1, \ldots, 2n_x \tag{3.69}$$

$$\hat{\mathbf{x}}_{k|k-1} = \textstyle\sum_{j=0}^{2n_x} w_j\,\mathbf{f}\left(\mathcal{X}_{k-1|k-1}^{(j)}\right) \tag{3.70}$$

$$\mathbf{P}_{k|k-1}^{\mathbf{xx}} = \textstyle\sum_{j=0}^{2n_x} w_j\,\left[\mathbf{f}\left(\mathcal{X}_{k-1|k-1}^{(j)}\right) - \hat{\mathbf{x}}_{k|k-1}\right]\left[\mathbf{f}\left(\mathcal{X}_{k-1|k-1}^{(j)}\right) - \hat{\mathbf{x}}_{k|k-1}\right]^T + \mathbf{Q} \tag{3.71}$$

4. **Observation vector and covariance prediction:**

$$\mathcal{X}_{k|k-1}^{(j)} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{D}_{k|k-1}\,\mathbf{c}^{(j)},\ j = 0, 1, \ldots, 2n_x \tag{3.72}$$

$$\hat{\mathbf{z}}_{k|k-1} = \textstyle\sum_{j=0}^{2n_x} w_j\,\mathbf{h}\left(\mathcal{X}_{k|k-1}^{(j)}\right) \tag{3.73}$$

$$\mathbf{P}_{k|k-1}^{\mathbf{zz}} = \textstyle\sum_{j=0}^{2n_x} w_j\,\left[\mathbf{h}\left(\mathcal{X}_{k|k-1}^{(j)}\right) - \hat{\mathbf{z}}_{k|k-1}\right]\left[\mathbf{h}\left(\mathcal{X}_{k|k-1}^{(j)}\right) - \hat{\mathbf{z}}_{k|k-1}\right]^T + \mathbf{R} \tag{3.74}$$

$$\mathbf{P}_{k|k-1}^{\mathbf{xz}} = \textstyle\sum_{j=0}^{2n_x} w_j\,\left[\mathbf{f}\left(\mathcal{X}_{k-1|k-1}^{(j)}\right) - \hat{\mathbf{x}}_{k|k-1}\right]\left[\mathbf{h}\left(\mathcal{X}_{k|k-1}^{(j)}\right) - \hat{\mathbf{z}}_{k|k-1}\right]^T \tag{3.75}$$

5. **Kalman filter update:**

$$\mathbf{K}_{gain} = \mathbf{P}^{\mathbf{xz}}_{k|k-1} \left(\mathbf{P}^{\mathbf{zz}}_{k|k-1} \right)^{-1} \tag{3.76}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{gain} \left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \right) \tag{3.77}$$

$$\mathbf{P}^{\mathbf{xx}}_{k|k} = \mathbf{P}^{\mathbf{xx}}_{k|k-1} - \mathbf{K}_{gain} \, \mathbf{P}^{\mathbf{zz}}_{k|k-1} \, \mathbf{K}^{T}_{gain} \tag{3.78}$$

6. **Storing results:**

Storing $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}^{\mathbf{xx}}_{k|k}$ in appropriate container

Steps no. $3 \rightarrow 6$ shall be repeated for each instant of time.

3.5 Alternatives to the UKF

The *Sigma Points Class* of Kalman filters, besides the UKF, includes:

1. Spherical Simplex Kalman Filter (SSKF)
2. Gauss-Hermite Kalman Filter (GHKF)
3. Monte Carlo Kalman Filter (MCKF)

They all share the same structure but they differ in the formulation of $\mathbf{c}^{(j)}$ and w_j. In addition, they pose different ways to approximate non-linear functions. Each type of filter will be described, getting information from Reference [15].

3.5.1 Spherical Simplex Kalman Filter (SSKF)

The UKF is based on a choice of symmetric points on the axes of an n_x-dimensional Cartesian coordinate system. Differently, in the Spherical Simplex Kalman Filter formulation, the requirement for symmetry about the origin is replaced by a requirement that all vector points must be equidistant from the origin and from each other; thus, they lay at the vertices of an n_x-dimensional simplex.

It considers $n_x + 2$ integration points, which makes it the most efficient from the computational point of view. However, according to the theory, it should produce the least accurate approximation of non-linear functions. The reasoning is based on the low order of polynomials that are involved.

Sigma points and weights are computed as follows:

$$\mathbf{c}^{(j)}_{\mathbb{R}^{n_x}} = \begin{cases} [\mathbf{0}] \in \mathbb{R}^{n_x} & j = 0 \\ \begin{bmatrix} \mathbf{c}^{(j)}_{\mathbb{R}^{n_x - 1}} \\ -\frac{1}{\sqrt{n_x(n_x+1)w_1}} \end{bmatrix} & j = 1, 2, \ldots, 2n_x \\ \begin{bmatrix} \mathbf{0} \in \mathbb{R}^{n_x - 1} \\ \frac{n_x}{\sqrt{n_x(n_x+1)w_1}} \end{bmatrix} & j = n_x + 1 \end{cases} \tag{3.79}$$

$$\begin{cases} 0 \leq w_0 \leq 1 \\ w_j = \frac{1-w_0}{n_x+1} \quad j = 1, \ldots, n_x + 1 \end{cases} \tag{3.80}$$

3.5.2 Gauss-Hermite Kalman Filter (GHKF)

Gauss-Hermite integration is a quadrature method that uses orthogonal polynomials, instead of simple polynomials. This type of filter takes into consideration 3^{n_x} *sigma points*. Since the problem being studied has 12 elements in the state vector ($n_x = 12$), it will not be applied due to the excessive amount of calculations. In fact, Reference [15] advises that the GHKF should not be used for any problem in which the dimension exceeds four.

3.5.3 Monte Carlo Kalman Filter (MCKF)

The main idea behind the MCKF is approximating the Gaussian density by a set of Monte Carlo samples from that density:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \int \tilde{\mathbf{f}}(\mathbf{c})\, \mathcal{N}(\mathbf{c}; \mathbf{0}, \mathbf{I})\, d\mathbf{c} \\ &= \frac{1}{N_s} \sum_{j=1}^{N_s} \tilde{\mathbf{f}}\left(\mathbf{c}^{(j)}\right) \end{aligned} \tag{3.81}$$

The number of integration point N_s can be arbitrarily set. But, it can be shown that as $N_s \to \infty$, the approximation of the integral (given by the sample mean) becomes exact. The proof can be found in Reference [10]. Lastly, it is worth mentioning that factor $1/N_s$ is the importance weight of each sample, not a probability.

The MCKF usually give the most accurate results since no approximations are used for the non-linear functions, as long as the number of random samples is high enough. The UKF, instead, relies on third-order polynomials. However, Reference [15] suggests to try all filters on a given estimation problem and choose the one that yields the lowest RMS error, within the problems computational constraints.

Sigma points and weights are defined as:

$$\mathbf{c}^{(j)} \sim \mathcal{N}(\mathbf{c}; \mathbf{0}, \mathbf{I}),\ j = 1, \ldots, N_s \tag{3.82}$$

$$w_j = \frac{1}{N_s} \tag{3.83}$$

Chapter 4

Experimental section

Another relevant goal of the present thesis is to support the simulation process with data obtained from experimental tests. A market analysis involving MEMS sensors was conducted at first. The comparison among different products led to the choice of a candidate from STMicroelectronics®. Subsequently, static tests were performed on the device in order to assess parameters related to noise and biases. These data were taken as a reference during the simulation phase. Unfortunately, the possibility of carrying out other tests was severely limited by the COVID-19 pandemic that accompanied the whole duration of the project.

In conclusion, a CAD model of the physical support that could hold the circuit boards of the MEMS sensors was designed.

4.1 Market analysis

Targets of the market analysis were MEMS sensors that could provide data required by the AHRS. At the beginning, 3-axis accelerometers, gyroscopes and magnetometers were analysed, then the focus shifted to integrated solutions. They embed the three (3-axis) sensors into a single chip and for this reason are defined 9-axis devices. This architecture was chosen because it guarantees a less demanding installation, easier cable and connection management and it also provides a unified output of all necessary data.

Three 9-axis models from different companies were identified:

1. LSM9DS1, produced by STMicroelectronics® ([38])
2. ICM-20948, produced by TDK® - Invensense® ([20])
3. BNO055, produced by Bosch® ([3])

All three sensors are designed for consumer applications, including smartphones, tablets, wearable sensors, IoT applications, augmented reality, navigation, gaming and robotics. Thus, performances are in line with the ones required by devices of the aforementioned category. This is certainly a limiting factor, since high-performance sensors are desirable when performing attitude estimation and especially navigation.

However, the idea behind this project is to exploit low-cost sensors that are not specifically developed for aerospace applications.

Furthermore, they are designed with particular attention to power consumption. This feature is not particularly useful during initial tests, since the devices are connected to electricity. Nevertheless, it becomes a significant aspect when the AHRS is installed on its final application, for instance a UAV. In fact, since the aircraft is generally powered by batteries, power saving capabilities can positively affect the range of the drone.

Table 4.1: Comparison among the selected 9-axis MEMS devices.

	LSM9DS1	**CM-20948**	**BNO055**
Manufacturer	STMicroelectronics	TDK	Bosch
Year	2015	2017	2019
Accelerometer FS [g]	±2/±4/ ±8/±16	±2/±4/ ±8/±16	±2/±4/ ±8/±16
Gyroscope FS [dps]	±245/±500/ ±2000	±250/±500/ ±1000/±2000	±125 to ±2000
Magnetometer FS [G]	±4/±8/ ±12/±16	±49	±13 (x, y) ±2500 (z)
Analog Supply Voltage [V]	1,9 to 3,6	1,71 to 3,6	1,7 to 3,6
Data Output [bit]	16	16	16 (Gyro) 14 (Acc)
Serial Interfaces	SPI / I^2C	SPI / I^2C	SPI / I^2C
I2C Modes [kHz]	100 / 400	100 / 400 + SPI 7 MHz	*(Not specified)*
Operating Temperature [°C]	-40 to +85	-40 to +85	-40 to +85
Temperature Sensor	Yes	Yes	*(Not specified)*
Package	LGA-24L	24-Pin QFN	LGA 28 Pin
Dimensions [mm]	3,5 x 3 x 1	3 x 3 x 1	3,8 x 5,2 x 1,13

The main technical specifications of the sensors are reported in Table 4.1. They are taken from data sheets that are available on the product websites ([38], [20] and [3]). Values of Full Scale (FS) related to accelerometers and gyroscopes are similar in all devices; instead, the major difference lies in the magnetometers.

Specifics about sensitivities, biases, non-linearities, offsets, etc. are provided but in most cases they are not directly comparable among all three devices. Data concerning noise are not reported in data sheets (except for one value relative to the TDK® device); hence, they must be assessed through data acquisition tests.

4.2 Choice of the sensor

The three sensors present similar features and they can not be effectively compared in terms of performance on the basis of data provided by the manufacturers. Therefore, the selection of the sensor was made as a result of a product research in the market, focusing on ready-to-use solutions.

One of the candidates was Digilent® Pmod NAV ([9]), which embeds the LSM9DS1 (analysed in the previous Section), plus a digital barometer (LPS25HB) to provide users with 10-DOF functionality. But, since a barometer is not considered in the present configuration of *TetrAHRS*, the final choice fell on STMicroelectronics® STEVAL-MKI159V1 ([39]) that embeds the same LSM9DS1. It is an adapter board for standard DIL24 socket, designed to facilitate fast system prototyping and device evaluation.

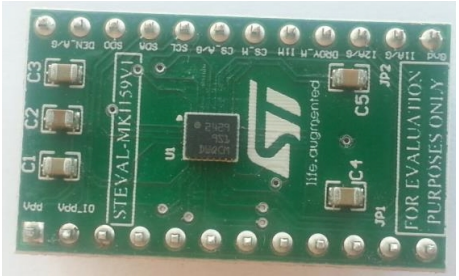


Figure 4.1: STMicroelectronics® STEVAL-MKI159V1, adapter board.

The MEMS sensor is integrated on the board with all the necessary electronic circuits. In fact, the adapter provides the complete LSM9DS1 pinout and comes ready to use with the required decoupling capacitors on the V_{DD} power supply line. This solution allowed to avoid the design and the production of a specific circuit board, necessary to place and operate the MEMS device. Data acquisition is performed through the SPI interface.

4.3 Configuration of the device

The STEVAL-MKI159V1 was connected to a STM32 Nucleo-144 board ([40]) in order to interface the sensor with the computer. The configuration of the MEMS device was performed by writing a C program, based on the code that is made available by the manufacturer on its website ([38]). Three main categories of parameters had to be set for each type of sensor:

1. Full Scale (FS)
2. Output Data Rate (ODR)
3. Anti-Aliasing (AA), High-Pass (HP) and Low-Pass (LP) filters (with their corresponding cut-off frequencies)

All settings were selected among the options that are reported in the product data sheet. The choices of FS and ODR are listed in Table 4.2 and Table 4.4. Two

configuration of Full Scale are proposed; this allows to analyze how noise and bias change according to the two settings. Starting from the accelerometers, a FS of ±2 g could not be enough during some maneuvers while ±16 g is far higher than typical values of acceleration; thus, settings in the middle of the range are chosen.

Regarding the gyroscopes, a FS of ±245 dps should be sufficient for most conditions; however, the value of ±500 dps is tested to assess how the noise changes with an wider range of measure.

Lastly, values of ±4 G and ±8 G are selected for the magnetometers. This choice was justified by the fact that the intensity of Earth's magnetic field ranges approximately between 0.25 G and 0.65 G [30].

Table 4.2: Two configurations of Full Scale (FS) values.

	Configuration 1	Configuration 2
Accelerometers FS [g]	±4	±8
Gyroscopes FS [dps]	±245	±500
Magnetometers FS [G]	±4	±8

All settings were compared with the FS that characterize MEMS sensors included in the two commercial AHRS, reported as examples in the thesis introduction. The values are presented in Table 4.3 .

Table 4.3: Values of Full Scale (FS) in the two commercial AHRS.

	AHRS-S305	POLAR-300
Accelerometer FS [g]	±10	±8
Gyroscope FS [dps]	±100	±300
Magnetometer FS [G]	±1	±10

As concerns the ODR, it shall be mentioned that accelerometers and gyroscopes work at higher frequencies while magnetometers reach lower rates. Therefore, a value of 119 Hz is chosen for the first two sensors and 80 Hz for the magnetometer. According to the sensor data sheet, 80 Hz is the highest available data rate without enabling *FAST ODR*. It has been hypothesized that such operating mode could worsen the performance of the magnetometer, due to the increased frequency of outputs. Specific tests should be performed in order to verify the hypothesis.

Table 4.4: Choices of the Output Data Rate (ODR).

	Output Data Rate
Accelerometers	119 Hz
Gyroscopes	119 Hz
Magnetometers	80 Hz

The 9-axis sensor offers a wide range of options in terms of filtering. To be more specific, the accelerometer is equipped with an AA filter, two LP filters and one

HP; the gyroscope is based on the same configuration, except for the anti-aliasing filter. Differently, the magnetometer does not have settable filtering parameters.

