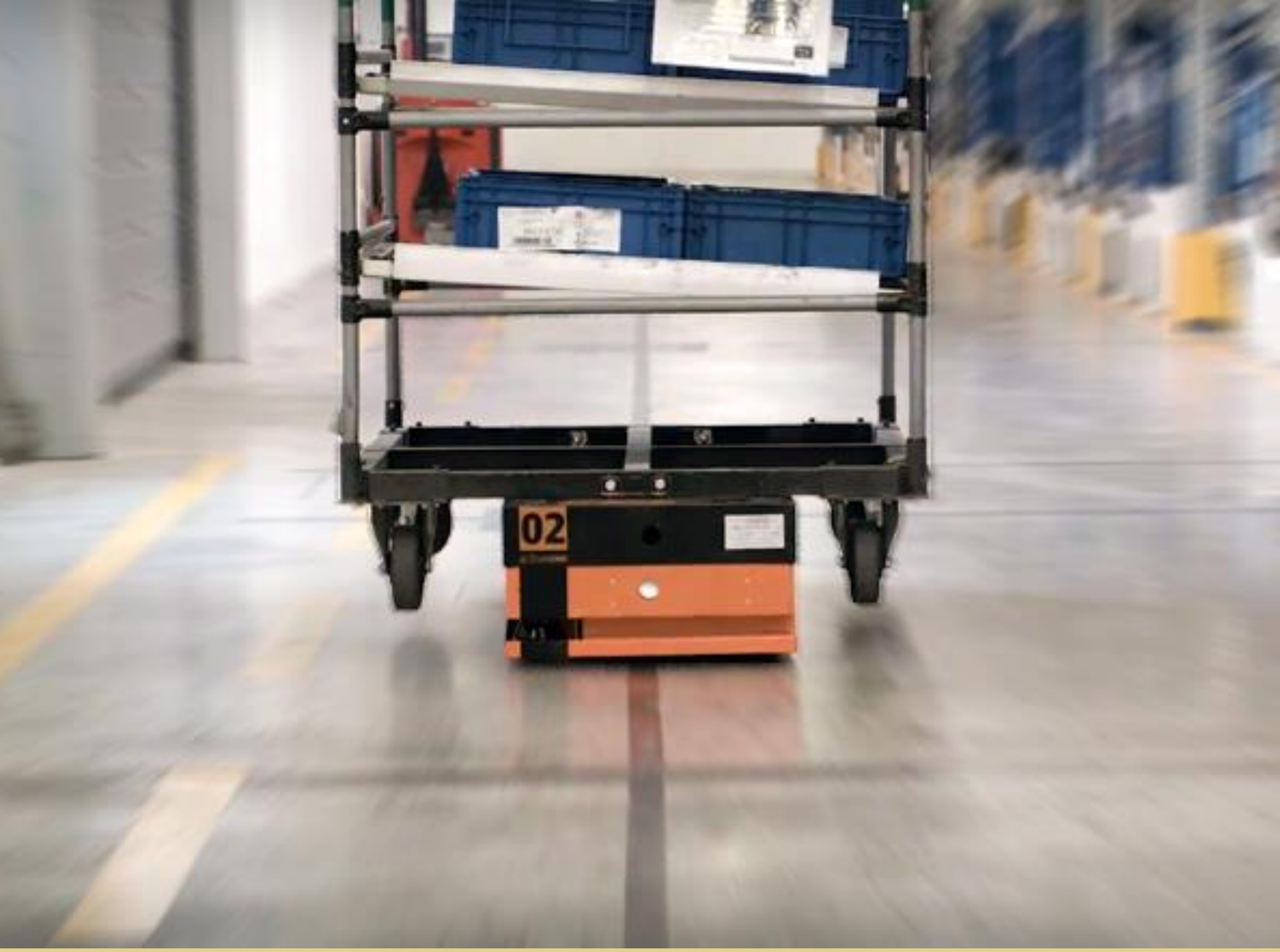


Bruno Miguel Silva Fernandes

# Fusion of inertial data in industrial Automated Guided Vehicles

A dissertation presented for the degree of Master of Science in Electrical and Computer Engineering

September, 2017

· U C ·

Universidade de Coimbra

· U C ·

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

# Fusion of inertial data in industrial Automated Guided Vehicles

**Bruno Miguel Silva Fernandes**

A dissertation presented for the degree of Master of Science in Electrical and Computer Engineering

Coimbra, September 2017

*This page is intentionally left blank.*

# Fusion of inertial data in industrial Automated Guided Vehicles

**Supervisor:**

Prof. Dr. Urbano José Carreira Nunes

**Co-Supervisor:**

Dr. Luís Manuel Conde Bento
Eng. Luis Coelho

**Jury:**

Prof. Dr. Gabriel Falcão Paiva Fernandes

Prof. Dr. Lino José Forte Marques

Prof. Dr. Urbano José Carreira Nunes

Coimbra, September 2017

*This page is intentionally left blank.*

# Acknowledgements

This dissertation, as well as my entire academic career, were possible thanks to the support of several people who had an important role, directly or indirectly, somewhere during this journey.

First, I would like to thank my parents and brother for their support and efforts made over the last five years. A special thanks to Patrícia Henriques for her determination, motivation and love.

I would like to thank my supervisor Dr. Urbano Nunes for the opportunity of this dissertation as well as my co-supervisor Dr. Luís Conde for his assistance and guidance during this work.

I would like to thank *Active Space Technologies* company for providing this opportunity, specially to my co-supervisor Eng. Luis Coelho for leading this project and to the rest of the Automation Team.

I would also like to thank my housemates and colleagues Beatriz Oliveira, Daniel Costa and Luis Parente as well as my other colleagues João Laranjo, João Marques, Jorge Monteiro and Miguel Mendes for all the special memories during this period.

Finally, I would like to thank IEEE-UC Student Branch and NEEEC for giving me the opportunity to be part of the group. Also, a special thanks to the mixed choir, CMUC, for all the amusement and adventure over this last year.

*This page is intentionally left blank.*

# Resumo

Veículos guiados automaticamente são um factor chave na otimização do transporte de materiais no setor industrial. O interesse destes veículos está diretamente relacionado com a sua capacidade de transportar cargas pesadas entre as diferentes etapas de uma linha de montagem, de modo repetitivo, e exigindo uma reduzida intervenção humana. A navegação destes veículos é frequentemente baseada num guiamento óptico ou magnético. No entanto, a inflexibilidade das linhas de guiamento combinada com a sua deterioração ao longo do tempo, promove o desenvolvimento de métodos alternativos ou complementares para a navegação destes veículos.

O objetivo principal deste trabalho é desenvolver e validar um sistema de guiamento inercial, composto por acelerómetros e giroscópios, para complementar um sistema de guiamento magnético existente. O sistema guiado inercialmente deve ser capaz de orientar o veículo, permitindo a sua navegação ao longo de caminhos em linha reta nos quais a banda magnética não está presente. Para concretizar esta tarefa, o projeto e a assemblagem de um sistema de aquisição de dados inerciais é primeiramente realizado, seguindo-se um estudo de algoritmos de fusão sensorial bem como da sua implementação em sistemas embebidos. Os algoritmos de fusão mais precisos e normalmente escolhidos para fundir as medições dos sensores e mitigar erros existentes são computacionalmente pesados, exigindo uma grande capacidade computacional o que é incompatível com dispositivos de baixo custo. Deste modo, este trabalho foca-se em abordagens de fusão sensorial de baixa complexidade, capazes de fundir a informação de sensores inerciais de baixo custo e que possam ser implementados em sistemas embebidos com baixa capacidade computacional. Os resultados demonstram que o sistema de guiamento inercial permite emular adequadamente o comportamento do sistema de guiamento magnético existente, para caminhos em linha reta.

Este trabalho faz parte do projeto AGVPOSYS (em inglês, *Automated Guided Vehicle with innovative indoor POsitioning SYStem for the factory of the future*) liderado pela empresa *Active Space Technologies* e foi realizado nas instalações desta mesma empresa.

**Palavras-Chave:** Filtros Complementares, Filtros de Kalman, Guiamento Inercial, Robôs Móveis, Sistemas Embebidos, Unidade de Medição Inercial.

*This page is intentionally left blank.*

# Abstract

Automated guided vehicles are a key factor in the optimization of the transportation of materials in the industry sector. The interest of these vehicles is directly related with their capability of carrying heavy loads among the different steps of an assembly line, repetitively, while demanding a reduced human intervention. The navigation of these vehicles is often based on optical or magnetic guidance. However, the inflexibility of guidelines combined with its deterioration over time leads to the development of alternative or complementary methods of navigation for these vehicles.

The main objective of this work is to develop and validate an inertial-based guidance system, composed of accelerometers and gyroscopes, in order to complement an existing magnetic-based guidance system. The inertial-based guided system should be capable of guiding the vehicle, allowing its navigation along straight paths in which the magnetic tape is not available. For this task, the design and assembly of an inertial data acquisition system is firstly performed, followed by a study of sensor fusion algorithms as well as their implementation in embedded systems. Standard accurate fusion algorithms implemented to merge sensor measurements and mitigate existing errors are computationally intensive, demanding an high computational capability which is incompatible with low-cost devices. Therefore, this work focus on low-complexity fusion filter approaches, which are capable of merging low-cost inertial sensors data and which can be implemented in embedded systems with low-computational capabilities. The results demonstrate that the inertial-based guidance system properly emulate the behaviour of the magnetic-guided system, for straight paths.

This work is included in the AGVPOSYS (Automated Guided Vehicle with innovative indoor POsitioning SYStem for the factory of the future) project led by the company *Active Space Technologies* and was developed in the facilities of this company.