A thorough analysis of the sensor would be required to assess the influence of signal filtering on the outputs. But, since the investigation was not viable, it was decided to put two configurations to the test:

1. A first set of settings the minimizes the effect of filters, by excluding some and setting the remaining ones to *ULTRA LIGHT* mode.

2. A second configuration that was taken from the code provided by ST®, as example for data polling.

The details about all settings are reported on the data sheet ([38]).

4.4 Static data acquisition tests

Several tests of static data acquisition were performed with the equipment described in Section 4.2. The sensor was placed on a flat surface in the natural position, allowed by the wires that connected it to the board. Unfortunately, a precise horizontal alignment of the device was not possible due to the lack of necessary tools. Therefore, the orientation was random.

The acquisitions were not executed in a laboratory due to the unavailability of such space. This fact implies that many aspects, such as magnetic interference and other disturbing factors, were not taken into account. For this reason, the tests can not be considered as part of sensor characterization. However, they allowed to compute an estimate of the noise and offsets.

The first analysis focuses on the measurements acquired for approximately 2 hours with the following settings:

1. FS set to Configuration 1, described in Table 4.2

2. ODR as reported in Table 4.4

3. Filtering chain selected according to Configuration 1 (which minimizes the effects of filters)

Data were sorted into vectors associated with different axes of each type of sensor. Subsequently, the initial and final blocks of the data sets were removed, in order to minimize the effects of human interaction during the setting. Lastly, statistical properties were evaluated. They are reported in Table 4.5.

Starting with standard deviations, it can be observed that all values are compatible with the typical performance of a MEMS sensor. As expected, the gyroscope suffers from a considerable amount of noise, significantly higher than the other components inside the device.

By computing the norm of the means along the three axis of the accelerometer, a value of 958.35 mg is obtained. In theory, the sensor should measure 1 g, considering that the it is fixed and no other relevant accelerations act on it. The difference between the two values is due to the low accuracy of MEMS. The same point can be

Table 4.5: Statistical properties with minimum filtering (option no. 1).

Sensor	Axis	Mean (μ)	SD (σ)
Accelerometer [mg]	x	-163.19	1.1123
	y	944.35	1.3834
	z	2.96	2.5250
Gyroscope [mdps]	x	-1121.79	79.6380
	y	-2736.74	82.4513
	z	770.00	76.7795
Magnetometer [mG]	x	169.53	3.4759
	y	213.98	6.4358
	z	36.40	0.1542

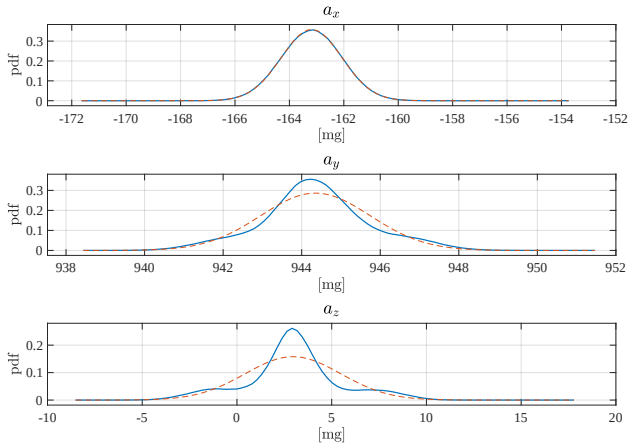
made by looking at the arithmetic means associated to the gyroscope axes. Instead of values close to zero (owing to the fact that the device is still), high angular rates are read. This gap requires proper compensation through filtering and eventually software, as it will be explained afterwards.

Finally, the estimated probability distribution functions (PDF) are computed through the dedicated MATLAB® command (which considers 100 points of the set) and analyzed. Figure 4.2 illustrates that the gyroscope is characterized by the closest PDFs to normal distributions, followed by the accelerometer. This result justifies the use of an additive Gaussian white noise as source of noise in the simulation. Differently, the magnetometer shows a more peculiar behavior on the y-axis, with a second small "bell curve", in addition to the main one.

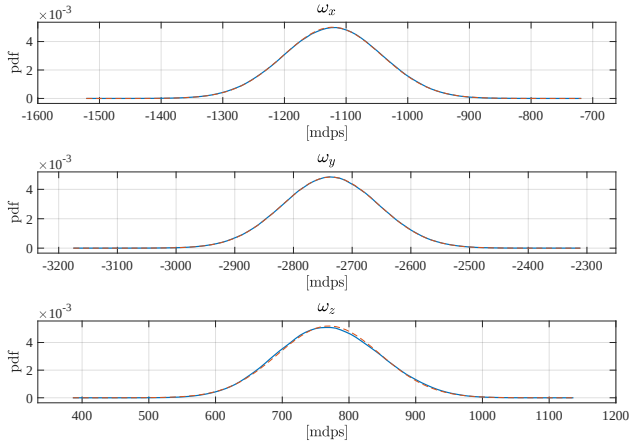
Two further data acquisitions with a similar duration (about 2 hours) were performed. In this case, the filtering settings were taken from the example uploaded by ST® (as described in the filtering option no. 2). Both configurations of FS were tested separately. Means and standard deviation are collected in Table 4.6.

Table 4.6: Statistical properties with filtering options taken from ST®; comparison between Configuration 1 and 2 of the FS.

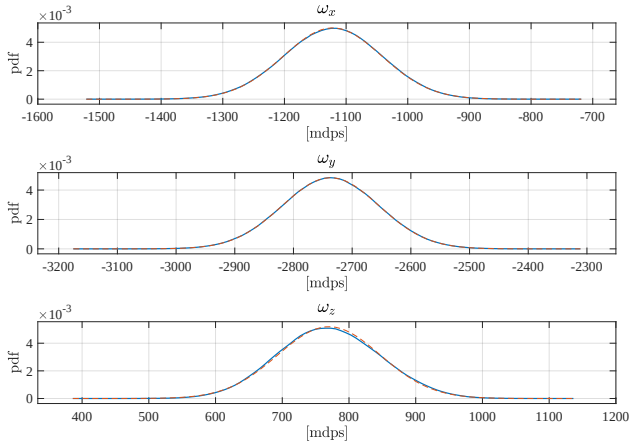
Sensor	Axis	Mean (μ)		SD (σ)	
FS Configuration		1	2	1	2
Accelerometer [mg]	x	-158.97	-162.46	2.5439	0.5100
	y	942.46	938.01	0.5579	0.3312
	z	3.26	8.2705	0.8883	0.7110
Gyroscope [mdps]	x	-4.37	-8.75	71.2126	67.4626
	y	-4.37	-8.76	75.3147	72.6352
	z	-4.37	-8.75	72.8967	75.7511
Magnetometer [mG]	x	168.83	169.53	3.9417	3.5200
	y	213.51	218.44	7.5628	17.3153
	z	36.40	2.6856	0.1102	12.3241



(a) Accelerometer



(b) Gyroscope



(c) Magnetometer

Figure 4.2: Comparison between probability distribution functions computed by MATLAB® (blue lines) and reference Gaussian distributions (dotted red lines) created on the basis of mean and standard deviation of data sets.

Comparing the statistics relative to the two filtering strategies (presented in Table 4.5 and Table 4.6), the aspect that stands out is the mean of the measurements obtained from the gyroscope. In fact, differently from the case with minimum filtering, the mean values along the three axes are close to zero. More specifically, they assume the exact same value along the three axes; a value that precisely doubles when the FS is doubled. The same means are obtained with other data acquisitions of unalike durations, performed on different days and with dissimilar orientations of the sensor.

This implies that the filtering chain was designed and set to those values by the manufacturer with the purpose of removing the large offset that was detected in static measurements (without filtering). The direct consequence is that the filters could influence the dynamic behavior of the sensor. Thus, during normal use in flight, measurements might be negatively affected in terms of accuracy. However, this hypothesis should be verified through a proper sensor characterization, focusing on both static and dynamic conditions.

The other statistical properties are similar in all three tests, with exception of the z-axis of the accelerometer and magnetometer. Standard deviations decrease as a result of the filtering and also by widening the FS. Once more, the magnetometer sets itself apart from the other two sensors, with a significant increase in σ associated to the broader FS.

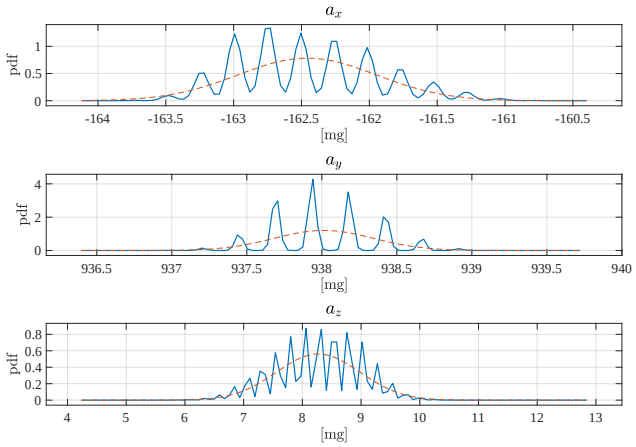
The estimated probability distributions functions (PDF) with the new settings are reported in Figure 4.3. The consequences of filtering are apparent in the curves relative to the accelerometer and the gyroscope; they are not smooth, as in the minimally filtered case. Nevertheless, the distributions of the gyroscope measurements have their peaks next to zero, as attested by the mean values (computed directly on the basis of the whole data sets).

In conclusion, the magnetometer is only affected by the change of the FS, since no filtering options are available (according to the data sheet). This is reflected on the curves, which are in fact very similar to the other tests, except for the z-axis. It is worth recalling the the magnetometer is subjected to EMI (which was not controlled during the acquisition), so that might affect the measurements.

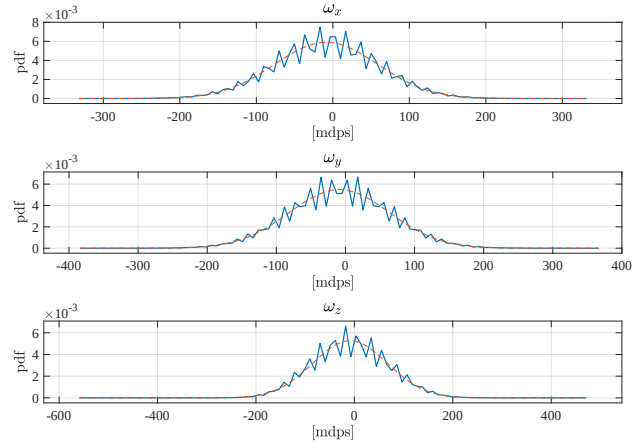
4.5 CAD model of the tetrahedral support

A prototype for testing purposes was designed. The starting point was a tetrahedron that could hold the four Digilent® Pmod NAV boards into position. It was soon noticed that only the central areas of the faces were useful for placing the sensors, so the vertex corners were cut out; this allowed to save space and material.

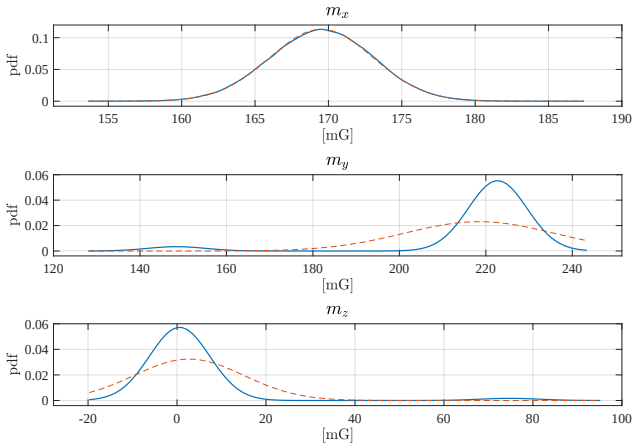
Subsequently, the focus moved to the way of fixing sensors to the structure. 3D-printed holders were designed to host the boards and secure them to the support through screws. The holders were thought specifically for the circuit board that was chosen at first. To be more specific, the negative side of Pmod NAV was recreated on the printed component; this way, sockets were obtained where resistors and capacitors protruded from the board. As a result, the latter could easily be glued to the holder.



(a) Accelerometer



(b) Gyroscope



(c) Magnetometer

Figure 4.3: Same comparison presented in Figure 4.2 but, in this case, the filters are chosen by ST® and the FS is set to Configuration 2.

A new thought concerned cable management; thus, dedicated cavities were added to the truncated tetrahedron. This choice allows to lay the prototype on a flat surface avoiding the interference of cables, which could cause misalignments.

The final improvement consisted in combining two mirrored truncated tetrahedron into a single structure. This way, up to eight sensors can be placed on a sole support. The CAD model of the prototype is presented in Figure 4.4. It clearly shows the holders, fixed on each face.

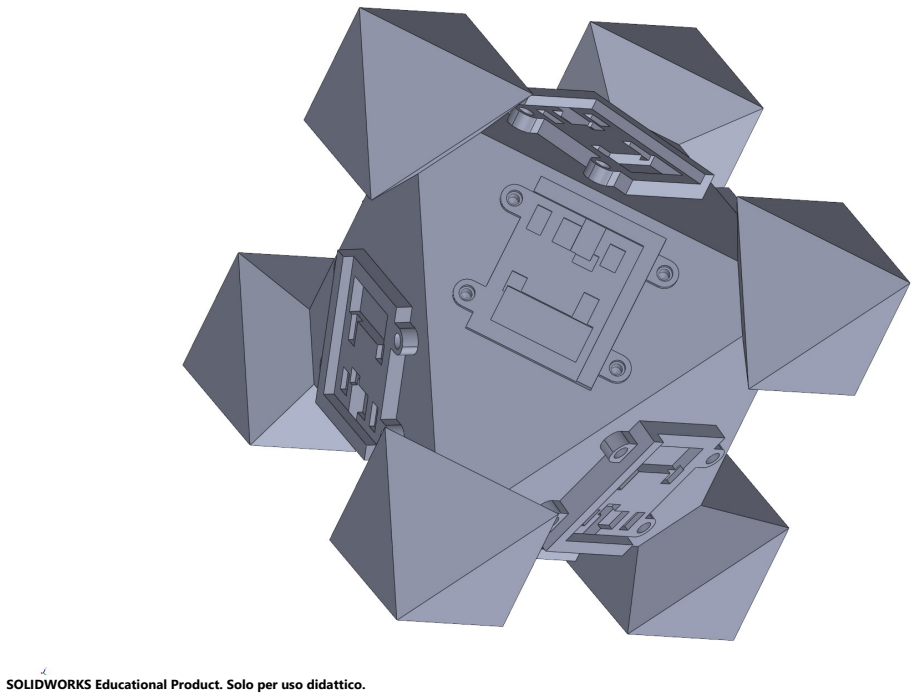


Figure 4.4: CAD model of the prototype for testing purposes.

As described in Section 4.2, the Pmod NAV was abandoned in favor of the adapter board from ST®. This means that the holders should be re-designed for the new circuit board. Nevertheless, the general concept and the geometry of the support remain valid.

Chapter 5

Results

The present Chapter collects and presents the results of the simulation process, which involves both the generation of measurements and Kalman filtering. At first, a simple trajectory between two waypoints is considered; the aim is to simulate a rapid maneuver. Thereafter, an arbitrary path becomes object of study.

The second part of the Chapter is dedicated to sensitivity analyses. In the first place, effects of the change in size of *TetrAHRS* are investigated. This study is performed in view of the miniaturization goal. Subsequently, several values of white noises are tested in order to assess their influence on the performance of the filter.

It must be taken into consideration that all test are performed on the basis of measurements obtained from an approximated model of MEMS sensors. In order draw more accurate conclusions on the filter, the model shall be improved, including more aspects and drawbacks that characterize real-world sensors. Further details can be found in Section 2.2.

5.1 Simulation of a manoeuvre

It was decided to start with one simple trajectory that was prescribed to *TetrAHRS* between the initial reference waypoint and an ending one:

Table 5.1: Position and attitude in correspondence of the two waypoints.

	N [m]	E [m]	D [m]	φ [deg]	ϑ [deg]	ψ [deg]
Initial waypoint	0	0	0	0	0	0
Ending waypoint	5	3	-2	70	60	210

Values associated with the ending waypoint are chosen arbitrarily. They could represent the attitude of a UAV at some point during a rapid maneuver. Larger variations are expected in the yaw angle (ψ) whereas roll (φ) and pitch (ϑ) angles are assumed to remain well bound within the range of $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. The assumption is made on the basis of some basic considerations about the flight dynamics of a generic UAV. The simulation settings of the test are reported in Table 5.2. They were all set through the MATLAB® App, described in Appendix A.

Table 5.2: Settings of the test.

Generation of measurements		
Edge length of the tetrahedron [m]		0.15
Distance of the sensor from the side [m]		0.005
Simulated data rate [Hz]		100
Simulation times	Total duration [s]	5
	Motion time [s]	3
Reference Earth's magnetic field vector	N [mG]	400
	E [mG]	200
	D [mG]	150
SD (σ) associated to Gaussian white noise	Accelerometers [mg]	1
	Gyroscopes [mdps]	100
	Magnetometers [mg]	5
Kalman filtering		
Sampling time of the filter [s]		0.01
Uncertainties related to the state	Linear acceleration [m/s^2]	0.01
	Angular acceleration [rad/s^2]	0.01
Uncertainties associated to the sensors	Accelerometers [mg]	1
	Gyroscopes [mdps]	100
	Magnetometers [mg]	5

The edge length of the tetrahedron is assumed to be 15 cm, the approximated size of a prototype that could be utilized for testing. It was selected an output data rate of 100 Hz since it is close to the mean of the ODRs utilized in the experimental tests; otherwise, it can be adjusted arbitrarily. The time of motion is set to 3 s, in order to simulate a generic maneuver.

As explained in Section 2.1.5, in the early stage of simulation it is not necessary that Earth's magnetic field vector (assumed as reference) is exact and that it corresponds to the test location. Therefore, arbitrary components along NED axes are chosen, with the only requirement that the magnitude of the magnetic vector ranges between 0.25 G and 0.65 G, approximately.

In accordance to what was reported in Section 2.2, a band-limited Gaussian white noise is added to the "ideal" measurements. The variances of the noise are obtained from the standard deviations (SD) relative to the three types of sensors. They were set as a result of the analysis performed on experimental data obtained from the sensor (outlined in Chapter 4).

It was decided to consider higher values compared to the experimental ones to further test the behavior of the filter. It consists of a conservative approach, seeing that results are expected to improve with a lower noise. The same values are used

for the uncertainties relative to the sensors - with reference to Eq. (3.30) - inserted in matrix **R**. Since all sensors are assumed to be of the same model, variances are repeated for the four of them.

One last observation has to be made: variances are assessed on the basis of static tests. The dynamic behavior of the sensors shall be analyzed too, possibly through experiments with dedicated equipment.

The uncertainties related to the state deserve a separate discussion, since there are no unique criteria to determine them. Thus, several tests with different values are performed to study their effects on the filter. However, more specific analyses should be carried out on this topic. Finally, going back to this simulation run, the selected uncertainties guarantee stable results with all the combinations of ending waypoints and time intervals of motion that were examined.

5.1.1 Outputs of the UKF

The simulation and Kalman filtering platform was designed to create plots of the estimated values of the state vector and measurements. These results are directly compared with simulated data obtained in the first phase of simulation. This process allows to check the validity and the accuracy of the estimates.

Starting with the state, the three components of the G-acceleration vector, defined in Eq. (2.33), are analyzed at first. As shown in Figure 5.1, the tracking of the acceleration is excellent, in spite of the rapidity of the maneuver (evidenced by the large values).

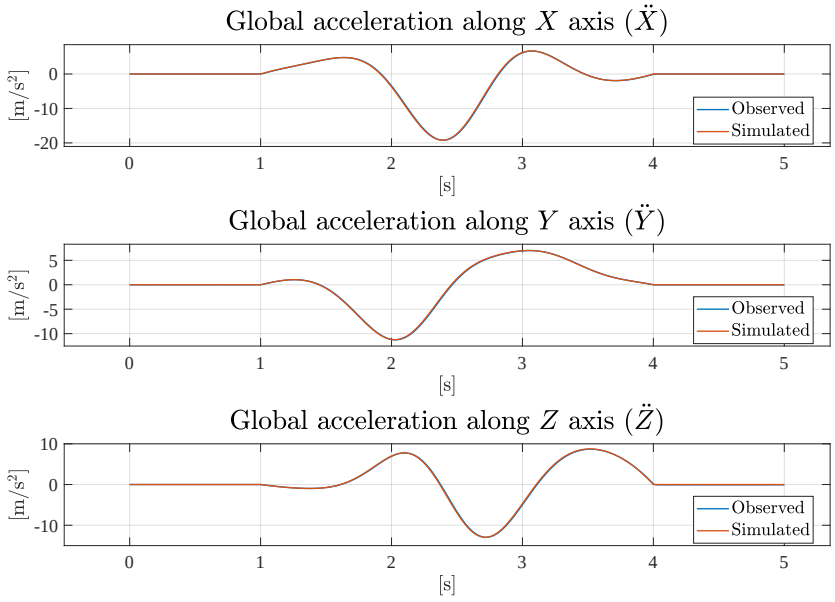


Figure 5.1: Single maneuver: Estimated acceleration of the CM.

The same considerations can be made on the angular velocity vector components along G axes. Figure 5.2 illustrates that estimated angular rates follow precisely the data obtained from the simulation process,introduced in Eq. (2.25).

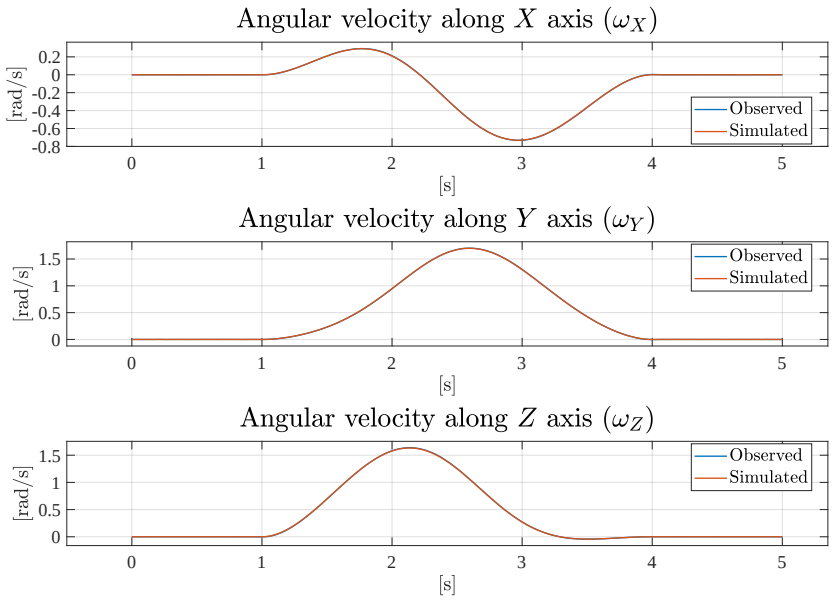


Figure 5.2: Single maneuver: Estimated angular velocity.

As regards to the angular acceleration vector - presented in Eq. (2.27) - Figure 5.3 reveals that the components do not coincide with the simulated ones. This phenomenon is perfectly explicable considering that angular acceleration is not directly measurable; in fact, *TetrAHRS* is not provided with sensors to measure that quantity. Hence, it is computed as derivative of the angular velocity.

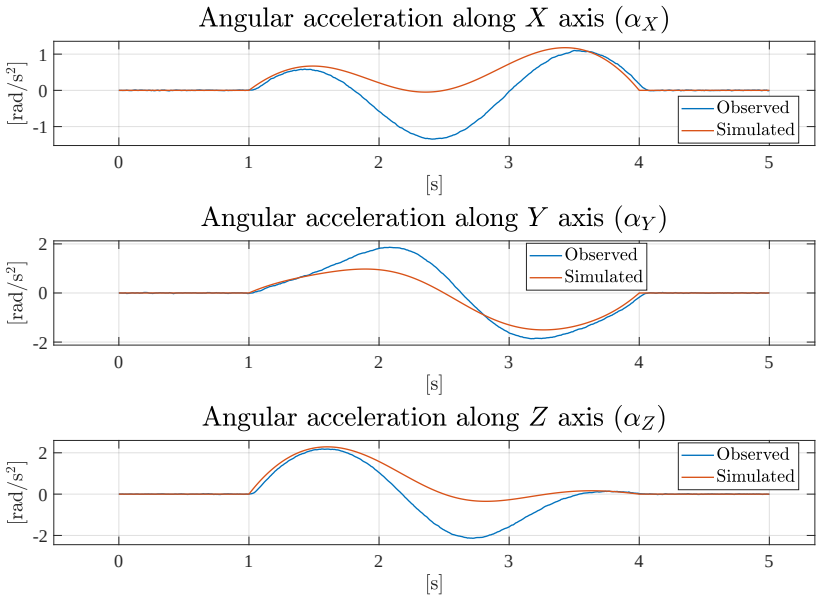


Figure 5.3: Single maneuver: Estimated angular acceleration.

For this reason, a correspondence between the two sets of data was not expected from the beginning. Nevertheless, it can be observed that the estimated values and

the simulated ones share a similar trend; this fact makes it a satisfying result.

The last three elements of the state vector are Euler angles. As shown in Figure 5.4, their estimate is accurate compared to the reference. Thus, the filter is able to track the maneuver precisely.

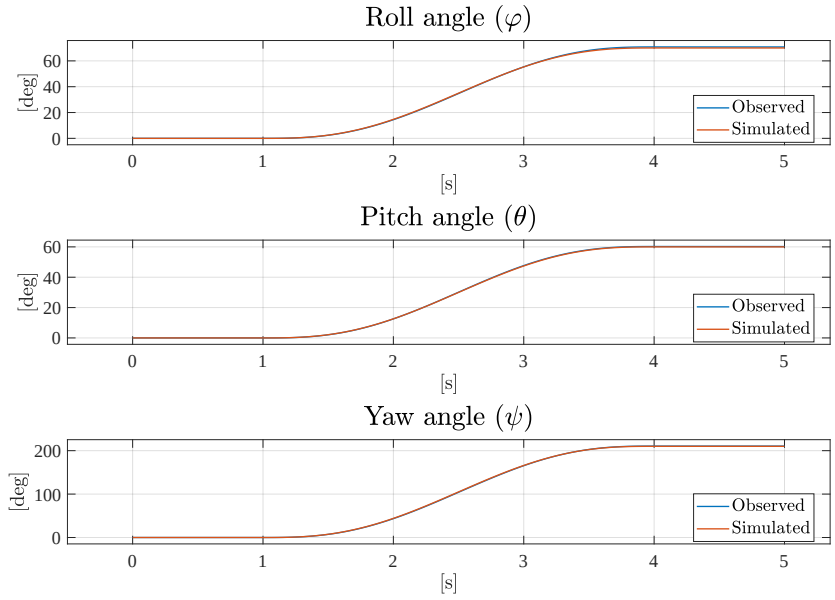


Figure 5.4: Single maneuver: Estimated Euler angles.

The UKF is designed to collect all the predicted elements of the measurement vector. This choice allows to verify the existence of a match between predicted measurements and simulated values. A lack of correlation between the two data sets would imply that the filter is not working properly. This could be due to problems in the non-linear sensor model that predicts measurements on the basis of the estimated state vector or in the dynamic model of the system which is responsible for the estimates.

The comparison between the two types of measurements can be conducted with all four sensors, exploiting the plots that are automatically generated by the simulation platform. For reasons of space, only the graph related to Sensor no. 1 is reported in Figure 5.5. It is noticeable that the two sets are coincident.

The white noise is not clearly visible in the figures (except for the magnetometer readings); this is due to the scale of the plots. In fact, the variations of measured quantities are significantly larger than the noise. Zooming in the graph is sufficient to show the details.

5.1.2 Effects of the discontinuity in ϑ

At the beginning of Section 5.1, it was stated that roll angle φ and pitch angle ϑ are assumed to be between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. This condition is valid for the vast majority of UAVs on the market. Exceeding that bound would in fact mean that the drone would tend to the upside-down flight, which is impossible (if the aircraft is not specifically designed for it).

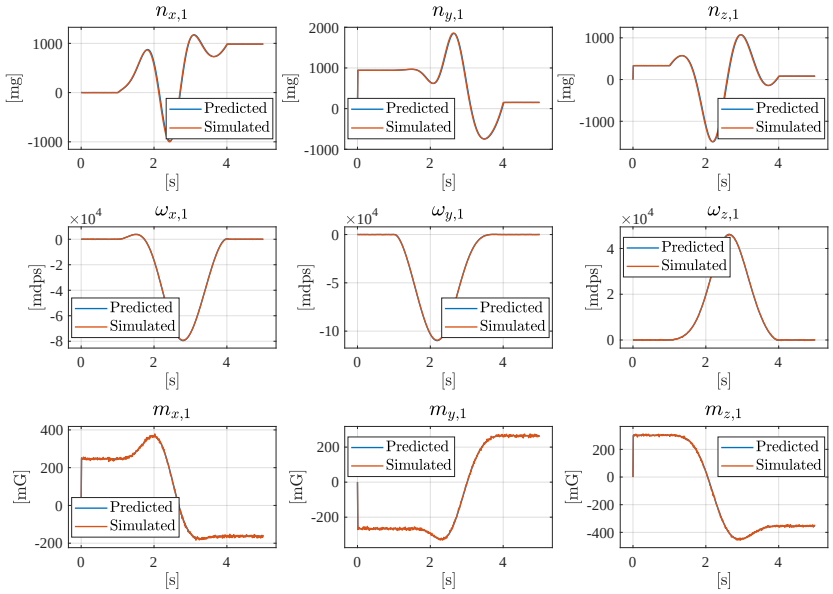


Figure 5.5: Single maneuver: Comparison between predicted (through the non-linear model) and simulated measurements.