**Keywords:** Complementary Filters, Kalman Filters, Inertial Guidance, Mobile Robots, Embedded Systems, Inertial Measurement Unit.

*This page is intentionally left blank.*

*"Failure is an option here. If things are not failing, you are not innovating enough."*

— Elon Musk

*This page is intentionally left blank.*

# Contents

*This page is intentionally left blank.*

# List of Acronyms

| | |
|---|---|
| **3D** | Three Dimensional |
| **ACK** | Acknowledgement |
| **ADC** | Analog-to-Digital Converter |
| **AGV** | Autonomous Guided Vehicle |
| **AHRS** | Attitude Heading Reference System |
| **CF** | Complementary Filter |
| **DC** | Direct Current |
| **DCM** | Direction Cosine Matrix |
| **DOSP** | Detection Of Stationary Periods |
| **EKF** | Extended Kalman Filter |
| **GDA** | Gradient Descent Algorithm |
| **GPS** | Global Positioning System |
| **I2C** | Inter-Integrated Circuit |
| **IGCS** | Inertial Guidance Complementary System |
| **INS** | Inertial Navigation Systems |
| **IMU** | Inertial Measurement Unit |
| **KF** | Kalman Filter |
| **LIDAR** | Light Detection and Ranging |
| **LPF** | Low-Pass Filter |

| | |
|---|---|
| **MEMS** | Micro-Electromechanical Systems |
| **MGS** | Magnetic Guidance System |
| **NACK** | Not Acknowledgement |
| **PCB** | Printed Circuit Board |
| **PF** | Particle Filter |
| **PI** | Proportional-Integral |
| **PLC** | Programmable Logic Controller |
| **RFID** | Radio-Frequency Identification |
| **RPY** | Roll-Pitch-Yaw |
| **RSSI** | Received Signal Strength Indicator |
| **SCL** | Serial Clock |
| **SDA** | Serial Data |
| **SLAM** | Simultaneous Localization and Mapping |
| **SPI** | Serial Peripheral Interface |
| **UART** | Universal Asynchronous Receiver-Transmitter |
| **UKF** | Unscented Kalman Filter |

# List of Figures

# List of Tables

*This page is intentionally left blank.*

# Chapter 1

# Introduction

Fully automated factories are about to become reality. The industry sector continually searches for new cost-effective or upgraded solutions to overcome their difficulties. Regarding the transportation, Autonomous Guided Vehicles (AGVs) are an enormous asset since they are able to move heavy loads among different places, efficiently and with reduced human intervention. As a result of a partnership between the University of Coimbra and the company *Active Space Technologies*, the opportunity of working in this project arises and it takes place, mainly, in the company facilities.

Nowadays, there are many different approaches for the navigation of industrial AGVs, for instance, using lasers, cameras or the Global Positioning System (GPS). However, the *Active Space Technologies* company invests in a magnetic tape guidance approach, focusing on the low cost of their vehicles. Therefore, these AGVs navigate along magnetic tape circuits using a Magnetic Guidance System (MGS) composed of Hall Effect sensors, orthogonal to the ground. The main drawbacks of this method are the lack of flexibility since the path is fixed and the damage of the magnetic tape over time due to the rough industrial environment. Therefore, the main goal of this work is to develop an Inertial Guidance Complementary System (IGCS) in order to guide the vehicle along some desired straight paths in which the magnetic tape is not available. For this purpose, the hardware must be developed and assembled, to include inertial sensors in addition to the Hall Effect sensors. Among the different solutions available in the market, the most interesting solution is to add a low-cost Inertial Measurement Unit (IMU) which encapsulates 3-axis accelerometers and gyroscopes.

The measurements retrieved by these sensors should be fused using a fusion filter in order to estimate the orientation of the IGCS developed. Even though there are many relevant approaches used in sensor fusion, this work prioritizes low-computational solutions which can be implemented in low-cost devices. Unfortunately, the orientation estimate is disturbed by various errors since the inertial sensors are vulnerable to noise and temperature variations. Nevertheless, the work developed in this dissertation mitigates some of these errors and describes an inertial guidance algorithm to allow the navigation of the vehicle along straight paths.

## 1.1 Goals and developed work

The goals of this work can be distributed in three major stages. In the first place, the inclusion of inertial sensors in the existing MGS. Secondly, the study of different sensor fusion techniques with the purpose of obtaining an accurate orientation estimate at a reduced computational cost. Finally, the development of an algorithm that allows the navigation of the vehicle along straight lines, using the inertial information.

The work developed in order to achieve these objectives is detailed below.

- **Hardware**
  - Planning of the hardware required for the IGCS;
  - Redesign of the existing Printed Circuit Board (PCB) for the inclusion of the low-cost inertial sensors, using Altium software, and its assembly;
- **Sensor fusion filters**
  - Characterization of the inertial sensors and study of the errors associated with their measurements as well as solutions to mitigate their effects;
  - Study and implementation of existing low-complexity filters for orientation estimating.
  - Adaptation of these filters to the specific environment.
- **Firmware**
  - Implementation of one of the low-complexity filters studied in a dsPIC microcontroller.
  - Development of an algorithm that allows the navigation of the vehicle along straight lines, using the inertial information.
- **Simulation**
  - Evaluation of the performance of the diverse low-complexity filters studied using MATLAB software.

## 1.2 Dissertation structure

This document is organized in six chapters which details the different stages of this work.

The first chapter introduces the context and motivation of this work as well as the main objectives.

In the second chapter, the related state-of-the-art is described, which includes a survey of the navigation systems adopted in industrial AGVs as well as an overview of relevant data fusion techniques used for pose determination in low-cost devices.

The third chapter consists of a characterization of the inertial sensors used and also a study of the errors associated with their measurements. Furthermore, in this chapter, the low-complexity filters are presented as well as the different methods to represent the attitude of a rigid body.

In the fourth chapter, the industrial environment of this work is presented along with the prototype of the IGCS developed. Adaptations to the studied filters and an inertial guidance algorithm are also discussed in this chapter.

The fifth chapter presents the tests and results of the implemented algorithms in the IGCS prototype.

Finally, the sixth chapter provides some final conclusions and a discussion of the future work.

*This page is intentionally left blank.*

# Chapter 2

# Related State-of-the-art

The automation of transportation in the production, trade and service sector is a key factor in the optimization of intralogistics [1]. Because AGV Systems are cost-effective and can operate as standalone systems with higher efficiency, they are widely used in the transportation of materials [2]. AGVs can operate in many industries, such as pharmaceutical, chemical, hospital, manufacturing, automotive and so on, since they have customized designs to best suit an industry's requirements.

The aim of this chapter is to present the related state-of-the-art of the existing navigation systems available in the industry of AGVs as well as to introduce some techniques used for pose estimation. In the first section, a survey of the most used navigation systems is made, along with a discussion of the major advantages and drawbacks of each method. Then, in the second section, it is presented the background of the most relevant data fusion techniques used for pose determination.

## 2.1 Navigation systems for industrial AGVs

The initial era of AGVs began in America in 1953 with its invention, followed by Europe a few years later [3]. The first AGV was simply a tow truck that followed a wire in the floor. In 1973, AGVs were commercialized for the purpose of moving car bodies in an assembly line [4]. With the growth of technology and industry more AGVs manufacturers emerged over the years culminating in new and improved solutions for the navigation of these vehicles.

Navigation refers to the science of finding a way from one place to another. In the environment of mobile robots, it is the process of accurately determining the vehicle's pose and plan a path to move towards its destination. Regarding the navigation of AGVs in the industrial environment, there are different navigation procedures which can be distributed in three categories according to the structure of the guide path: guideline navigation, markers navigation and free navigation. Generally, a good navigation system rely on information of the combination of several navigation procedures. The identification of payloads or destinations is usually done using bar coding or Radio-Frequency Identification (RFID) [5] techniques.

The navigation of mobile robots is usually related with dead-reckoning and odometry. The former concept is described as the process of estimating the current position of a vehicle using the previously determined position, its course and the estimated or known speeds over a known interval of time. Evidently, dead reckoning is vulnerable to cumulative errors. A dead-reckoning sensor system and a tracking algorithm for mobile robots was developed in [6, 7]. Odometry is considered as a sub topic of dead-reckoning since the distance travelled over a period of time is estimated exclusively from wheel displacement calculations. Thus, it is sensitive to errors which can be systematic (for instance, unequal wheel diameters) or non-systematic (due to wheel slippage and uneven floors). In [8, 9] it is discussed an odometry calibration method and the correction of systematic odometry errors in mobile robots, respectively.

The following sections briefly introduce the navigation procedures of AGVs and summarize the relevant pros and cons.

### 2.1.1 Guideline navigation

Navigation along a physical guideline is the oldest method used in mobile robots navigation. Due to the low-cost and straightforward installation of the guideline and to the software's low requirements these procedures are still implemented by many AGVs manufacturers. The physical structure can be located on or in the floor. Usually, the distance travelled is measured by wheel odometers.

**Wire Guidance**

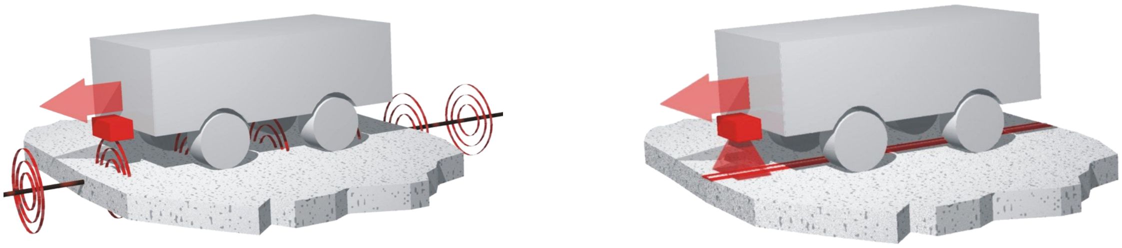
One of the methods adopted in AGVs navigation uses a charged wire embedded in the floor to represent the guide path [3]. Antennas composed of inductive coils are mounted in the vehicle. The voltage induced in the coils is directly related with the field between the wire and the antennas. Since the wire has to be buried in the floor, the whole process is expensive, inflexible and consumes a lot of time. Regardless these limitations, this method is not affected by deterioration or dirtiness which usually is the main drawback of the Magnetic Tape and Optical methods described below.

**Magnetic Tape Guidance**

Another navigation method adopted by many AGV manufacturers uses magnetic tape glued on the floor's surface to mark the guide path of the vehicle [10]. Usually, this tape is detected by an array of Hall Effect sensors mounted underneath the vehicle. This approach is widely used in industry areas such as automotive, electronics and assembly lines. In spite of the fact that paths are fixed, they are easily and quickly adjustable, overcoming the limitations of Wire Guidance.

**Optical Guidance**

In Optical guidance, the guide path is a coloured strip, painted or taped down on the floor. The vehicle has a sensor to detect the color variation between the coloured strip and the floor. Sometimes, it is used an ultraviolet light source to illuminate the strip. In terms of performance, this method is similar to Magnetic Tape guidance. The relevant benefit is that in Optical guidance the sensor can read coded tape and optical markers.



(a) Wire guidance.

(b) Magnetic tape and optical guidance.

Figure 2.1: Navigation methods along a physical guideline.[1]

## 2.1.2 Markers navigation

Navigation along physical guidelines has been the main technology for many years because of its simplicity and low cost. However, with the constant growth of industry and technology, new navigation methods have emerged. In these methods, the guide path is not physically fixed. Instead, it is established in a microcontroller or a computer. In comparison with guideline methods which simply follow the band, markers and free navigation methods have the advantage to know the pose of the vehicle in the world in every time sample.

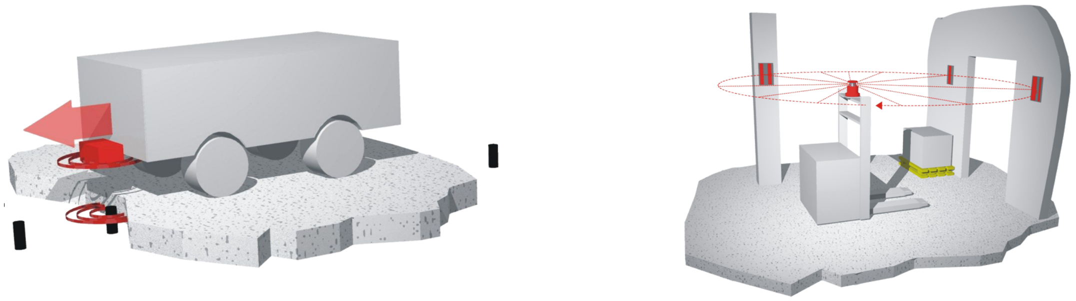
[1] Images from: `http://www.goetting-agv.com/solutions`

### Magnetic Spot Grid Guidance

In this method of navigation, the operating area is marked with magnetic reference points placed on or in the floor in a grid pattern or as a series of points [3]. These reference points, which positions are stored in the vehicle's memory, are detected by a sensor, typically a magnet sensor bar constituted by hall sensors which convert the magnetic field flow into a voltage proportional to the strength of the field. This method provides the vehicle's navigation within the grid area which can be easily expanded or adjusted.

### Laser Guidance

One of the most promising methods of navigation for AGVs is the laser navigation [1]. Multiple-fixed reference points are mounted on the working environment. These reference points with reflective proprieties are placed on the walls or pillars at known positions and they can be detected by a laser scanner, mounted on the top of the vehicle. The position of the vehicle is determined via triangulation using the reflection time information returned by the laser, and then, it is matched with the programmed path. The high popularity of this method is due to its high accuracy, reliability and flexibility. Even though the system's layout can be easily expanded, the laser scanner head must have a visibility of 360 degrees.



(a) Magnetic spot guidance.

(b) Laser guidance.

Figure 2.2: Navigation methods using markers on the ground or on the walls.[2]

### Inertial Guidance

In Inertial guidance, magnets are embedded in the floor at known positions. The vehicle uses a magnet sensing device to detect those reference points as well as a gyroscope to measure the vehicle's heading. The distance travelled can be calculated using encoders. It is possible to obtain an accurate pose using the feedback from the three sensors. One of the main AGV manufacturers using this technology is Savant Automation [3].

[2] Images from: `http://www.goetting-agv.com/solutions`

[3] For more details visit `http://www.agvsystems.com/navigation-landing/`.

### 2.1.3 Free navigation

Free navigation methods are distinguished from physical markers methods by not using any dedicated physical structure to help the robot's guidance, apart from the robot's sensors. The main drawback of these methods is the high software requirements to treat the data.

#### Global Positioning System Guidance

The GPS is a navigation system that grants location and time information to any GPS receiver anywhere on Earth, provided there are a line of sight to four or more GPS satellites. This method is totally free of field installations and it is very flexible. In spite of the advances of this technology which resulted in accuracy improvements, the GPS is not recommended for indoor positioning due to the slow update rate and the sensitivity to blocking of the satellite signals [11].

#### Natural Guidance

In Natural guidance, a set of reference images of the operating area should be stored in the memory of the vehicle. A laser or a camera should be used in order to collect features during the navigation [12]. The pose of the vehicle is calculated through the comparison of the natural and the collected features. This type of navigation is particularly interesting when high accuracy is needed for crowded environments or small sized facilities.

#### Three Dimensional (3D) Map Guidance

In the subject of 3D map building and navigation, the most relevant method is Simultaneous Localization and Mapping (SLAM). The SLAM problem focuses in the possibility for an autonomous vehicle to start in an unknown location of an unknown environment and then incrementally build a map of the surroundings while simultaneously computing the absolute vehicle's location using the map constructed [13]. Usually, when referring to SLAM, two technologies are used: cameras [14] and Light Detection and Ranging (LIDAR) [15]. The former technology is cheaper but does not work in the absence of light. The latter overcomes this limitation but it fails to detect walls of glass, since the rays could pass through.

## 2.2 Data fusion techniques for pose determination

The navigation systems presented in the last section require a set of sensors to determine the pose of the vehicle along the guide path. Therefore, to obtain an accurate output, the data retrieved by the sensors must be fused through an algorithm. The most straightforward data fusion architecture is the centralized data fusion, where all the measurements are directly fed to a master filter (see Figure 2.3). A deeper understanding of this topic, as well as of other data fusion architectures and some examples can be found in [16].

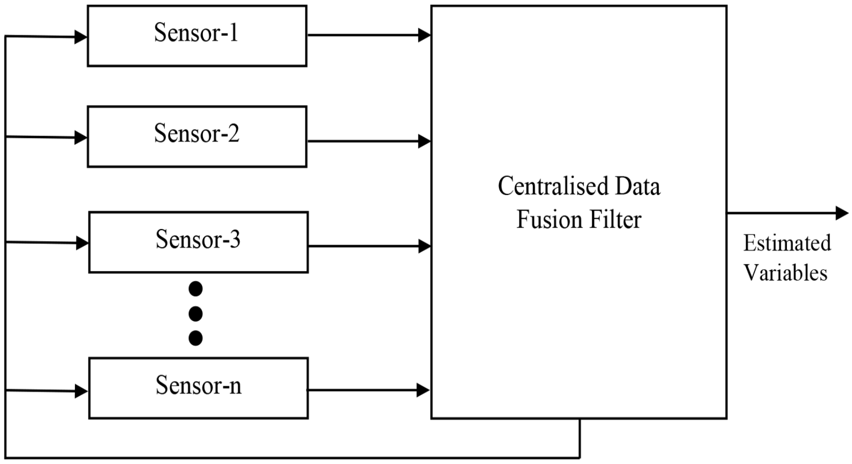


Figure 2.3: Centralized Data Fusion.

The determination of the pose of a rigid body is obtained through the relation between the body frame and a fixed reference frame, usually referred as inertial frame. The combination of sensors or markers in these frames is widely adopted when an accurate estimation of the pose is desired. For instance, the work developed in [17, 18] fuses the data collected by an IMU with data retrieved by wireless communication (a technique called Received Signal Strength Indicator (RSSI)) in order to estimate the localization of a vehicle. In [19] it is used an IMU along with a GPS unit instead. Other approaches, e.g. [20], use visual sensors together with IMUs for the same purpose.

In the context of this work, the navigation of the AGV in specific stages should be exclusively based on the data retrieved by inertial sensors, such as accelerometers and gyroscopes. The main challenge of these sensors is to minimize the accumulated errors during the motion of the vehicle. A summary of gyroscope and accelerometer error sources can be found in [21] and is also described in this work. Diverse techniques can be applied to mitigate these errors when estimating the orientation or position of a vehicle. For instance, in [22], a method to only integrate acceleration in moments of effective motion was developed, reducing the errors due to double integration of accelerometers. In [23, 24] it was developed an automatic mitigation strategy to compensate drifts introduced by low-cost IMUs. Also, in [25], a self-calibration approach was considered.