However, some particular applications (for instance, acrobatic or military airplanes) could need to overcome that limit. If φ can be tracked without problems, pitch angle ϑ is subjected to the limitations given by the parametrization in terms of Euler angles. Thus, when $\vartheta \to -\frac{\pi}{2}$ or $\vartheta \to \frac{\pi}{2}$, the filter suffers from instability. The phenomenon is apparent in Figure 5.6.

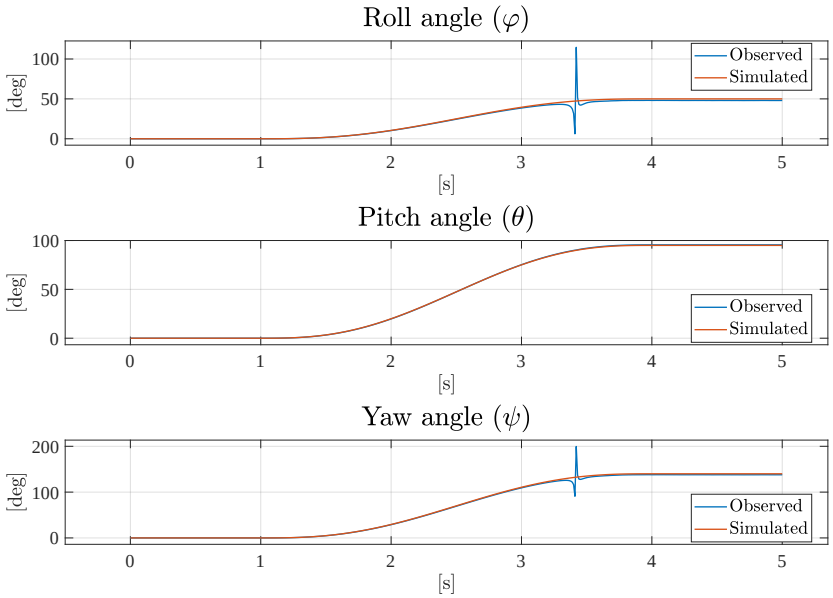


Figure 5.6: Discontinuity in ϑ: Attitude estimation utilizing all four sensors.

The simulation was performed with the same settings of the previous one (reported in Table 5.2). The only change was made to the attitude of *TetrAHRS* in the ending waypoint: $\varphi = 50\ deg$, $\vartheta = 95\ deg$ and $\psi = 140\ deg$. Once more, values were chosen arbitrarily, with the condition that $\vartheta > \frac{\pi}{2}$.

As expected, the discontinuity affects both φ and ψ and it happens in correspondence of $\vartheta = \frac{\pi}{2}$. From Figure 5.6, it can be observed that the filter is able to recover from the jump in the values. In fact, after less than half a second, the tracking becomes accurate again. This ability of the filter was encountered in the great majority of tests, with some exceptions associated to specific combinations of Euler angles. The reason should be further investigated.

The same test was carried out using only one MEMS sensor. As a matter of fact, the UKF was also designed to work with a single sensor, instead of the four of them together. The estimated attitude with Sensor no. 1 only is reported as example in Figure 5.7; the same occurrence happens with the other three sensors.

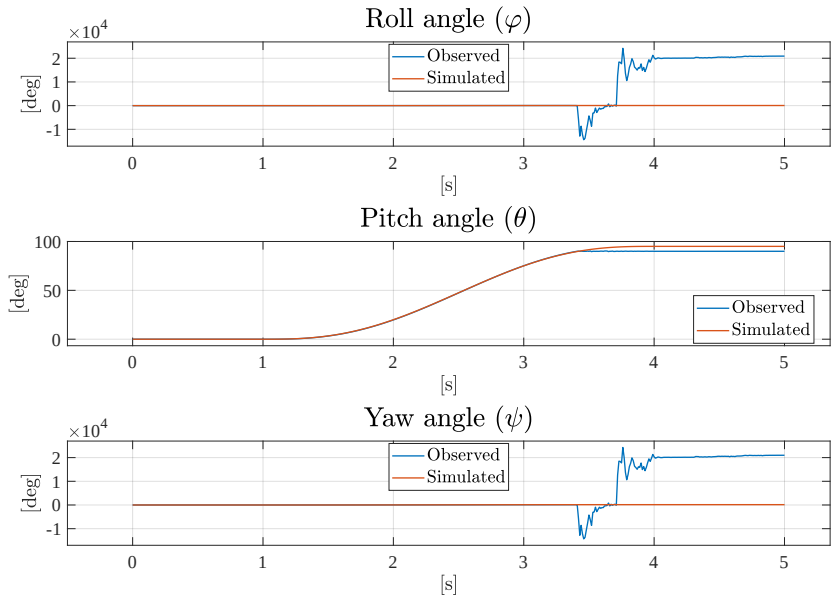


Figure 5.7: Discontinuity in ϑ: Attitude estimation using a single sensor (no. 1).

With the single-sensor configuration, the filter is not able to recover from the discontinuity. The result is a steep rise in φ and ψ values. Eventually, they stabilize into a condition that is not acceptable.

5.1.3 Addition of turn-on biases

Up to this point, a band-limited Gaussian white noise was added to the "ideal" measurements. Unfortunately, experience teaches that the sole addition of white noise is generally not sufficient to accurately model sensor outputs. Therefore, in order to improve the modeling of sensors, a turn-on to turn-on bias was included. On the contrary, other sources of error, especially those that vary over time or temperature, could not be implemented due to the lack of time and resources (reference to Section 2.2). Thus, all observations will be based on these assumptions.

Turn-on to turn-on biases are incorporated in most of sensor models proposed in the literature since they affect MEMS sensors. The aforementioned biases could not be assessed experimentally, so some plausible values are chosen for each component of the device. It was decided upon larger values than typical ones in order to take into account other possible contributions that are not directly correlated to turn-on. Reference biases (listed in Table 5.3) are multiplied for a random number drawn from the standard normal distribution within the range of $[-1, 1]$. The choice was made with the intent of trying to test the filter with a source of unpredictability. However, a proper sensor characterization, combined with a suitable sensor model, would be more revealing.

Table 5.3: Turn-on to turn-on biases reference values.

Sensor	Bias reference
Accelerometer [mg]	100
Gyroscope [mdps]	1000
Magnetometer [mG]	250

Five simulations with the same settings (indicated in Table 5.2) were run. Subsequently, root-mean-square errors (RMSE) were computed between the estimated attitudes and the simulated one (taken as reference). The errors are also compared with the results obtained from the model with white noise only. The comparison is effective since all RMSEs (collected in Table 5.4) are computed with respect a unique reference set of simulated Euler angles. In fact, according to Reference [18], scale-dependent measures (to which RSMEs belong) are useful when comparing different methods applied to the same set of data, but should not be used, for example, when comparing across data sets that have different scales.

Table 5.4: Single maneuver: Comparison of RMSEs corresponding to different simulation runs with the addition of turn-on to turn-on biases.

Simulation run	Roll angle (φ)	Pitch angle (ϑ)	Yaw angle (ψ)
No bias	0.0030	0.0080	0.0083
1	0.0264	0.0110	0.0092
2	0.0122	0.0058	0.0467
3	0.0078	0.0127	0.0348
4	0.0035	0.0065	0.0055
5	0.0199	0.0144	0.0206

As expected, in the vast majority of cases, the errors with respect to the simulated attitude increase due to the addition of biases. The variability is due to the source of randomness that was included. Nevertheless, all runs provide accurate estimates of the attitude. The Euler angles obtained by simulation no. 5 are shown as example in Figure 5.8. The visible larger deviations from the reference data (compared to the case without biases) are consistent with the higher RSMEs.

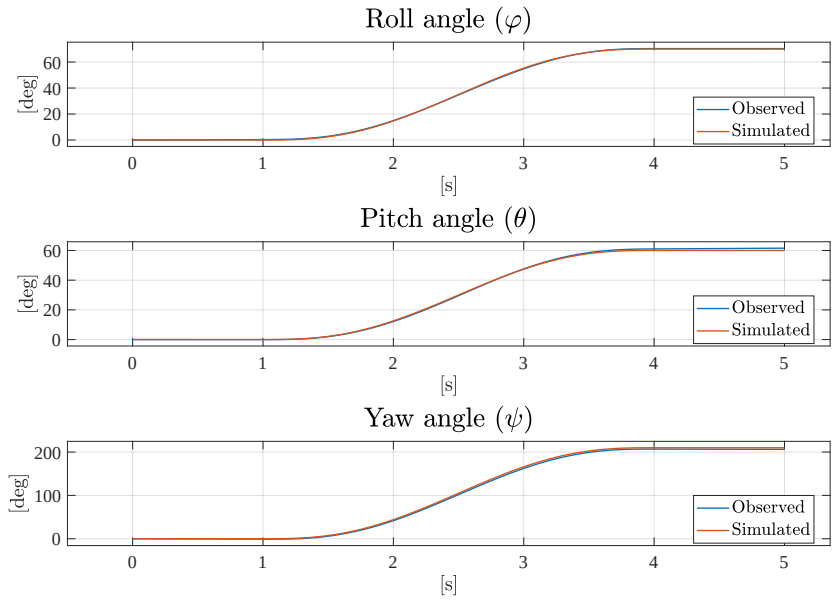


Figure 5.8: Single maneuver: Estimated Euler angles after the addition of turn-on to turn-on biases.

5.2 Motion along a path

The goal of this Section is to test the system on a more realistic trajectory, comprised of a sequence of waypoints, to whom positions and attitudes are associated. The settings are the same ones of the previous simulation (reported in Table 5.2) while waypoints are listed in Table 5.5.

Table 5.5: Position, attitude and time of motion for each waypoint of the sequence.

Waypoint	N [m]	E [m]	D [m]	φ [deg]	ϑ [deg]	ψ [deg]	Time [s]
1	3	3	-2	30	35	80	5
2	2	4	-1	40	55	210	3
3	5	3	-4	70	20	330	5
4	2	1	-2	25	60	20	10
5	0	0	0	0	0	0	2
6	-2	-3	4	-40	-20	-270	10
7	-4	-2	3	20	-40	-80	5
8	-1	-1	1	-30	15	60	10
9	2	1	-2	50	-25	170	5
10	0	0	0	0	0	0	5

TetrAHRS moves from a waypoint to the next one, taking the time that is prescribed to perform the motion. All values of position, attitude and time intervals are chosen arbitrarily, following the assumptions that were made in Section 5.1. In fact, larger variations are set to the yaw angle ψ. Differently, roll (φ) and pitch (ϑ) angles are selected in the range between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. To be more specific, values

of aforementioned angles generally remain well below the bounds. Once again, imagining typical maneuvers performed by UAVs is sufficient to justify the last observation.

The first analysis is performed on the results obtained from the model based on the sole addition of Gaussian white noise, whereas turn-on to turn-on biases will be included afterwards. Similarly to the simulation of a single maneuver, the UKF provides estimates of the state vector for the whole path.

Since the aim of an AHRS is to provide attitude and heading, the estimates of Euler angles are examined at first. The tracking with respect to the simulated reference data is excellent. This means that the goal of combining measurements (simulated as Gaussian white noise) obtained from several MEMS sensors is accomplished. The filter is in fact perfectly able to estimate the attitude of the system.

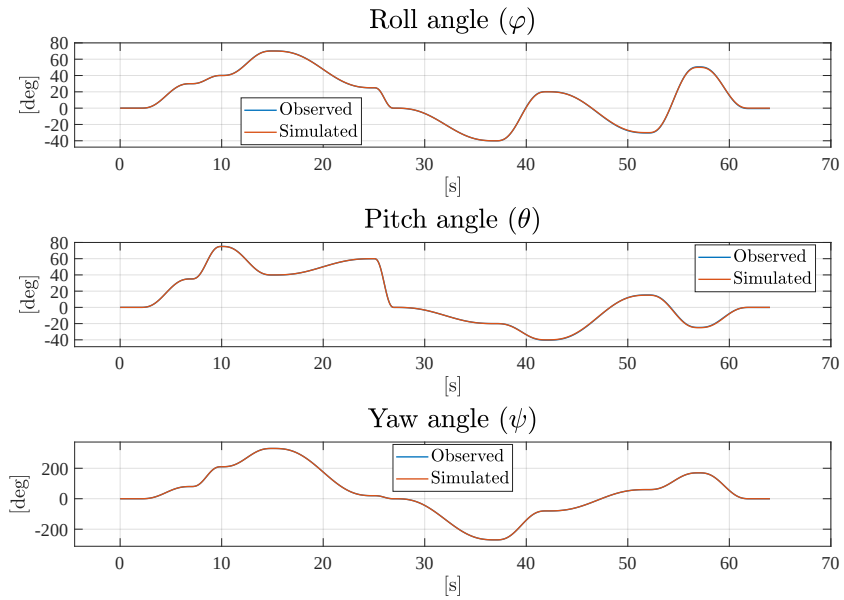
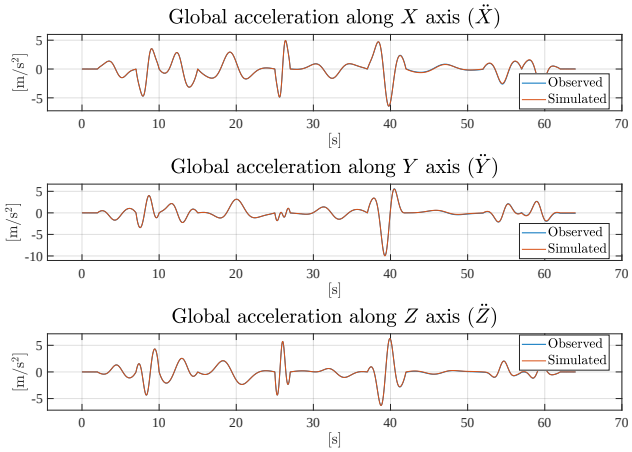


Figure 5.9: Motion along a path: Estimated Euler angles with Gaussian white noise.

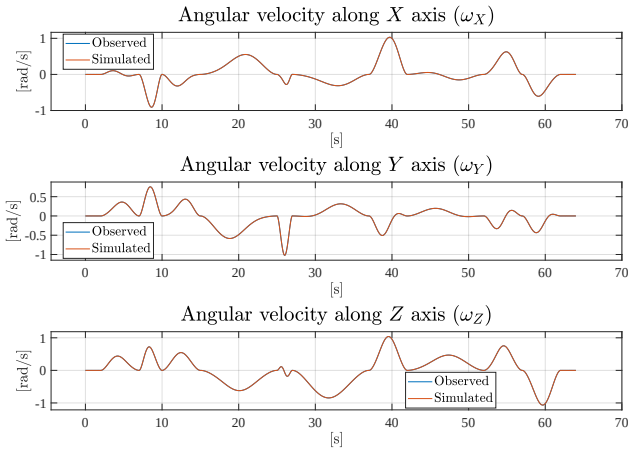
Moving to the other elements of the state vector, it can be noticed the G-acceleration is estimated precisely. An identical remark can be made on the angular velocity vector components. Instead, for the same reason that was explained in Section 5.1.1, the angular acceleration is not expected to be guessed accurately. On the contrary, the filter produces a good estimate of the aforementioned simulated quantity. Results are illustrated in Figure 5.10.

Predictions of sensor measurements are stored and, therefore, available at each instant of time. By comparing the predicted data with simulated outputs (used as input for the filter), excellent matches are gotten with all four sensors. The relating plots were not included in the dissertation for reasons of space.

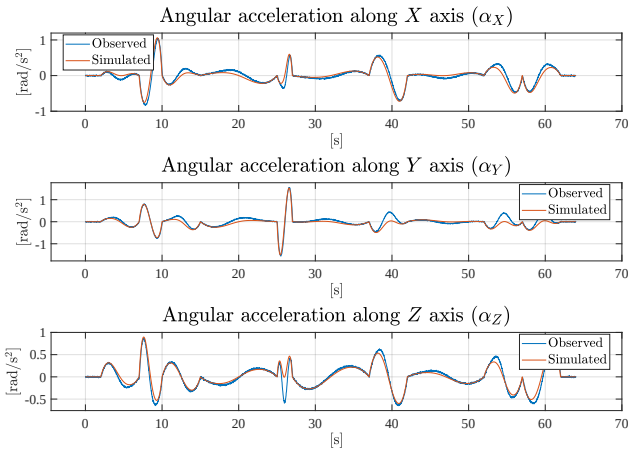
In conclusion, the analysis of the filter behavior with waypoints characterized by $\vartheta > \frac{\pi}{2}$ along the path was performed. Differently from the single maneuver, in this instance, the filter was not able to recover from the discontinuity. Thus, if the aircraft on which the AHRS is installed requires exceeding the limits on the



(a) G-acceleration



(b) Angular velocity



(c) Angular acceleration

Figure 5.10: Motion along a path: Estimated state vector components with Gaussian white noise.

pitch angle ϑ, the implementation of the parametrization in terms of quaternions is strongly recommended.

5.2.1 Comparison between TetrAHRS and a single-sensor configuration

The UKF was also designed to work with a single sensor instead of the four of them together. As illustrated in Appendix A, the desired option can be set through a dedicated switch. This way, the state vector remains the same while the measurement vector includes only the outputs of the selected sensor. Subsequently, the non-linear sensor model that computes the predicted measurements based on the state is modified accordingly. As a result, the Kalman filter works as if just one sensor is connected and feeds it with data.

The comparison between the single-sensor configuration and the one that utilizes all of them is based on a simulation that follows the same path described in Table 5.5, with identical settings. At first, it is performed with white noise only, then, turn-on to turn-on biases (introduced in Section 5.1.3) are added. The goal is to test the filter in two different operating conditions; in particular, the second one sees an additional source of unpredictability to the Gaussian white noise.

Beginning with the first analysis, it was decided to run 10 simulations with the same settings (including the variance associated to the white noise). Analogously to Section 5.1.3, RMSEs are computed between the simulated attitude (which serves as reference) and the estimated ones, differentiating *TetrAHRS* from single-sensor solutions. Aforementioned errors are plotted in Figure 5.11.

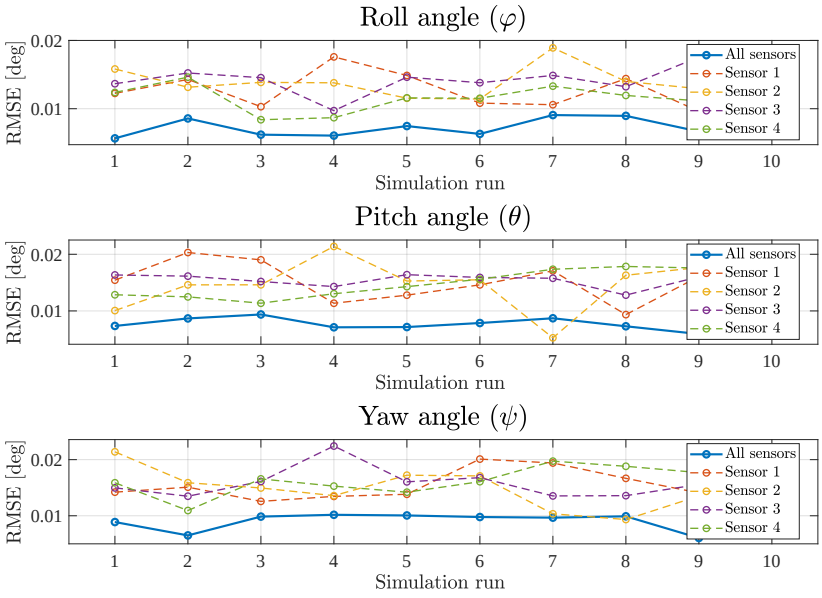


Figure 5.11: White noise only: RMSEs on Euler angles obtained by 10 simulation runs.

The use of RMSEs is justified by the fact that they all refer to the same quantities, obtained from different models. Moreover, the errors are all computed with respect to the same reference data set.

Figure 5.11 illustrates that *TetrAHRS* provides a more accurate overall estimate of the attitude compared to single sensors. This is attested by the lower RMSEs that are obtained in all 10 runs.

The same comparison is performed with the addition of the random turn-on bias (described in Section 5.1.3) to the Gaussian white noise, in order to study the new behaviour of the filter. The estimated Euler angles, computed with the different configurations, are shown in Figure 5.12. The simulated data, taken as reference, were not included in the graph with the intent of avoiding confusion with colors. Instead of performing a visual comparison, it was preferred to measure the accuracy by exploiting the RMSEs, whose values are collected in Table 5.6.

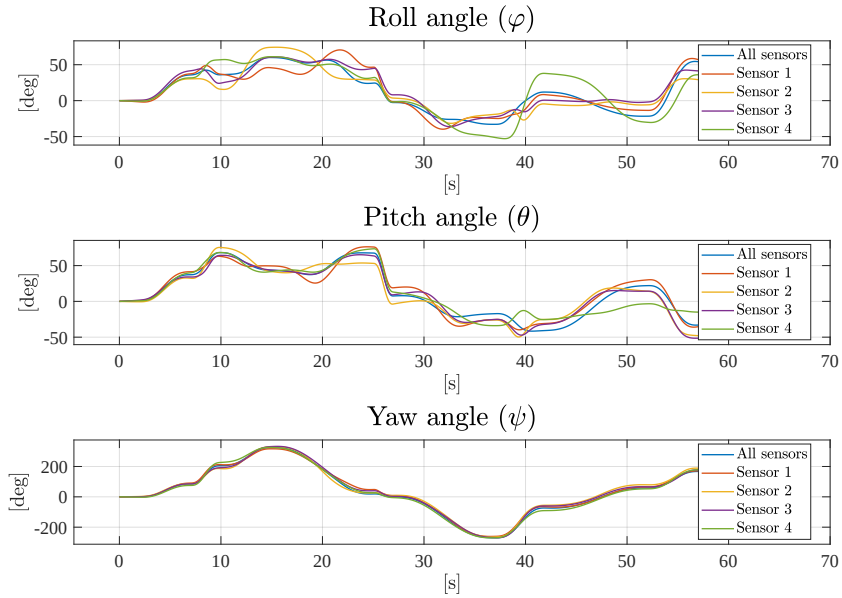


Figure 5.12: Turn-on bias added to white noise: Comparison between *TetrAHRS* and single-sensor configurations.

Table 5.6: Turn-on bias added to white noise: Comparison of RMSEs between *TetrAHRS* (with the four sensors combined together) and single-sensor configurations.

	Roll angle (φ)	Pitch angle (ϑ)	Yaw angle (ψ)
All four sensors	6.8664	5.6714	4.0792
Sensor 1	15,2736	12.0007	12.4141
Sensor 2	15.2321	11.7941	13.9427
Sensor 3	14.9114	12.3139	10.9723
Sensor 4	11.2623	10.0043	8.4368

By analysing the RMSEs, it can be observed that also after the addition of the bias, *TetrAHRS* performs better than single-sensor solutions. In fact, the errors relative to the three angles are approximately half of the lowest ones among the other four configurations.

In order to verify the better accuracy provided by the combination of four sensors, the same simulation is run 10 times. RMSEs are computed for each test and results are revealed in Figure 5.13.

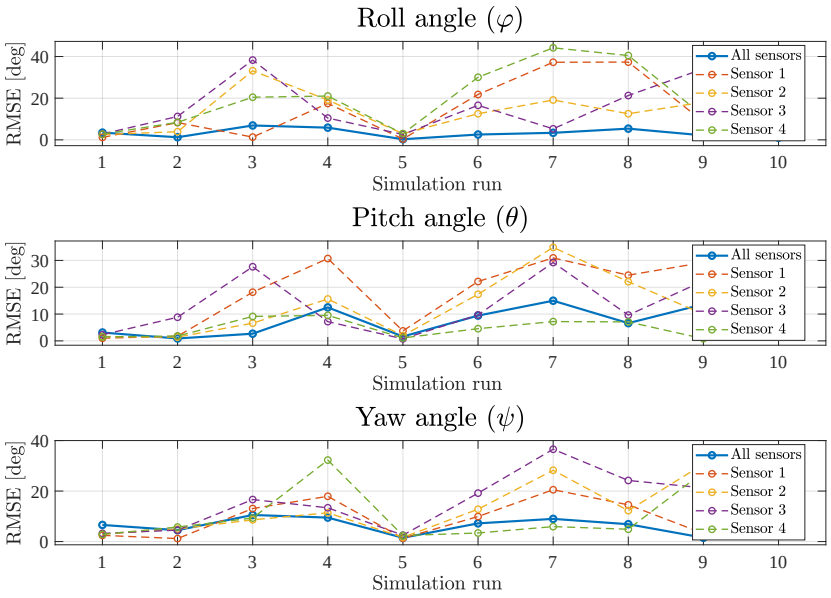


Figure 5.13: Turn-on bias added to white noise: RMSEs on Euler angles obtained from 10 simulation runs.

Again, examining the errors got by the 10 runs, TetrAHRS proves an overall increased accuracy compared to single sensors. The merit is attested by the indexes reported in Table 5.7, calculated by averaging the sum of RMSEs (referred to the three angles) over the number of tests. They are not proper figures of merit but they give a quantitative idea of the errors, including all three Euler angles together and taking into account a larger amount of simulations.

It must be specified that the aforementioned statement is formulated on the basis of the hypothesis that were introduced in the dissertation. In order to assess its truthfulness, further testing must be performed with more accurate models and proper sensor characterization. Moreover, building the prototype that was proposed in Section 4.5 could help the study.

Table 5.7: Turn-on bias added to white noise: Mean over 10 runs of the sum of RMSEs related to Euler angles.

	Mean of RMSEs
All four sensors	16.6605
Sensor 1	41.7922
Sensor 2	37.8308
Sensor 3	43.7691
Sensor 4	34.7378

5.3 Sensitivity analyses

The performance of *TetrAHRS* is influenced by several factors. Hence, sensitivity analyses are performed in order to assess the dependence on two variables: different sizes of the tetrahedron and variances of white noise.

5.3.1 Sensitivity to the size of the tetrahedron

Three simulations are run, with identical settings (with reference to Table 5.2) except for the edge length of the tetrahedron. The path followed by *TetrAHRS* differs from the previous tests; this choice is made with the intent of increasing the level of generality. The three lengths of the tetrahedron side are:

- 20 cm (compatible with a large scale prototype for preliminary testing)
- 2 cm
- 2 mm (extremely miniaturized version of the system)

The randomness introduced by turn-on to turn-on biases is removed, in order to have only one controlled (varying) parameter: the size. Since the estimated Euler angles are extremely close to the simulated ones, it was preferred to report the absolute value of the deviation from the reference at each instant of time.

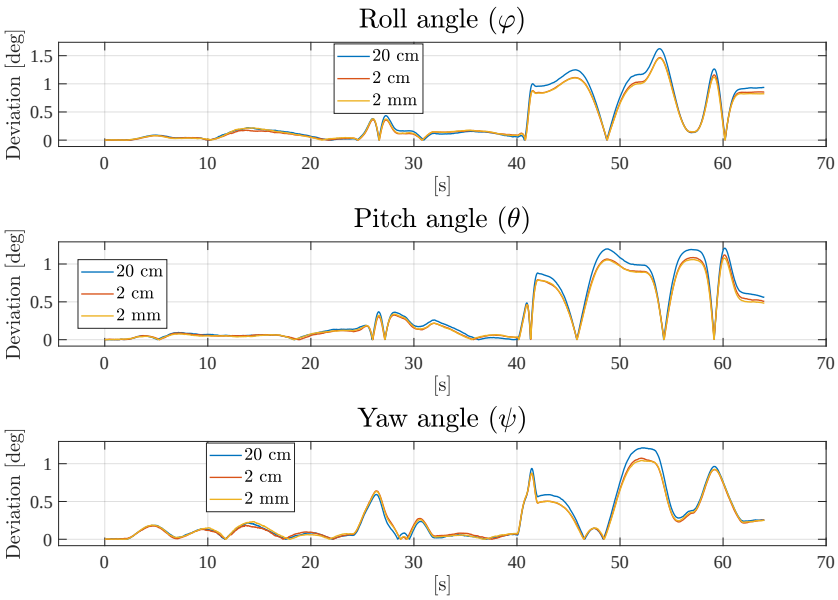


Figure 5.14: Absolute value of the error between estimated Euler angles and simulated ones by varying the edge length of *TetrAHRS*.

Figure 5.14 demonstrates that a size reduction of *TetrAHRS* leads to a slight increase in accuracy. This fact can be explained by the decreased "lever arm" (which goes together with size) that multiplies the angular acceleration. More details can be found in Section 3.2.1.

Once more, RMSEs are utilized to quantify the accuracy of *TetrAHRS* according to its size. Values are reported in Table 5.8. Despite the small reduction in the RMSEs, this outcome pushes in favor of miniaturization; a process that involves new challenges.

Table 5.8: Sensitivity to size: values of RMSEs.

Size	Roll angle (φ)	Pitch angle (ϑ)	Yaw angle (ψ)
20 cm	0.5812	0.5250	0.4142
2 cm	0.5232	0.4726	0.3774
2 mm	0.5166	0.4646	0.3765

5.3.2 Sensitivity to white noise

Similarly to the previous Section, other three simulations are preformed (on a dissimilar path). The difference is that, in this case, the variable parameter is the standard deviation (σ) associated with the Gaussian white noise. Values of σ are chosen in relation to the reference variance that was indicated in Table 5.2. The first case considers a variance that is 1/10 of the reference, while the other two are characterized by values that are 5 and 10 times the original SD.