Even though inertial sensors can accumulate considerable errors, it is possible to achieve relative positioning for short periods of time [26]. The advances in Micro-Electromechanical Systems (MEMS) technology culminated in the manufacturing of small and light Inertial Navigation Systems (INS). This progress has expanded the range of applications of these sensors, attracting many researchers to develop algorithms that can accurately estimate the orientation and position of a rigid body. In this context, the most significant approaches to fuse sensor data and mitigate sensor errors are Kalman filtering and Complementary filtering. Nevertheless, there are plenty of other data fusion methods, such as Particle filtering, H-infinity filtering or Luenberger observers, which are unsuitable for this work due to their limited applications or intensive computational requirements. In [16] it is summarised the pros and cons of the most widely implemented filtering techniques in robotics. In [27] it is also provided a survey of filtering methods, particularly for non-linear systems.

### 2.2.1 The Kalman Filter

The Kalman Filter (KF) has been widely used for decades to estimate the behaviour of linear systems under Gaussian noise. The original derivation of the this filter [28] was based on the least squares approach [29]. Although, the same equations can be derived from pure probabilistic Bayesian analysis [30]. The Bayesian derivation of the KF proves that when the noise is Gaussian, the KF is the optimal filter. The least squares derivation proves that it is the optimal linear filter, regardless of the probability density function of the noise [31].

The KF is a recursive prediction-correction method, as shown in Figure 2.4 [32], used to estimate the state of a dynamic system from noisy observations or measurements. The state of a system concerns to the set of dynamic variables which fully describe the system (for instance: position, velocity or orientation) while the noise means that the measurements have some associated uncertainty [31]. The two-step equations of the KF in the discret time are detailed below [32].

The predictor step equations are responsible for projecting forward the current state ($\hat{x}$) and error covariance ($P_k$) estimates to obtain the *a priori* estimates for the next time step.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{2.1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2.2}$$

The correction step equations are responsible for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate. In this step, the Kalman gain, $K_k$, is calculated.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{2.3}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.4}$$

$$P_k = (I - K_k H) P_k^- \tag{2.5}$$

In equations (2.1) to (2.5), the matrices $A$ and $H$ represent the transition matrix of the dynamic model and the measurement model, respectively. The process noise covariance $Q$ and the measurement noise covariance $R$ are parameters to be tuned and, if they are constant, the Kalman gain $K_k$ and the error covariance $P_k$ will stabilize quickly.

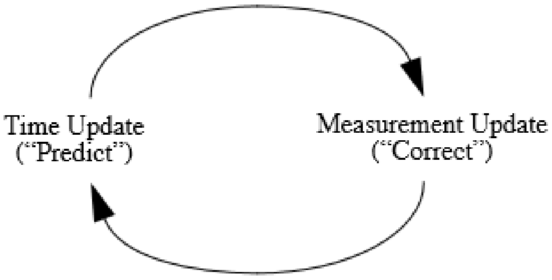


Figure 2.4: The discrete Kalman Filter cycle.

The KF is widely used due to its simplicity in designing, coding, and tuning, making it convenient for real-time processing in many engineering applications [31]. However, in non-linear models it shows deficient robustness. Therefore, to overcome this issue, many non-linear approaches of the KF have been developed over the years, being highlighted the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF).

### Extended Kalman Filter

Since the KF works as a least square error optimizer, it requires a linear system to work properly. The aim of the EKF is to linearise the system around its current state in order to estimate non-linear systems. This approach involve the computation of Jacobian matrices to derive the state-transition and observation matrices. Therefore, equations 2.1 to 2.5 can be rewritten as equations 2.6 to 2.11 [33].

Similarly to the KF, in the predictor step of this filter, the current state and uncertainty of the system at the previous time step are propagated.

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}) \tag{2.6}$$

$$P_k^- = A_k P_{k-1} A_k^T + Q, \tag{2.7}$$

$A_k$ is the Jacobian matrix containing the partial derivatives of the system function $f(\cdot)$ with respect to the current state, evaluated at the posterior state estimate of the last time step.

$$A_k = \left.\frac{\partial f(x)}{\partial x}\right|_{x=\hat{x}_{k-1}} \tag{2.8}$$

In the correction step, the prior state estimate is rectified with a full measurement.

$$K_k = P_k^- H_k^T (H P_k^- H_k^T + R)^{-1} \tag{2.9}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-)) \tag{2.10}$$

$$P_k = (I - K_k H_k) P_k^-, \tag{2.11}$$

$H_k$ is the Jacobian matrix that contains the partial derivatives of the measurement function $h(\cdot)$ with respect to the current state, evaluated at the prior state estimate.

$$H_k = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}_k^-} \tag{2.12}$$

Since the Jacobian matrices $A_k$ and $H_k$ are evaluated at the most recent state estimates, they must be computed on-line, leading to high computational costs [33]. Furthermore, if the process model is inaccurate, the EKF will diverge due to the Jacobians, which represent the linearisation of the model.

Nevertheless, the EKF is widely adopted in many works. For instance, in [24,34] it was applied to fuse data from a wireless Ultra-Wideband with data from inertial sensors in order to build a real-time indoor navigation system and tracking of mobile robots in factories and warehouses. In [35,36] an EKF was used to fuse the data from several IMUs in a wearable system for real-time human motion capture. In [37] it was applied for estimating the 3D orientation of a rigid body using an IMU and a tri-axial magnetic sensor and is it was also used for pedestrian navigation in [38,39].

### Unscented Kalman Filter

The first-order linearisation of the non-linear system used in the EKF can introduce large errors in the true posterior mean and covariance values, particullarly if the predict and update functions are highly non-linear, leading to the divergence of the filter or to a deficient performance [40]. The UKF overcomes this limitation by employing an unscented transform [16]. Therefore, instead of linearising the system using Jacobian matrices, the UKF uses a set of deterministically weighted samples [33], named as $\sigma - points$. This sampling points are chosen around the current mean. Then, they are propagated through the non-linear system in order to obtain the posterior mean and covariance estimates. The detailed algorithm as well as the pseudo-code for the UKF can be found in [41].

Even though the UKF and the EKF perform evenly well on non-linear systems [42], the former does not require the calculation of Jacobians, but even so its computational complexity is a little higher [43]. In [44], it was made a comparison of Unscented and Extended Kalman Filtering for estimating quaternion motion. Also, a quaternion-based UKF for estimating the attitude of a rigid body is described in [45].

Kalman filters are proved to be the best choice when the system requires a high accuracy. However, their high computational complexity makes them inappropriate for this work. The next section introduces the complementary filtering which, compared with Kalman filtering, has a reduced computational burden, and therefore it can be implemented in low-cost devices.

### 2.2.2 Complementary Filters

In complementary filtering the filter is obtained by an analysis in the frequency domain. The goal is to combine low pass and high pass filters to filter high frequency and low frequency signals, respectively. The basic Complementary Filter (CF) is shown in Figure 2.5, where $x$ and $y$ are noisy measurements of a $z$ signal and $\hat{z}$ is the output estimate produced by the filter [46]. It is assumed that the noise in $x$ is mostly low frequency and the noise in $y$ is mostly high frequency. Therefore, $G(s)$ is a low-pass filter and $1 - G(s)$ is its complement.

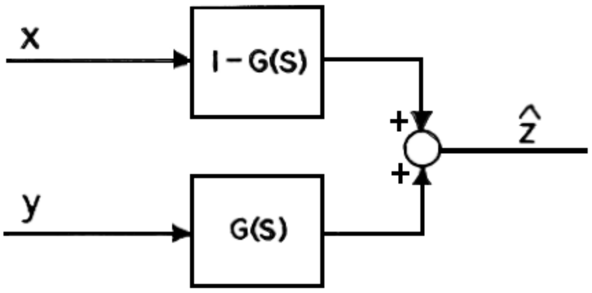


Figure 2.5: The basic complementary filter.

Complementary filtering is a common alternative to Kalman filtering because of its simplicity and effectiveness [47], being the appropriate choice when the system demands a low computational complexity. The work developed in [48] demonstrates that the execution time of a CF cycle can be around 5% of the execution time of an EKF cycle. Even though, the EKF outperforms the CF and its generality and adaptability allows adding further sensory information without the need to completely re-design the filter. Nevertheless, CFs can also be adapted to best fit a system requirements by estimating sensors bias and mitigating measurement errors [49, 50].

The study of CFs has been growing over the last few years due to their good accuracy while keeping low computational needs, which is highly significant in embedded systems, and therefore in the work of this dissertation. Two relevant approaches in complementary filtering are the Mahony filter [51] and the Madgwick filter [52] and will be described later in this document.

*This page is intentionally left blank.*

# Chapter 3

# Background and Supporting Technologies

Automated Guided Vehicle Systems require a set of sensors to obtain the position and orientation of a vehicle along a desired path. This chapter describes the models of the motion sensors used in this work as well as the algorithms associated with the data fusion techniques introduced in the previous chapter.

In the first section, an overview of the inertial sensors used in this work is presented along with a characterization of the most common error sources of these sensors. The second section introduces the most common techniques adopted for attitude representation of a rigid body as well as the relation among them. Later, in the third section, the filters implemented for estimating the orientation of a rigid body are presented.