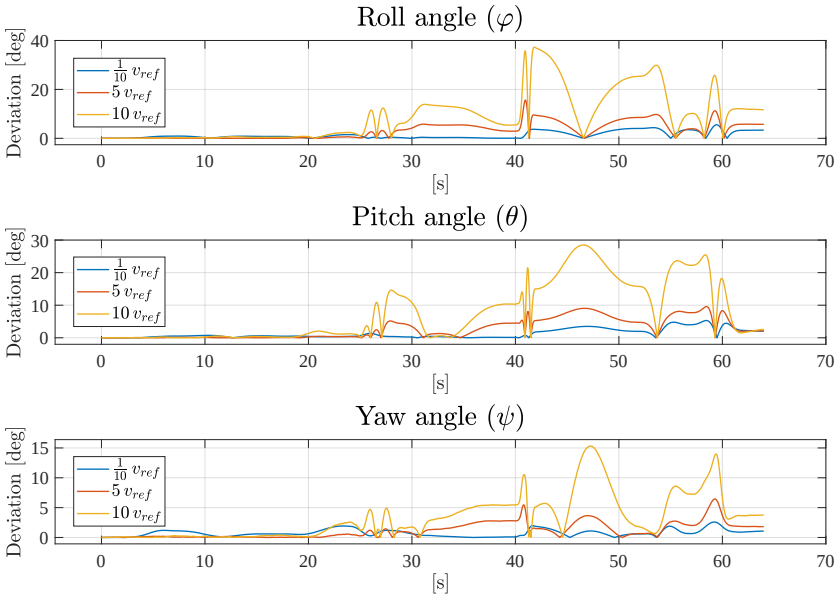


Figure 5.15: Absolute value of the error between estimated Euler angles and simulated ones by varying the standard deviation (σ) related to white noise.

As could be expected, an increase in the variance that defines the Gaussian white noise causes a decrease in accuracy. It is in fact more demanding to filter a noise that is described by an higher SD. This observation is confirmed by the RMSEs collected in Table 5.9. With the aim at generality, a different path (from the one that was followed in Section 5.3.1) is considered.

Table 5.9: Sensitivity to white noise: values of RMSEs.

SD (σ)	Roll angle (φ)	Pitch angle (ϑ)	Yaw angle (ψ)
$\frac{1}{10}\, v_{ref}$	1.9137	1.8531	0.9389
$5\, v_{ref}$	4.4906	4.0771	1.7616
$10\, v_{ref}$	13.1075	11.8166	4.9754

The result of this analysis leads to the reasonable conclusion that *TetrAHRS* works better with less noisy sensors.

5.4 Comparison among Sigma Points Kalman filters

With reference to Section 3.5, three alternatives to the UKF are presented. The Spherical Simplex Kalman Filter (SSKF) and the Monte Carlo Kalman Filter (MCKF) will be analysed, whereas the Gauss-Hermite Kalman Filter (GHKF) will not be tested due to the excessive amount of calculations required. It is in fact not suitable for problems with more than four elements in the state [15].

The two aforementioned Sigma Points Kalman filters are compared with the UKF on the basis of a single simulation. Since the number of Sigma Points for the MCKF can be arbitrary chosen, two values are selected: 25 (for a direct comparison with the UKF) and 100 points. RMSEs are computed among the estimated Euler angles obtained from the filters and the simulated (reference) attitude.

Table 5.10: Alternatives to the UKF: values of RMSEs.

Filter	Roll angle (φ)	Pitch angle (ϑ)	Yaw angle (ψ)
UKF	1.4313	1.2541	0.8047
SSKF	169.3694	32.8320	599.2391
MCKF (25 points)	15.2874	10.9494	22.2557
MCKF (100 points)	5.9789	4.2218	3.6152

By examining the errors reported in Table 5.10, it can be pointed out that the UKF delivers a significantly better performance compared to the other members of the Sigma Point family. The SSKF is completely incapable of tracking the attitude; as a matter of facts, its estimates diverge and eventually stabilize to unacceptable values. Therefore, its implementation is not suggested. The behavior is apparent in Figure 5.16.

In conclusion, the MCKF deserves a separate discussion. The idea behind it is to consider a set of integration points taken from a normal probability distribution; this implies a certain variability in the estimates. The statement is supported by the outcome of 10 simulations, run on the basis of a single set measurements given as input to the non-linear filter. RMSEs with respect to the reference attitude are plotted in Figure 5.17.

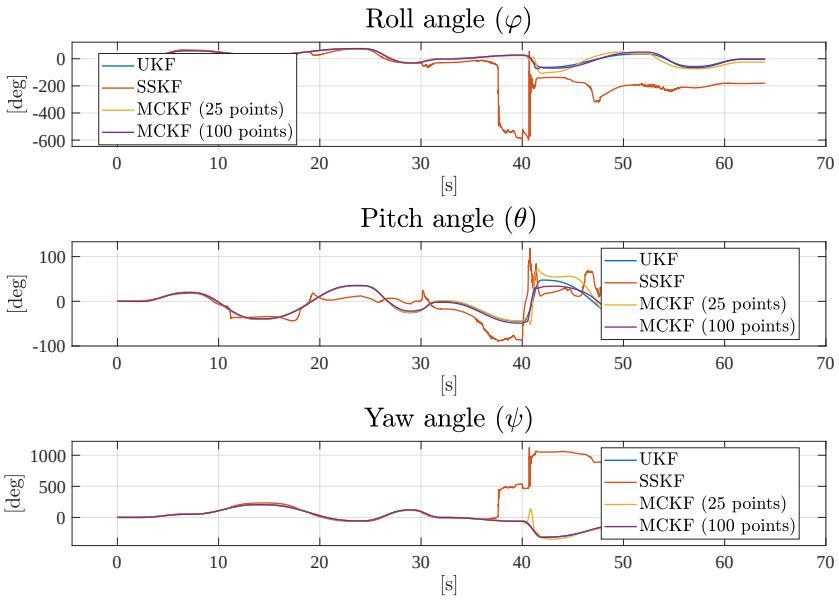


Figure 5.16: Alternatives to the UKF: Estimated Euler angles.

The UKF exploits 25 integration points, so it was chosen the same value for the MCKF, in order to have a direct match. Figure 5.17 clearly shows the fluctuating trend of errors related to the MCKF. On the contrary, the UKF provides the same estimate at each simulation run; the latter is more accurate than the vast majority of estimates obtained from the opponent. Therefore, out of the three choices, the Unscented Kalman filter (UKF) delivers the best performance and, moreover, it provides stable results; for these reasons, it was chosen for *TetrAHRS*.

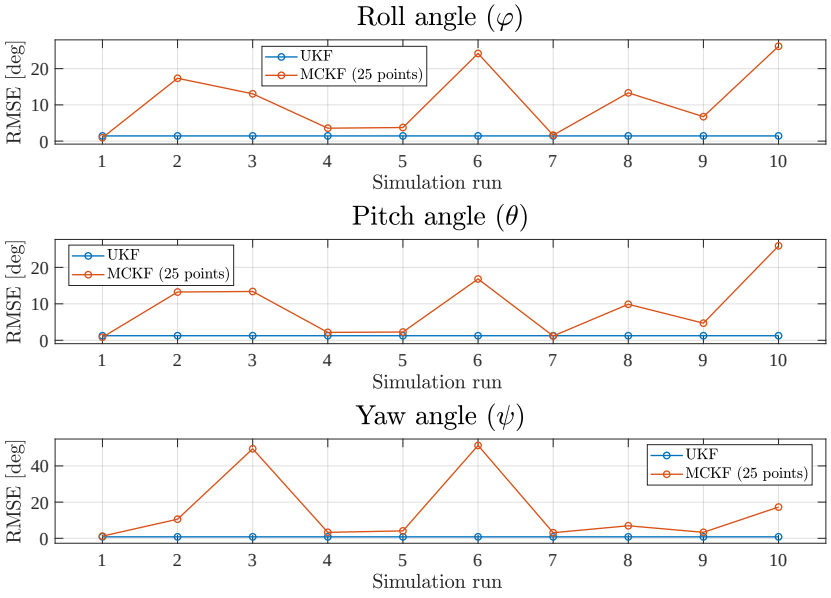


Figure 5.17: MCKF with 25 Sigma Points: RMSEs obtained from 10 simulation runs.

Conclusions

The present thesis outlines a viable approach to the design of a redundant AHRS, based on four 9-axis MEMS sensors placed on the faces of a tetrahedron. The choice of such configuration has two significant advantages: working with multiple skewed sensors (which implies a redundancy of measurements) and the potential capability to detect faults and isolate them. This product could meet the increasing demand in the market of UAVs, in particular consumer ones.

The study begins with the development of a simulation platform that allows to generate sensor measurements according to the motion of *TetrAHRS* along a desired trajectory. Outputs are obtained from a sensor model that simulates a Gaussian white noise (whose associated variances are obtained from experimental data) and an additional random turn-on bias.

An Unscented Kalman Filter (UKF) is implemented in order to combine sensor measurements and estimate the state of the system. Therefore, the definition of the framework at the core of the Kalman filter is provided. To be more specific, an UKF is chosen with the purpose of dealing with the non-linearities included in the problem. Alternative Sigma Point Kalman filter formulations are proposed and compared with the Unscented one.

In the end, an experimental section is dedicated to static data acquisitions, performed with a 9-axis MEMS sensor manufactured by STMicroelectronics®. Outputs are examined and used to assess the variance of noise that characterizes static measurements.

Two major achievements can be highlighted. The first is having successfully designed a simulation environment that generates consistent sensor measurements. The second one concerns the combination of the outputs, which is accomplished through a Kalman filter that delivers accurate estimates of the attitude.

The behavior and performance of the UKF are analyzed through several simulations that reproduce different conditions and characteristics of the system. To this end, having available simulated attitude data (from which sensor measurements are computed) is a remarkable benefit because they can be taken as a reference for the comparison with the corresponding estimated quantities.

A single rapid maneuver is tested at first, followed by the simulation of a more complex path comprised of arbitrary waypoints. The filter provides excellent performance, with the condition that the limitations given by Euler angles are met. The positive aspect is that this requirement is compatible with the vast majority of typical UAVs.

A second step consists in adding a random turn-on to turn-on bias to the white

noise. This input introduces a source of unpredictability with the intent of assessing the performance of the UKF in a more challenging condition. As expected, the accuracy reduces but not enough to compromise the validity of the filter.

In both cases, before and after the addition of the bias, *TetrAHRS* shows a better accuracy compared to the same non-linear filter fed by a single sensor. The present outcome pushes in favor of the exploitation of multiple skewed sensor but it is must be emphasized that it is obtained on the basis of a limited sensor model. Further testing, together with proper sensor characterization, should be more revealing.

Sensitivity analyses allow to conclude that size reduction of the system has positive effect on accuracy; therefore, miniaturization shall be pursued. Moreover, a last (easy to guess) beneficial impact is given by the adoption of less noisy sensors. However, this aspect could collide with the low-cost target that is aimed at.

In conclusion, the reason behind the choice of the UKF is justified by the comparison with other two Sigma Points Kalman filters. The SSKF diverges to unacceptable values while the MCKF provides estimates with fluctuating accuracy; thus, their implementation is not suggested.

The design of an AHRS is a very complex project and it can not be exhausted with a Master's thesis. Numerous correlated activities and further developments can be proposed:

1. Conducting a thorough sensor characterization, in parallel with the development of an accurate sensor model. This process allows to obtain a simulation environment that is much closer to reality.

2. Implementing a parametrization by means of quaternions instead of Euler angles. They are more efficient and eliminate problems related to Gimbal Lock. However, they are less intuitive and not prone to direct visualization.

3. Building a "large scale" prototype for testing. About that, the CAD model proposed in Section 4.5 can be a source of inspiration. Alternatively, following straight the path to miniaturization, beginning with a dedicated design.

4. Implementing a method to accomplish *fault detection* and *isolation*. This useful capability is enabled by the fact of having redundant sensors. A possible strategy is presented in the Section dedicated to the thesis objectives. In addition, the Receiver Autonomous Integrity Monitoring (RAIM), a technology developed to assess the integrity of the Global Positioning System (GPS), could provide other ideas.

Appendix A

Simulation and Kalman filtering platform

The mathematical procedures presented in Section 2.1 and Section 2.2 were converted into MATLAB® codes. Subsequently, the scripts were organized into an all-in-one simulation and Kalman filtering platform with an user-friendly interface. This choice made the whole process faster and it also facilitated the control over the important settings.

Motion between two waypoints

The first version of the platform allows to simulate of motion of *TetrAHRS* between the starting point (described by $N = 0\ m$, $E = 0\ m$, $D = 0\ m$, $\varphi = 0\ deg$, $\vartheta = 0\ deg$, $\psi = 0\ deg$) and a desired ending waypoint. The values associated with the latter can be entered in the dedicated fields contained in the MATLAB® App. Other parameters can be set, including:

- Dimension of the tetrahedron and distance of the sensors from the faces
- Total duration of the simulation and motion time
- Reference values of Earth's magnetic field vector (defined in Eq. 2.43)
- Standard deviation of the white noise that is applied to "ideal" measurements

The body stands still for a settable time interval, then moves along the prescribed trajectory at a speed that is computed according to the duration of motion. At the end, *TetrAHRS* remains in the final position for the same initial amount of time. This time period is defined as half of the difference between the *Total Time* and the *Motion* time, both expressed in seconds.

Motion along a complex path

The idea of computing quantities that define trajectory and attitude of *TetrAHRS* between two waypoints was presented in Section 2.1.3. The following step involves

the generation of data along a path that is composed of multiple waypoints, defined arbitrarily. Position and attitude related to each waypoint can be set through the interface shown in Figure A.1. The initial condition is the same one that was considered in Section 5.1.

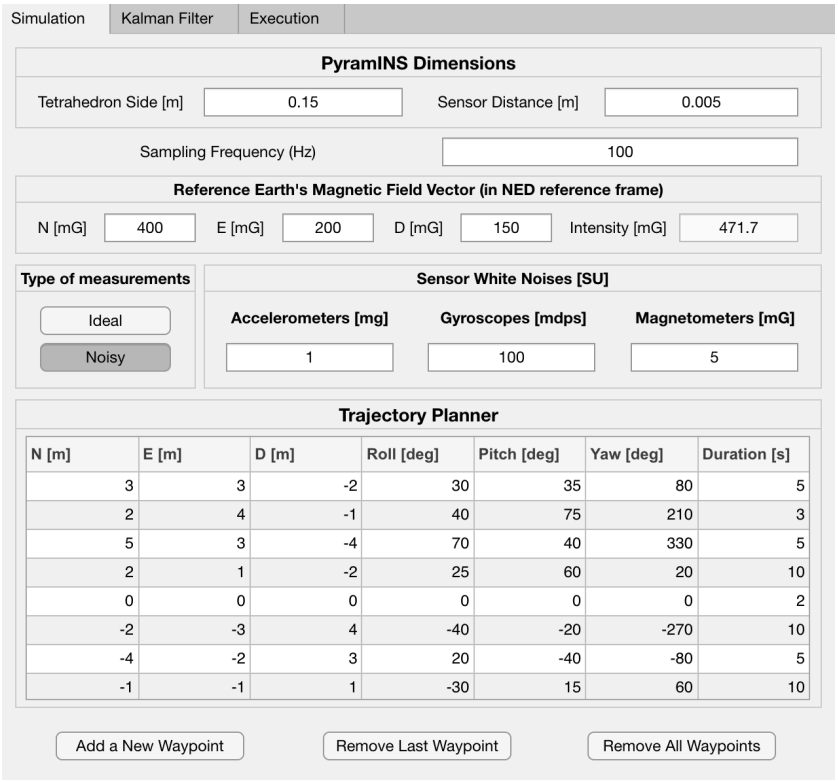


Figure A.1: Simulation: Parameters setting.

This upgrade allows to simulate *TetrAHRS* into an environment that is closer to real-world applications. Finally, a further improvement could be getting data from a real simulator or directly from test flights.

Kalman filtering

The MATLAB® codes that implement equations of the UKF (presented in Section 3.4.1) were embedded into the all-in-one platform. This step was followed by the addition of the other types of nonlinear filters described in Section 3.5. A dedicated Tab in the App (illustrated in Figure A.2) leads to the settings of the Kalman filter. The latter include:

– Uncertainties relative to the state (presented in Section 3.2.3)

– Uncertainties of sensors (reported in Section 3.2.6)

– Choice among the different types of filters

– Selection of the sensor to utilize (all four sensors or a single one)

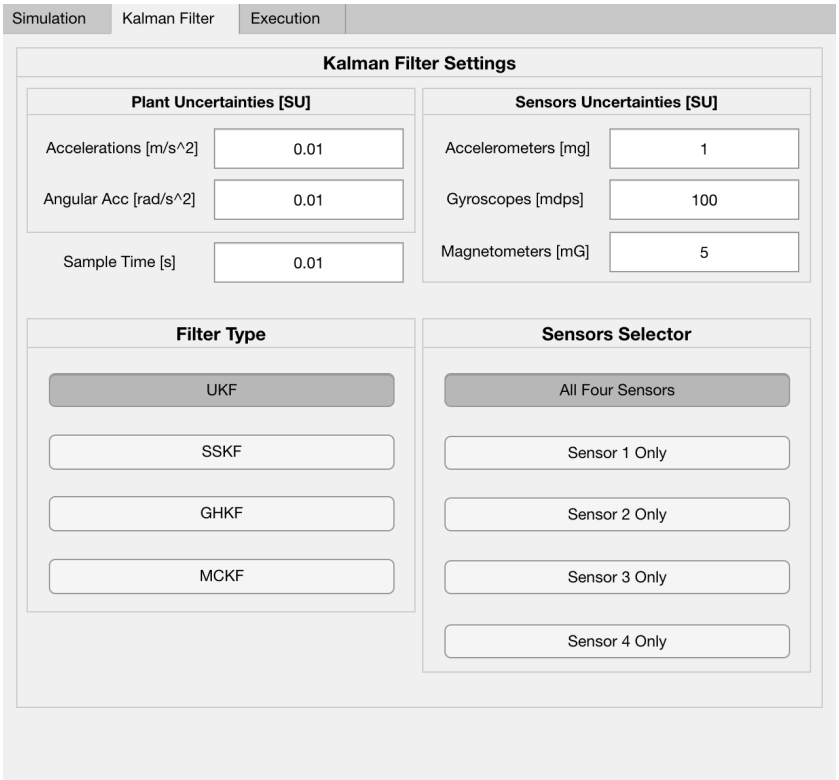


Figure A.2: Kalman filtering: Parameters setting.

It is worth specifying that the covariance matrix $\mathbf{R}$ (described in Section 3.2.6) is able to take into account independent values of variance that characterize the different axes of individual sensors. For simplicity, a single value associated with each type of sensor is considered.

Operating mode selection

Since the goal is to create a comprehensive platform, an operating mode selector was implemented. As shown in Figure A.3, it allows to switch between simulation (generation of data) and Kalman filtering, defined in Chapter 3.

In conclusion, an additional feature was introduced: the possibility to display a video of *TetrAHRS* moving along the desired trajectory with its corresponding attitude, instant by instant. It can be helpful to better visualize the path and rotations performed by the tetrahedron. An example of a frame extracted from the video is presented in Figure A.4.

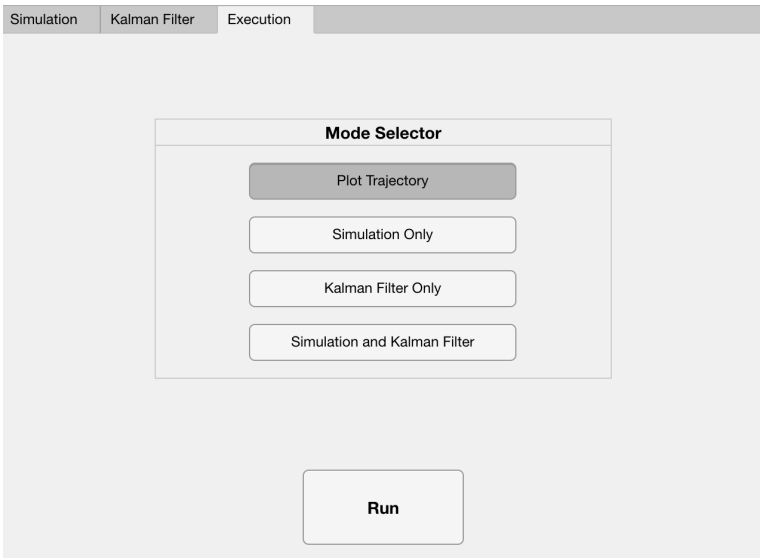


Figure A.3: Operating mode selector.

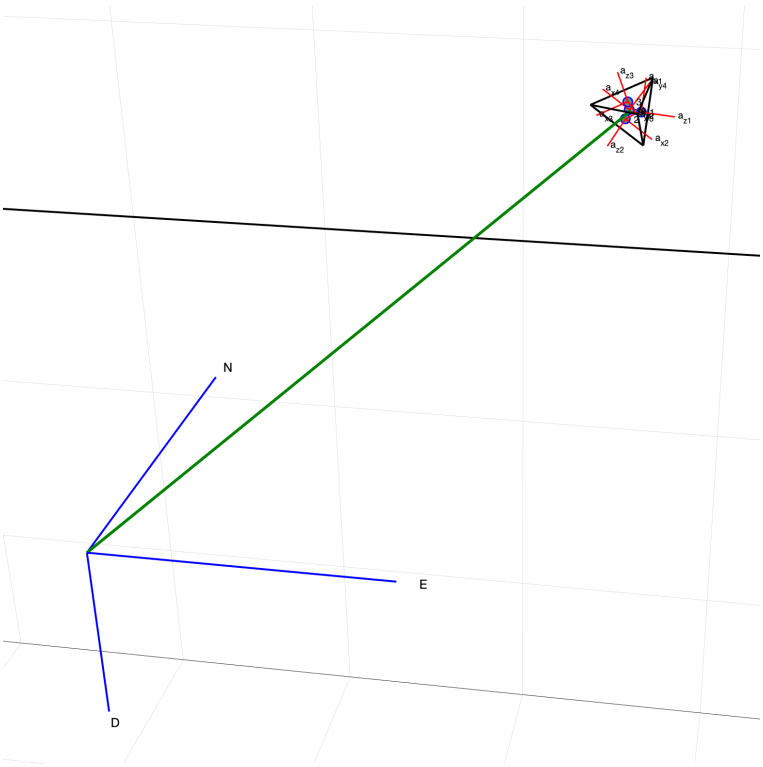


Figure A.4: Frame extracted from a video of the trajectory.

Appendix B

Validation code and results

The validation procedure is run by a dedicated MATLAB® App that was designed for the purpose. The user interface (showed in Figure B.1 and Figure B.2) allows to set the simulation parameters, including the list of trajectories to be checked. At first, simple translations along single axes are performed. Subsequently, rotations around one single axis. Finally, combinations of the two type of motions are considered in order to test all possible paths.

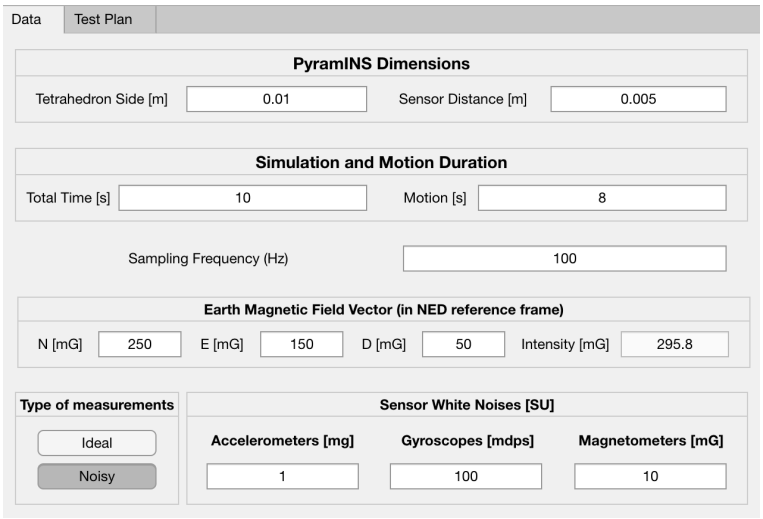


Figure B.1: App for validation: Parameters setting.

The process is repeated with both "ideal" and noisy data. As expected, in the first case an identity is obtained with all tested trajectories. The same applies to noisy measurements, excluding the effects of noise by considering the mean of the signal. It is evident to the eye that data computed from sensors match the corresponding simulated ones. An example of a trajectory is presented in Figure B.3, Figure B.4 and Figure B.5. *TetrAHRS* takes 8 s to move from the initial to the final position with the desired attitude; values are reported in Table B.1.

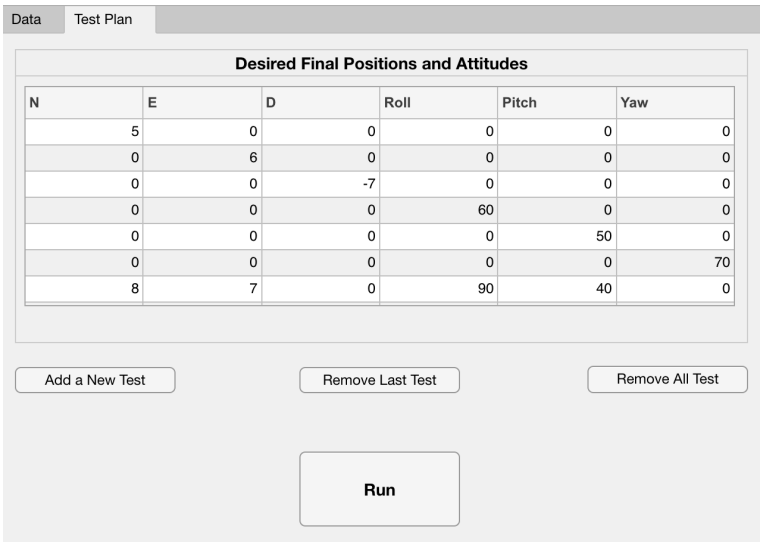


Figure B.2: App for validation: Choice of test trajectories.

Table B.1: Initial and final conditions used in the example of validation.

	$N\ [m]$	$E\ [m]$	$D\ [m]$	$\varphi\ [deg]$	$\vartheta\ [deg]$	$\psi\ [deg]$
Initial condition	0	0	0	0	0	0
Final condition	10	7	-8	80	60	70

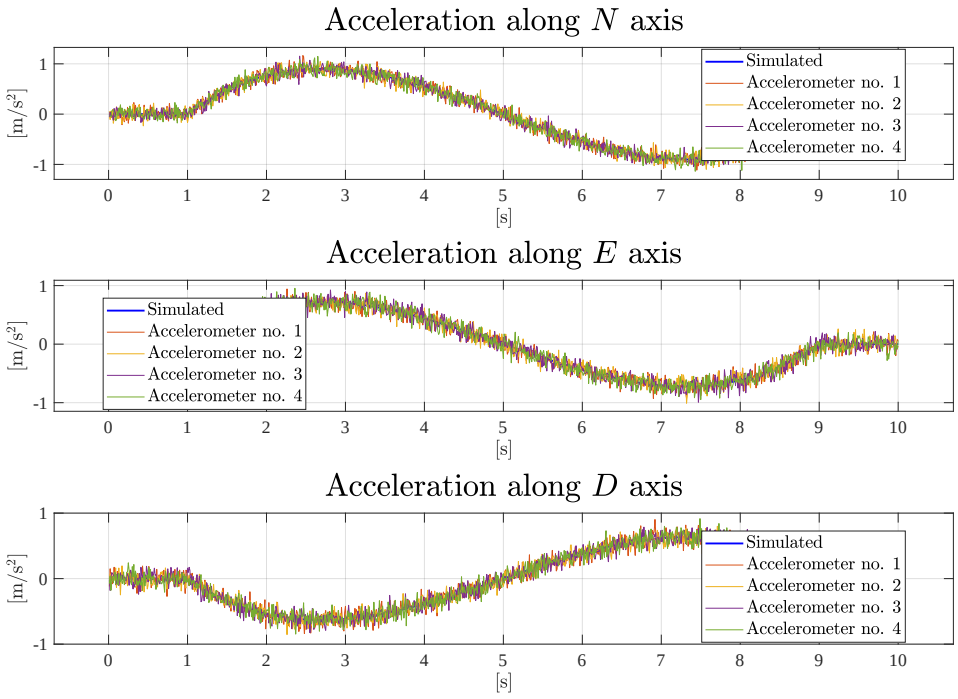


Figure B.3: Validated acceleration vector components in the NED frame.