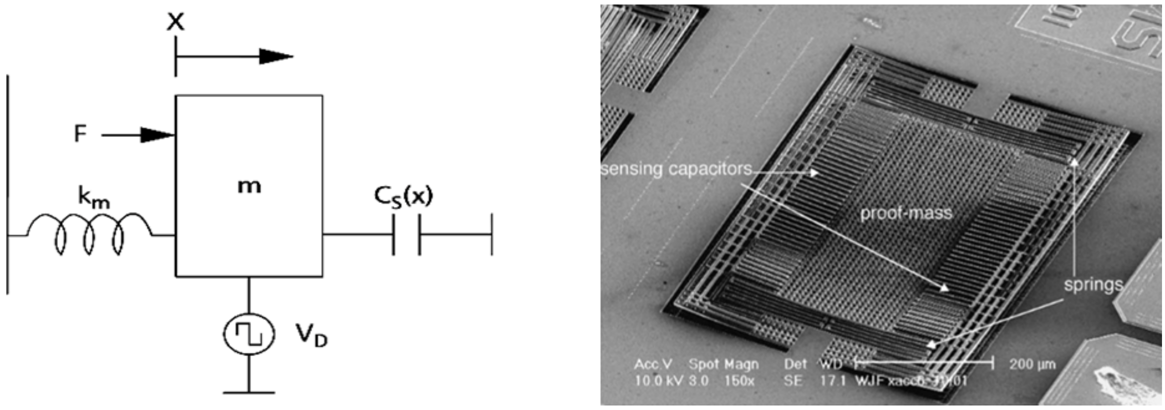
## 3.1 Inertial Sensors

The development of MEMS inertial sensors has been driven by the need for inexpensive sensing solutions in military and industrial applications. The miniaturization of these sensors also increased their use in personal devices due to their low-cost, lightweight, low power consumption, small size and short start-up time [21, 53]. However, their fabrication process introduce large bias instability and noise, affecting the accuracy of the estimates obtained from these sensors [23].

With the advances in inertial sensing technologies, the topic of inertial navigation has been growing as well as novel techiques to fuse sensor data and improve the accuracy of these systems [21]. Since the scope of this work is to build a low-cost inertial system, consisted of accelerometers and gyroscopes, to guide an AGV along straight paths, the models and common error sources of these sensors are presented in this section.

### 3.1.1 Accelerometer

Accelerometers are devices capable of measuring acceleration. As shown in Figure 3.1, MEMS accelerometers include a proof-mass suspended on a fixed frame by a spring [54]. Sensing capacitors measure the deflection of the proof masses produced by accelerations. Therefore, acceleration can be generated by the motion of the sensor or due to gravity and is typically measured in units of gees ($g$), where $1g$ represents the acceleration due to gravity at the surface of the Earth[1].



(a) Electromechanical schematic [54].

(b) Structural model [56].

Figure 3.1: MEMS accelerometer scheme.

Multi-axis accelerometers are often used for gravity sensing since, at a rest position, they only measure the gravitational acceleration. A simple 3-axis accelerometer can measure the orientation of the gravity directly, which is widely used, for instance, in mobile devices for screen orientation, gaming and compass tilt compensation.

Besides gravity sensing, they can also be used to measure motion or vibration. Thus, they play an important role evaluating the vibration on cars, machines or seismic activity. In inertial navigation these devices can be used to calculate the orientation, velocity or position of a vehicle. The velocity and position estimates are obtained, respectively, through the single and double integration of the accelerometer measurements [21].

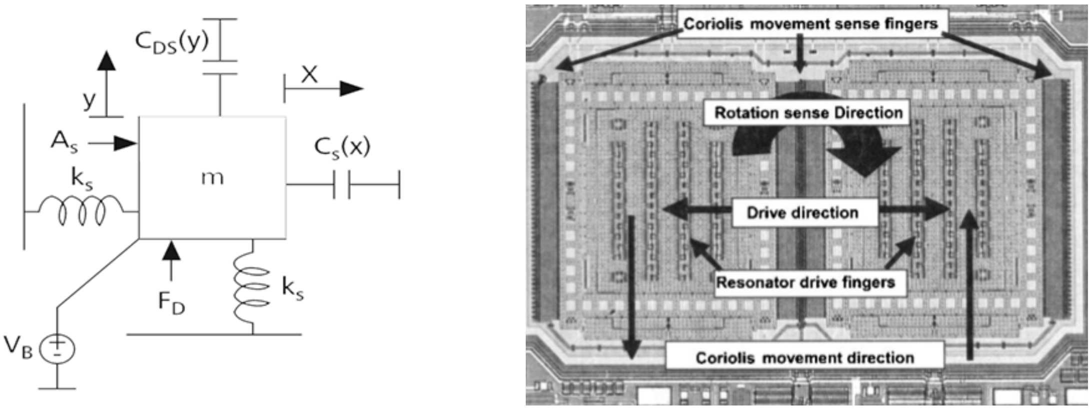
Unfortunately, MEMS accelerometers measurements are vulnerable to errors which make them unsuitable for certain applications. Therefore, some calibration techniques are required and will be presented later in this section.

[1] Even though the standard gravity is defined as 9.80665 $m/s^2$, the effective gravity varies by around 0.7%, from 9.76392 $m/s^2$ on the Nevado Huascarán mountain in Peru to 9.83366 $m/s^2$ at the surface of the Arctic Ocean [55].

### 3.1.2 Gyroscope

Gyroscopes are devices capable of measuring the angular rate of rotation, commonly expressed in degrees per second ($dps$). Contrarily to accelerometers, MEMS gyroscopes electromechanical system has two degrees of freedom which include a sense axis and a drive axis [54], respectively represented by $x$ and $y$ as shown in Figure 3.2.

The principle of operation of these sensors is based on measuring the Coriolis force[2]. The proof mass is driven into sinusoidal motion along the drive axis. Then, when the gyroscope is rotated, a secondary vibration is induced along the perpendicular sense axis due to the Coriolis force and the angular velocity can be measured.



(a) Electromechanical schematic [54].

(b) Structural model [57].

Figure 3.2: MEMS gyroscope scheme.

Gyroscopes are used in many applications, such as stability assistance, gyrocompass[3] or navigation. In fact, these devices can help in the stabilization of planes, bicycles or robots and also detect and measure the movement of some hand-held devices.

In navigation, gyroscopes are used to measure the orientation of a vehicle, which require the integration of the angular rate of rotation. Similar to MEMS accelerometers, their measurements are vulnerable to errors which propagate faster when the data is integrated. The next section will present the common error sources in MEMS accelerometers and gyroscopes.

[2] Coriolis effect states that in a frame of reference rotating at angular velocity $\omega$, a mass $m$ moving with velocity $v$ experiences a force equal to $-2m(\omega \times v)$ [21].

[3] A gyrocompass is a device used to find the true North. Contrarily to regular compasses, they are not affected by external magnetic fields, being used to measure relative changes in direction in boats and planes.

### 3.1.3 Error sources and models

As previously mentioned, MEMS accelerometers and gyroscopes measurements are vulnerable to errors which may be incompatible with certain applications, mainly when an integration of the data is required. In this section, the common error sources of these sensors as well as their models will be presented. A deeper analysis of the errors and the result of their integration can be found in [21,58]

The three main sources of errors affecting accelerometers and gyroscopes measurements are: constant bias, temperature variation and noise.

The bias of a sensor is the average signal output when the sensor is at rest, also referred as the non-zero offset. It can be estimated by calculating the average of a certain number of measurements as long as the sensor is static during the process. Once the bias is known, it can be posteriorly compensated by simply subtracting the bias from the output measurements, as demonstrated in equation 3.1.

$$CalibratedData = MeasuredData - Offset \tag{3.1}$$

Unfortunately, the constant bias removal is a more complicated process in accelerometers due to gravity, which appears as a bias. Therefore, the precise orientation of the device respecting to the gravitational field must be know [21]. The offset is estimated by removing the gravity from the average of the measurements. This process can easily propagate errors since it defines the direction of gravity relatively to the sensor.

The output samples obtained from MEMS sensors are perturbed by an uncorrelated white Gaussian noise sequence. Furthermore, due to flicker noise in the electronic components, the bias of these sensors drifts over time. These oscillations are usually modelled as a random walk [21].

The temperature of the sensor may change over time due to self-heating or changes in the environment resulting in fluctuations of the bias.

Even though these are the main sources of errors affecting the accuracy of the estimates of MEMS accelerometers and gyroscopes, other error sources as scale factors or axis misalignments should also be considered. Usually, when a combination of different sensors is considered, fusion algorithms attenuate the drifts in real-time, leaving the initial constant offset to be eliminated.

The characterization of MEMS accelerometers and gyroscopes leads to the models presented in equations 3.2 and 3.3. The calibrated acceleration and angular velocity estimates in the sensor frame, are represented by $\vec{a}$ and $\vec{\omega}$, respectively [59, 60].

$$\vec{a} = K_a \cdot (\vec{a}_{true} + \vec{g}) + \vec{b}_a + \vec{\eta}_a \tag{3.2}$$

$$\vec{\omega} = K_\omega \cdot \vec{\omega}_{true} + \vec{b}_\omega + \vec{\eta}_\omega \tag{3.3}$$

$K_a$ and $K_\omega$ represent scale factor matrices, $\vec{a}_{true}$ and $\vec{\omega}_{true}$ are measurements due to sensor motion, $\vec{b}_a$ and $\vec{b}_\omega$ are bias vectors, $\vec{\eta}_a$ and $\vec{\eta}_\omega$ represent uncorrelated white Gaussian measurement noise and $\vec{g}$ is the gravity vector. Bias vectors result of the sum of constant bias, bias drift over time and bias variation as a function of temperature [58].

## 3.2 Attitude representations

The orientation of a rigid body can be represented in the Euclidean space using a variety of different methods. Therefore, the choice of the attitude representation can provide a straightforward or intractable solution for the same problem [61].

The most popular methods for attitude representation are the Euler method, the Direction Cosine Matrix (DCM) method and the Quaternion representation. The use of quaternions is particularly advantageous in embedded systems because they are compact and efficient. For this reason they were used as the mathematical ground in this work. Nevertheless, since each method has its own particular advantages and disadvantages and their use depends on the application, they are discussed in Appendix A along with some relevant conversions among different attitude representations.