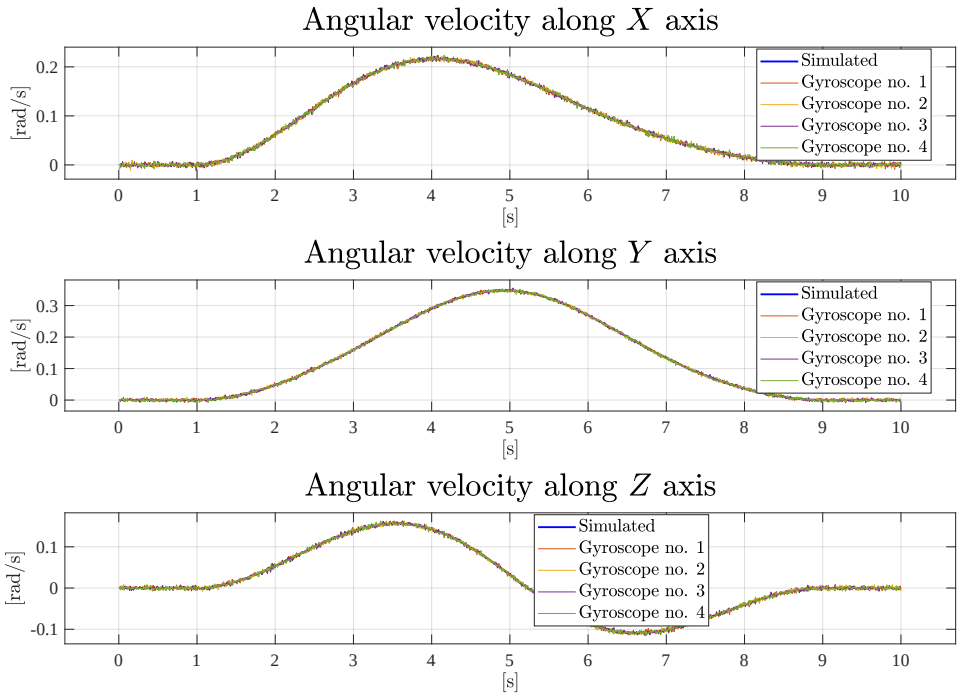


Figure B.4: Validated angular velocity vector components in the G frame.

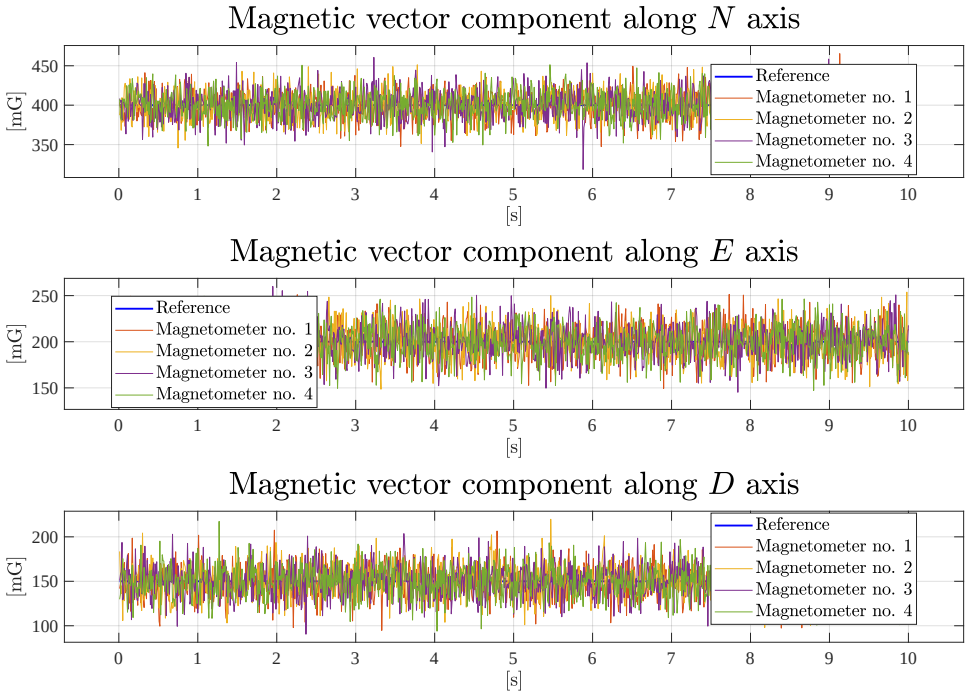


Figure B.5: Validated Earth's magnetic vector components in the NED frame.

Acronyms

AA	Anti-Aliasing filter
AHRS	Attitude and Heading Reference System
AMR	Anisotropic Magnetoresistive
AR	Auto-Regressive
ARW	Angle Random Walk
CF	Complementary Filtering
CM	Center of Mass
CPU	Central Processing Unit
DCM	Direction Cosine Matrix
EMI	Electromagnetic Interference
FS	Full Scale
GMR	Giant Magnetoresistive
GPS	Global Positioning System
HP	High-Pass filter
IFOG	Interferometric Fiber Optic Gyroscope
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LP	Low-Pass filter
MEMS	Micro Electro-Mechanical System
ODR	Output Data Rate
PDF	Probability Density Function
RLG	Ring Laser Gyroscope
RAIM	Receiver Autonomous Integrity Monitoring

RMSE Root-Mean-Square Error

RMOG Resonant Micro Optic Gyroscope

SD Standard Deviation

SQUID Superconducting Quantum Interference Device

UAV Unmanned Aerial Vehicle

Bibliography

[1] Ian Beavers. *The Case of the Misguided Gyro*. 2017. URL: https://www.analog.com/en/analog-dialogue/raqs/raq-issue-139.html (cit. on p. 10).

[2] Sanat K. Biswas, Li Qiao, and Andrew G. Dempster. "A quantified approach of predicting suitability of using the Unscented Kalman Filter in a non-linear application". In: *Automatica* 122 (2020), p. 109241. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2020.109241. URL: https://www.sciencedirect.com/science/article/pii/S0005109820304398 (cit. on p. 49).

[3] Bosch. *Smart sensor combining accelerometer, gyroscope, magnetometer and orientation software*. URL: https://www.bosch-sensortec.com/products/smart-sensors/bno055/ (cit. on pp. 57, 58).

[4] Alberto Brandl et al. "Sensitivity Analysis of a Neural Network based Avionic System by Simulated Fault and Noise Injection". In: *2018 AIAA Modeling and Simulation Technologies Conference*. American Institute of Aeronautics and Astronautics, 2018. DOI: doi:10.2514/6.2018-0122. URL: https://doi.org/10.2514/6.2018-0122 (cit. on p. 35).

[5] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB*. Springer, 2017, p. 83 (cit. on p. 32).

[6] Harold Scott MacDonald Coxeter. *Regular Polytopes*. 1st ed. Methuen & Co., 1948 (cit. on p. 22).

[7] Francesco De Vivo et al. "Ill-conditioned problems improvement adapting Joseph covariance formula to non-linear Bayesian filters". In: *WSEAS Transactions on Electronics*. Ed. by WSEAS Press. Vol. 7. 2016, pp. 18–25. DOI: 10.13140/RG.2.1.3027.0960 (cit. on p. 54).

[8] Francesco De Vivo et al. "Joseph covariance formula adaptation to Square-Root Sigma-Point Kalman filters". In: *Nonlinear Dynamics* 88.3 (2017), pp. 1969–1986. ISSN: 0924-090X. DOI: 10.1007/s11071-017-3356-x. URL: https://doi.org/10.1007/s11071-017-3356-x (cit. on p. 54).

[9] Digilent. *Pmod NAV: 10-DOF Inertial Measurement Unit*. URL: https://reference.digilentinc.com/pmod/pmodnav/start (cit. on p. 59).

[10] A. Doucet, N. deFreitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001 (cit. on p. 56).

[11] D. Gebre-Egziabher et al. "A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors". In: *IEEE 2000. Position Location and Navigation Symposium (Cat. No.00CH37062)*. 2000, pp. 185–192. DOI: 10.1109/PLANS.2000.838301 (cit. on p. 1).

[12] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. 3rd ed. John Wiley & Sons, Incorporated, 2008, p. 106 (cit. on pp. 40, 46).

[13] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Incorporated, 2001, pp. 179–224 (cit. on pp. 39–41, 43).

[14] J.F. Guerrero-Castellanos et al. "Design and implementation of an Attitude and Heading Reference System (AHRS)". In: *2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control*. 2011, pp. 1–5. DOI: 10.1109/ICEEE.2011.6106610 (cit. on p. 1).

[15] Anton J. Haug. *Bayesian Estimation and Tracking: A Practical Guide*. John Wiley & Sons, Incorporated, 2012, pp. 92–119 (cit. on pp. 49, 50, 52, 55, 56, 83).

[16] Agustin Herrera-May et al. "Resonant Magnetic Field Sensors Based On MEMS Technology". In: *Sensors (Basel, Switzerland)* 9 (Oct. 2009), pp. 7785–813. DOI: 10.3390/s91007785 (cit. on pp. 11–13).

[17] Agustín Leobardo Herrera-May et al. "Recent Advances of MEMS Resonators for Lorentz Force Based Magnetic Field Sensors: Design, Applications and Challenges". In: *Sensors* 16.9 (2016). ISSN: 1424-8220. DOI: 10.3390/s16091359. URL: https://www.mdpi.com/1424-8220/16/9/1359 (cit. on p. 13).

[18] Rob J. Hyndman and Anne B. Koehler. "Another look at measures of forecast accuracy". In: *International Journal of Forecasting* 22.4 (2006), pp. 679–688. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.03.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000239 (cit. on p. 74).

[19] Watson Industries. *Attitude and Heading Reference System AHRS-S305*. URL: https://watson-gyro.com/product/attitude/attitude-heading-reference-system-ahrs-s305/ (cit. on p. 4).

[20] TDK - Invensense. *World's Lowest Power 9-Axis MEMS MotionTracking Device*. URL: https://invensense.tdk.com/products/motion-tracking/9-axis/icm-20948/ (cit. on pp. 57, 58).

[21] Reza N. Jazar. *Advanced Dynamics : Rigid Body, Multibody, and Aerospace Applications*. 1st ed. John Wiley & Sons, Incorporated, 2011, pp. 810–814 (cit. on p. 28).

[22] R. D. Kinney. "Characterization of a MEMS Accelerometer for Inertial Navigating Applications". In: UNT Libraries Government Documents Department, 1999. URL: https://digital.library.unt.edu/ark:/67531/metadc687880 (cit. on p. 9).

[23] Bo Lan. "A Brief Guide to the Fundamentals of Passive and Active Rotations in Material Science". In: (Aug. 2015). DOI: 10.13140/RG.2.1.3750.8969 (cit. on pp. 16, 17).

[24] Sebastian Madgwick. "AHRS algorithms and calibration solutions tofacilitate new applications using low-cost MEMS". In: 2014 (cit. on pp. 8–10).

[25] MATLAB. *Estimate Orientation and Height Using IMU, Magnetometer, and Altimeter*. 2021. URL: https://it.mathworks.com/help/fusion/ug/estimate-orientation-and-height-using-imu-magnetometer-and-altimeter.html (cit. on p. 7).

[26] Ronald T. Merrill, Michael W. McElhinny, and Phillip L. McFadden. *The magnetic field of the earth: paleomagnetism, the core, and the deep mantle*. Academic Press, 1996 (cit. on p. 33).

[27] Neda Navidi and Rene Landry. "A New Perspective on Low-Cost MEMS-Based AHRS Determination". In: *Sensors* 21 (Feb. 2021), p. 1383. DOI: 10.3390/s21041383 (cit. on p. 3).

[28] UAV Navigation. *FOG Vs MEMS: An Introduction To Different AHRS Technologies*. URL: https://www.uavnavigation.com/company/blog/fog-vs-mems-introduction-different-ahrs-technologies (cit. on p. 4).

[29] UAV Navigation. *POLAR-300*. URL: https://www.uavnavigation.com/products/ahrs-imu/polar-300 (cit. on p. 4).

[30] NOAA. *Geomagnetism Frequently Asked Questions*. 2014. URL: https://www.ngdc.noaa.gov/geomag/faqgeom.shtml#What_are_the_magnetic_elements (cit. on pp. 33, 60).

[31] Élcio Jeronimo de Oliveira, Waldemar de Castro Leite Filho, and Ijar Milagre da Fonseca. "Inertial Measurement Unit Calibration Procedure for a Redundant Tetrahedral Gyro Configuration with Wavelet Denoising". In: *J. Aerosp. Technol. Manag.* 4 (Apr. 2012), pp. 163–168 (cit. on p. 4).

[32] OMEGA. *Types of Accelerometers*. 2021. URL: https://www.omega.com/en-us/resources/accelerometers-types (cit. on p. 8).

[33] Vittorio M. N. Passaro et al. "Gyroscope Technology and Applications: A Review in the Industrial Perspective". In: *Sensors* 17.10 (2017). ISSN: 1424-8220. DOI: 10.3390/s17102284. URL: https://www.mdpi.com/1424-8220/17/10/2284 (cit. on pp. 9, 10).

[34] W. Rindler. *Essential Relativity: Special, General, and Cosmological*. Springer, 2013, p. 61 (cit. on p. 32).

[35] Tomás Soler. "Active versus Passive Rotations". In: *Journal of Surveying Engineering* 144.1 (2018), p. 06017004. DOI: 10.1061/(ASCE)SU.1943-5428.0000247. eprint: https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29SU.1943-5428.0000247. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29SU.1943-5428.0000247 (cit. on pp. 16, 17).

[36] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. 1st ed. John Wiley & Sons, Incorporated, 2005, pp. 171–185 (cit. on pp. 25, 26).

[37] Brian L. Stevens, Lewis Frank L., and Johnson Eric N. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. 3rd ed. John Wiley & Sons, Incorporated, 2015 (cit. on pp. 14, 16, 18–20, 29, 45).

[38] STMicroelectronics. *9-axis iNEMO inertial module (IMU): 3D magnetometer, 3D accelerometer, 3D gyroscope with I2C and SPI*. URL: https://www.st.com/en/mems-and-sensors/lsm9ds1.html (cit. on pp. 57–59, 61).

[39] STMicroelectronics. *LSM9DS1 adapter board for standard DIL24 socket*. URL: https://www.st.com/en/evaluation-tools/steval-mki159v1.html (cit. on p. 59).

[40] STMicroelectronics. *STM32 Nucleo-144 development board with STM32F412ZG MCU, supports Arduino, ST Zio and morpho connectivity*. URL: https://www.st.com/en/evaluation-tools/nucleo-f412zg.html (cit. on p. 59).

[41] Salah Sukkarieh et al. "A Low-Cost Redundant Inertial Measurement Unit for Unmanned Air Vehicles". In: *I. J. Robotic Res.* 19 (Jan. 2000), pp. 1089–1103 (cit. on p. 3).

[42] SBG Systems. *What is an Attitude and Heading Reference System (AHRS)*. URL: https://www.sbg-systems.com/attitude-heading-reference-systems-ahrs/ (cit. on p. 7).

[43] Youssef Tawk et al. "Implementation and performance of a GPS/INS tightly coupled assisted PLL architecture using MEMS inertial sensors". In: *Sensors (Basel, Switzerland)* 14 (Feb. 2014), pp. 3768–3796. DOI: 10.3390/s140203768 (cit. on p. 36).

[44] Yangzhu Wang et al. "Design and Implementation of an AHRS Based on MEMS Sensors and Complementary Filtering". In: *Advances in Mechanical Engineering* 6 (2014). DOI: 10.1155/2014/214726. eprint: https://doi.org/10.1155/2014/214726. URL: https://doi.org/10.1155/2014/214726 (cit. on pp. 1, 3).