In this section, the most relevant quaternion operations are described as well as the related conversions to obtain the Euler angles from a quaternion.

### 3.2.1 Quaternions

The hypercomplex numbers of rank 4 were introduced by Sir William Hamilton in 1843 and named as quaternions [58]. A quaternion can be used to represent the orientation of a rigid body in the 3D space. The orientation of a frame B relative to a frame A can be achieved through a rotation of an angle $\theta$ around an axis $\hat{\boldsymbol{r}}$ defined in the A-frame, as represented in figure 3.3 [52].

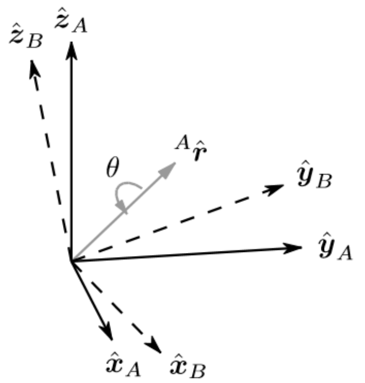


Figure 3.3: Rotation of the frame B by an angle $\theta$ around an axis $^A\hat{\boldsymbol{r}}$.

Therefore, quaternions are composed of a scalar and a vectorial component, where the former represents the magnitude of the rotation and the latter represents the axis of the rotation as described in equation 3.4.

$$^A_B\hat{\boldsymbol{q}} = \begin{bmatrix} q_s & \boldsymbol{q_v} \end{bmatrix} = \begin{bmatrix} q_s & q_x & q_y & q_z \end{bmatrix} = \begin{bmatrix} cos\left(\frac{\theta}{2}\right) & -r_x sin\left(\frac{\theta}{2}\right) & -r_y sin\left(\frac{\theta}{2}\right) & -r_z sin\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{3.4}$$

In order to describe an orientation, a quaternion should be a unit length vector. The normalization of a quaternion is achievable using equation 3.5.

$$\hat{\boldsymbol{q}} = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} = \frac{\boldsymbol{q}}{\sqrt{q_s^2 + q_x^2 + q_y^2 + q_z^2}} \tag{3.5}$$

The conjugate of the quaternion $^A_B\hat{\boldsymbol{q}}$, represented by $^B_A\hat{\boldsymbol{q}}$, describes the orientation of the A-frame relatively to the B-frame. If normalized, the quaternion conjugate and the inverse quaternion are equivalent entities, as shown in Equation 3.6.

$$^A_B\hat{\boldsymbol{q}}^* = {^A_B\hat{\boldsymbol{q}}}^{-1} = {^B_A\hat{\boldsymbol{q}}} = \begin{bmatrix} q_s & -q_x & -q_y & -q_z \end{bmatrix} \tag{3.6}$$

The quaternion product is denoted by ⊗ and is used to compute successive rotations, as described in equation 3.7.

$$ {}^{A}_{C}\hat{\boldsymbol{q}} = {}^{B}_{C}\hat{\boldsymbol{q}} \otimes {}^{A}_{B}\hat{\boldsymbol{q}} \tag{3.7}$$

The product of two quaternions, $\boldsymbol{p}$ and $\boldsymbol{q}$, is determined using the Hamilton rule. The quaternion multiplication is a non-commutative operation, i.e., $\boldsymbol{p} \otimes \boldsymbol{q} \neq \boldsymbol{q} \otimes \boldsymbol{p}$.

$$\boldsymbol{p} \otimes \boldsymbol{q} = \begin{bmatrix} p_s q_s - p_x q_x - p_y q_y - p_z q_z \\ p_s q_x + p_x q_s + p_y q_z - p_z q_y \\ p_s q_y - p_x q_z + p_y q_s + p_z q_x \\ p_s q_z + p_x q_y - p_y q_x + p_z q_s \end{bmatrix} \tag{3.8}$$

The rotation of a 3D vector is done using equation 3.9. The vector should start with a null element in order to fit the four-element quaternion size.

$$ {}^{B}\boldsymbol{v} = {}^{A}_{B}\hat{\boldsymbol{q}} \otimes {}^{A}\boldsymbol{v} \otimes {}^{A}_{B}\hat{\boldsymbol{q}}^{*} \tag{3.9}$$

### 3.2.2 Relevant conversions using quaternions

The rotation matrix that represents the orientation described by ${}^{A}_{B}\hat{\boldsymbol{q}}$ is shown in equation 3.10.

$$ {}^{A}_{B}\boldsymbol{R} = \begin{bmatrix} q_s^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y + q_s q_z) & 2(q_x q_z - q_s q_y) \\ 2(q_x q_y - q_s q_z) & q_s^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_s q_x + q_y q_z) \\ 2(q_x q_z + q_s q_y) & 2(q_y q_z - q_s q_x) & q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{3.10}$$

Therefore, Euler angles can be directly obtained from a quaternion using the equations in 3.11 to 3.13. It is considered that $\phi$ represents the rotation around the $x$ axis, $\varphi$ around the $y$ axis and $\psi$ around the $z$ axis. The function $atan2(x,y)$ represents the four-quadrant tangent inverse.

$$\phi = atan2(R_{32}, R_{33}) = atan2(\ 2(q_y q_z - q_s q_x)\ ,\ q_s^2 - q_x^2 - q_y^2 + q_z^2) \tag{3.11}$$

$$\varphi = -sin^{-1}(R_{31}) = -sin^{-1}(\ 2(q_x q_z + q_s q_y)) \tag{3.12}$$

$$\psi = atan2(R_{21}, R_{11}) = atan2(\ 2(q_x q_y - q_s q_z)\ ,\ q_s^2 + q_x^2 - q_y^2 - q_z^2) \tag{3.13}$$

## 3.3 Orientation from inertial sensors

The previous section presented the quaternions and their use in attitude representation. However, the orientation is not directly obtained from sensors measurements. This section demonstrates the procedures required to compute the orientation from gyroscopes and accelerometers using a quaternion representation.

### 3.3.1 Orientation from gyroscopes

A tri-axis gyroscope provides measurements of angular rate about the $x$, $y$ and $z$ axes of the sensor frame. This measurements, in $rad \cdot s^{-1}$ , can be arranged into a vector as shown in equation 3.14.

$$^{S}\boldsymbol{\omega} = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \end{bmatrix} \tag{3.14}$$

The orientation change rate of the sensor frame relatively to the earth frame is described by the quaternion derivative ${}^{E}_{S}\dot{\boldsymbol{q}}$. At a time step $t$, the quaternion derivative is determined using equation 3.15, where ${}^{E}_{S}\hat{\boldsymbol{q}}_{\omega,t\text{-}1}$ represents the previous estimate of orientation.

$${}^{E}_{S}\dot{\boldsymbol{q}}_{\omega,t} = \frac{1}{2}\,{}^{E}_{S}\hat{\boldsymbol{q}}_{\omega,t\text{-}1} \otimes {}^{S}\boldsymbol{\omega}_t \tag{3.15}$$

The orientation estimate is obtained by numerically integrating the quaternion derivative, provided known initial conditions. The sampling period is represented by $\Delta t$.

$${}^{E}_{S}\hat{\boldsymbol{q}}_{\omega,t} = {}^{E}_{S}\hat{\boldsymbol{q}}_{\omega,t\text{-}1} + {}^{E}_{S}\dot{\boldsymbol{q}}_{\omega,t}\ \Delta t \tag{3.16}$$

### 3.3.2 Orientation from accelerometers

A tri-axis accelerometer provides measurements of the magnitude and direction of the gravity field in the sensor frame. Linear accelerations caused by motion of the sensor are also measured by accelerometers but, in the context of an orientation estimation algorithm, they are not accounted.

The orientation of the sensor frame relative to the earth frame can be calculated using measurements from the sensors, provided that the direction of the field is known in the earth frame. However, for each measurement, there will be infinite orientation solutions [62].

A single solution can be reached through the formulation of an optimisation problem where the orientation of the sensor, ${}^{E}_{S}\hat{\boldsymbol{q}}$, is the one that aligns a predefined direction of the field in the earth frame ,${}^{E}\hat{\boldsymbol{d}}$, with the direction of the field measured by the sensor, ${}^{S}\hat{\boldsymbol{s}}$. Therefore, the goal is to solve equation 3.17 considering the objective function defined in equation 3.18.

$$\min_{{}^{E}_{S}\hat{\boldsymbol{q}} \in \mathbb{R}^4} \boldsymbol{f}\left({}^{E}_{S}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}\right) \tag{3.17}$$

$$\boldsymbol{f}\left({}^{E}_{S}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}\right) = {}^{E}_{S}\hat{\boldsymbol{q}}^{*} \otimes {}^{E}\hat{\boldsymbol{g}} \otimes {}^{E}_{S}\hat{\boldsymbol{d}} - {}^{S}\hat{\boldsymbol{s}} \tag{3.18}$$

The Gradient Descent Algorithm (GDA) is usually used as the optimisation algorithm due to its straightforward implementation and computation. It consists in a first-order iterative optimization algorithm that converges to a local minimum of a function, provided an initial condition. This algorithm is described in equation 3.19, where ${}^{E}_{S}\hat{\boldsymbol{q}}_k$ represents the orientation estimate after $k$ iterations, $\mu$ is the step-size that controls the convergence rate of the algorithm and $\nabla \boldsymbol{f}$ is the gradient of the objective function.

$${}^{E}_{S}\hat{\boldsymbol{q}}_{k+1} = {}^{E}_{S}\hat{\boldsymbol{q}}_k - \mu \frac{\nabla \boldsymbol{f}}{||\nabla \boldsymbol{f}||}, k = 0, 1, 2...n \tag{3.19}$$

The algorithm described in equation 3.19 requires multiple iterations to obtain an orientation estimate as well as an adaptive step-size $\mu$ for each iteration, which widely increases the computational needs of the algorithm. However, the computation of one iteration per time sample ensures sufficient accuracy for this application. Therefore, equation 3.19 can be replaced by equation 3.20. The gain $\mu_t$ can be assumed as a constant value and should ensure the convergence rate of ${}^{E}_{S}\boldsymbol{q}_{\nabla,t}$.

$${}^{E}_{S}\hat{\boldsymbol{q}}_{\nabla,t} = {}^{E}_{S}\hat{\boldsymbol{q}}_{est,t\text{-}1} - \mu_t \frac{\nabla \boldsymbol{f}}{||\nabla \boldsymbol{f}||} \tag{3.20}$$

The gradient is defined by the objective function and its Jacobian, as shown in equation 3.21.

$$\nabla \boldsymbol{f}\left({}^{E}_{S}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}\right) = \boldsymbol{J}^{T}\left({}^{E}_{S}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{d}}\right) \boldsymbol{f}\left({}^{E}_{S}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}\right) \tag{3.21}$$

The general form of the algorithm, described in equations 3.19 and 3.21, is applicable to a predefined field in any direction. However, when concerning to orientation estimation from accelerometers, it can be assumed that the direction of the field only have a component along the z-axis, which simplifies the computation of the gradient. Nevertheless, the computation of the gradient for a general field can be found in [52].

The computation of the gradient can be done using the matrices presented in 3.24 and 3.25. The direction of gravity and the accelerometers measurements are defined by ${}^{E}\hat{\boldsymbol{g}}$ and ${}^{S}\hat{\boldsymbol{a}}$ , respectively.

$$^{E}\hat{\boldsymbol{g}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

$$^{S}\hat{\boldsymbol{a}} = \begin{bmatrix} 0 & a_x & a_y & a_z \end{bmatrix} \tag{3.23}$$

$$\boldsymbol{f}_a\left({}^{E}_{S}\hat{\boldsymbol{q}}, {}^{S}\hat{\boldsymbol{a}}\right) = \begin{bmatrix} 2(q_x q_z - q_s q_y) - a_x \\ 2(q_s q_x + q_y q_z) - a_y \\ 1 - 2q_x^2 - 2q_y^2 - a_z \end{bmatrix} \tag{3.24}$$

$$\boldsymbol{J}_a\left({}^{E}_{S}\hat{\boldsymbol{q}}\right) = \begin{bmatrix} -2q_y & 2q_z & -2q_s & 2q_x \\ 2q_x & 2q_s & 2q_z & 2q_y \\ 0 & -4q_x & -4q_y & 0 \end{bmatrix} \tag{3.25}$$

## 3.4 Sensor fusion filters

The previous section demonstrated the procedures required to compute the orientation from gyroscopes and accelerometers using a quaternion representation. Gyroscope measurements are accurate and they are not affected by external forces. However, due to the integration of their data the measurements tend to drift over time, and for this reason they are only reliable on the short term. On the other hand, accelerometers are more reliable on the long term since they measure linear accelerations in addiction to the gravity vector.

Sensor fusion algorithms are required to combine the measurements of both sensors, merging the best attributes of each sensor while mitigating some errors introduced in the orientation estimates. The basic CF approach is illustrated in Figure 3.4.

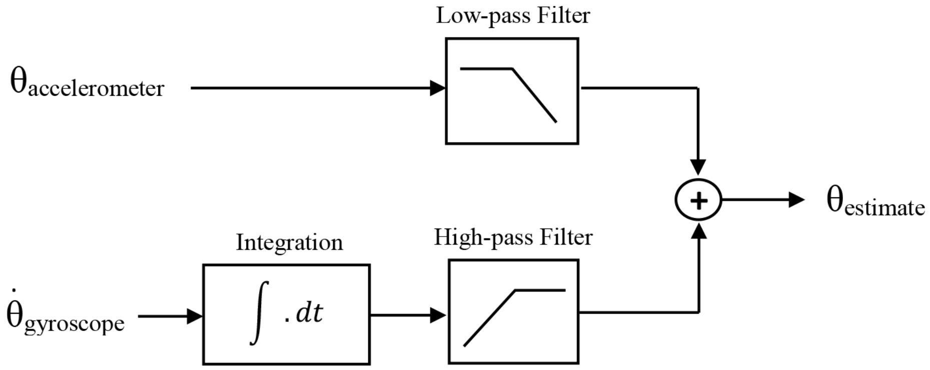


Figure 3.4: The basic IMU complementary filter.

### 3.4.1 Madgwick filter

In 2010, Sebastian Madgwick [52] presented a computationally efficient algorithm to estimate the orientation of IMUs using a quaternion representation.

The estimated orientation ${}^E_S\hat{\boldsymbol{q}}_{est,t}$ is obtained by merging the orientation from gyroscopes ${}^E_S\hat{\boldsymbol{q}}_{\omega,t}$ and accelerometers ${}^E_S\hat{\boldsymbol{q}}_{\nabla,t}$, which are calculated using equations 3.16 and 3.20, respectively. This fusion is represented in equation 3.26.

$$
{}^E_S\hat{\boldsymbol{q}}_{est,t} = \gamma_t\,{}^E_S\hat{\boldsymbol{q}}_{\nabla,t} + (1-\gamma_t)\,{}^E_S\hat{\boldsymbol{q}}_{\omega,t},\; 0 \leq \gamma_t \leq 1 \tag{3.26}
$$

The optimal value of $\gamma_t$ should ensure that the divergence rate of ${}^E_S\hat{\boldsymbol{q}}_{\omega,t}$ and the convergence rate of ${}^E_S\hat{\boldsymbol{q}}_{\nabla,t}$ have the same weight. This relation is expressed by equation 3.27, where $\beta$ represents the divergence rate of ${}^E_S\hat{\boldsymbol{q}}_{\omega,t}$ and $\frac{\mu_t}{\Delta t}$ represents the convergence rate of ${}^E_S\hat{\boldsymbol{q}}_{\nabla,t}$.

$$
(1-\gamma_t)\beta = \gamma_t\frac{\mu_t}{\Delta t} \tag{3.27}
$$

Since $\mu_t$ controls the convergence rate of ${}^E_S\hat{\boldsymbol{q}}_{\nabla,t}$, it can be assumed as a large value, provided that it is equal or greater than the physical rate of change of orientation. Hence, the term ${}^E_S\hat{\boldsymbol{q}}_{est,t-1}$ of equation 3.20 becomes negligible, and thus equation 3.20 can be re-written as equation 3.28.

$$
{}^E_S\hat{\boldsymbol{q}}_{\nabla,t} \approx -\mu_t\frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|} \tag{3.28}
$$

Furthermore, if equation 3.27 is rearranged in order to define $\gamma_t$, the equation is also simplified, as shown in equation 3.29.

$$
\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t}+\beta} \approx \frac{\beta\Delta t}{\mu_t} \approx 0 \tag{3.29}
$$

At this point, equations 3.15, 3.28 and 3.29 can be merged with equation 3.26. Hence, the estimated orientation is defined by equation 3.30 which can be simplified resulting in equation 3.31.

$$ {}^E_S\hat{\boldsymbol{q}}_{est,t} = \frac{\beta \Delta t}{\mu_t} \cdot \left( -\mu_t \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|} \right) + (1-0)\left( {}^E_S\hat{\boldsymbol{q}}_{est,t\text{-}1} + {}^E_S\dot{\boldsymbol{q}}_{\omega,t}\, \Delta t \right) \tag{3.30} $$

$$ {}^E_S\hat{\boldsymbol{q}}_{est,t} = {}^E_S\hat{\boldsymbol{q}}_{est,t\text{-}1} + \left( {}^E_S\dot{\boldsymbol{q}}_{\omega,t} - \beta \cdot \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|} \right) \Delta t \tag{3.31} $$

Equation 3.31 demonstrates that the orientation estimate ${}^E_S\hat{\boldsymbol{q}}_{est,t}$ is obtained by numerically integrating the orientation change rate ${}^E_S\dot{\boldsymbol{q}}_{est,t}$ , defined by $\left( {}^E_S\dot{\boldsymbol{q}}_{\omega,t} - \beta \cdot \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|} \right)$. The component $\frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|}$ represents the direction of the error of ${}^E_S\dot{\boldsymbol{q}}_{est,t}$. Figure 3.5 illustrates the block diagram of the Madgwick filter.

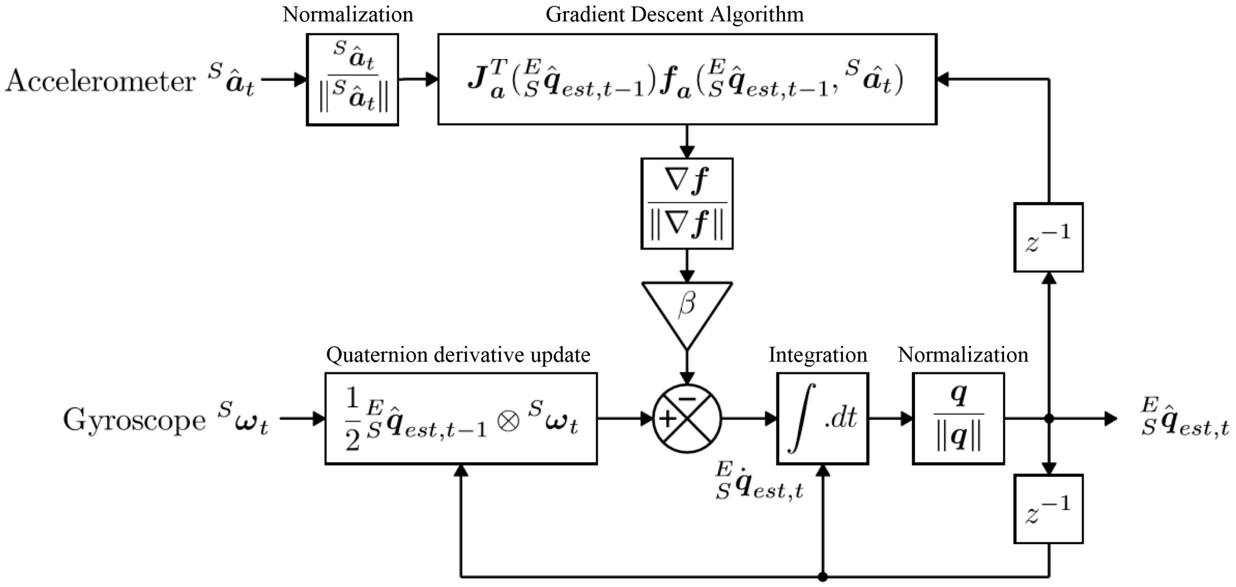

Normalization
Gradient Descent Algorithm
Accelerometer ${}^S\hat{\boldsymbol{a}}_t$
$\frac{{}^S\hat{\boldsymbol{a}}_t}{\|{}^S\hat{\boldsymbol{a}}_t\|}$
$\boldsymbol{J}^T_{\boldsymbol{a}}({}^E_S\hat{\boldsymbol{q}}_{est,t-1})\boldsymbol{f}_{\boldsymbol{a}}({}^E_S\hat{\boldsymbol{q}}_{est,t-1}, {}^S\hat{\boldsymbol{a}}_t)$
$\frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|}$
$\beta$
$z^{-1}$
Quaternion derivative update
Integration
Normalization
Gyroscope ${}^S\boldsymbol{\omega}_t$
$\frac{1}{2}{}^E_S\hat{\boldsymbol{q}}_{est,t-1} \otimes {}^S\boldsymbol{\omega}_t$
$+$
$-$
$\int .dt$
$\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|}$
${}^E_S\hat{\boldsymbol{q}}_{est,t}$
${}^E_S\dot{\boldsymbol{q}}_{est,t}$
$z^{-1}$

Figure 3.5: Block diagram of the Madgwick filter for an IMU implementation.

The filter gain $\beta$ represents the magnitude of the gyroscope measurement error. This gain can be computed using angular quantites as represented in equation 3.32. The constant $\tilde{\omega}_\beta$ corresponds to the estimated mean zero gyroscope measurement error of each axis.

$$ \beta = \left\| \frac{1}{2}\hat{\boldsymbol{q}} \otimes \begin{bmatrix} 0 & \tilde{\omega}_\beta & \tilde{\omega}_\beta & \tilde{\omega}_\beta \end{bmatrix} \right\| = \sqrt{\frac{3}{4}}\tilde{\omega}_\beta \tag{3.32} $$

### 3.4.2 Mahony filter

Robert Mahony *et al* [51, 63] also developed a complementary filter approach to estimate the orientation of IMUs. Even though the original version of this filter was described using rotation matrices, a quaternion representation of this filter is described in this work.

Contrary to the Madgwick filter which uses a GDA approach to obtain the orientation change rate error, the Mahony filter uses a Proportional-Integral (PI) gain approach. Therefore, the angular rate error ${}^{S}\boldsymbol{\omega}_{\varepsilon,t}$ can be obtained through the cross product of the measured orientation and the predicted one, as shown in equation 3.33. Vector ${}^{S}\hat{\boldsymbol{g}}_t$ represents the estimated gravity vector extracted from the quaternion estimate.

$$ {}^{S}\boldsymbol{\omega}_{\varepsilon,t} = {}^{S}\hat{\boldsymbol{a}}_t \times {}^{S}\hat{\boldsymbol{g}}_t \tag{3.33}$$

The corrected angular rate is obtained through the sum of the gyroscope measurements and the PI gain of the angular rate error, as described in equation 3.34.

$$ {}^{S}\boldsymbol{\omega}'_t = {}^{S}\boldsymbol{\omega}_t + K_p \cdot {}^{S}\boldsymbol{\omega}_{\varepsilon,t} + K_i \cdot \int {}^{S}\boldsymbol{\omega}_{\varepsilon,t} \tag{3.34}$$

The estimated orientation change rate, ${}^{E}_{S}\dot{\boldsymbol{q}}_{est,t}$, is determined using equation 3.35 which is similar to equation 3.15, though in this case the corrected angular rates are considered instead of the gyroscope angular rate measurements.

$$ {}^{E}_{S}\dot{\boldsymbol{q}}_{est,t} = \frac{1}{2}\, {}^{E}_{S}\hat{\boldsymbol{q}}_{est,t\text{-}1} \otimes {}^{S}\boldsymbol{\omega}'_t \tag{3.35}$$

Finally, the orientation estimate is obtained by numerically integrating the quaternion derivative, provided that initial conditions are known.

$$ {}^{E}_{S}\hat{\boldsymbol{q}}_{est,t} = {}^{E}_{S}\hat{\boldsymbol{q}}_{est,t\text{-}1} + {}^{E}_{S}\dot{\boldsymbol{q}}_{est,t}\Delta t \tag{3.36}$$

The block diagram of the Mahony filter is illustrated in Figure 3.6.

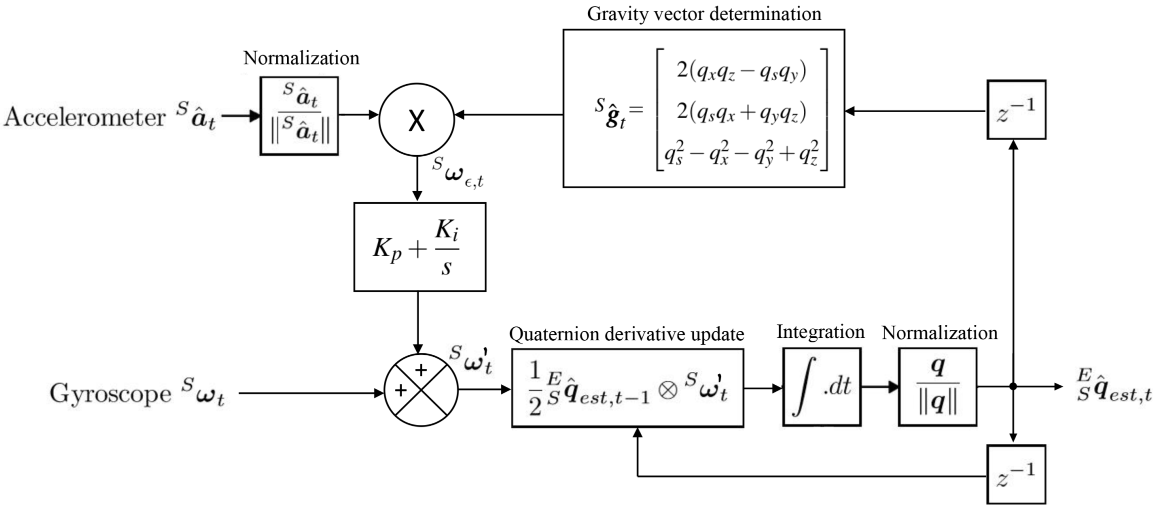

Gravity vector determination
Normalization
Accelerometer $^S\hat{\boldsymbol{a}}_t$
$\frac{^S\hat{\boldsymbol{a}}_t}{\|^S\hat{\boldsymbol{a}}_t\|}$
X
$^S\hat{\boldsymbol{g}}_t = \begin{bmatrix} 2(q_xq_z - q_sq_y) \\ 2(q_sq_x + q_yq_z) \\ q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}$
$z^{-1}$
$^S\boldsymbol{\omega}_{\epsilon,t}$
$K_p + \frac{K_i}{s}$
Quaternion derivative update
Integration
Normalization
$^S\boldsymbol{\omega}'_t$
Gyroscope $^S\boldsymbol{\omega}_t$
+
+
$\frac{1}{2}{}^E_S\hat{\boldsymbol{q}}_{est,t-1} \otimes {}^S\boldsymbol{\omega}'_t$
$\int .dt$
$\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|}$
${}^E_S\hat{\boldsymbol{q}}_{est,t}$
$z^{-1}$

Figure 3.6: Block diagram of the Mahony filter for an IMU implementation.

This filter has two gains corresponding to the proportional and integral gains, represented by $K_p$ and $K_i$, respectively. Similarly to the gain $\beta$ in the Madgwick filter, $K_p$ gain controls the weight of the orientation error which is calculated through the accelerometer measurements. The $K_i$ gain can be used to correct the gyroscope offset drift.

Both of these complementary filter approaches are suitable for low-power embedded systems due to their low computational requirements. Madgwick report [52] also includes an IMU filter implementation[4] optimised for C environment.

[4] Open source C code as well as some examples can be found in `http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/`.

# Chapter 4

# Inertial navigation in the industrial environment

The filters described in the last chapter represent generic approaches to estimate the orientation of a rigid body using an IMU. These filters accumulate various errors over time which reduce the accuracy of the orientation estimate, and therefore the efficiency of the navigation algorithm. Thus, some improvements can be made to these filters in order to adapt them to the operation environment.

The first section of this chapter introduces the industrial environment present in this work as well as the prototype of the IGCS developed. The second section presents a method to detect the motion of the vehicle and its implementation in the IGCS. Finally, the third section describes a simple method to allow the navigation of the vehicle along straight lines based on inertial data and also illustrates the implementation of the complete algorithm in the hardware.

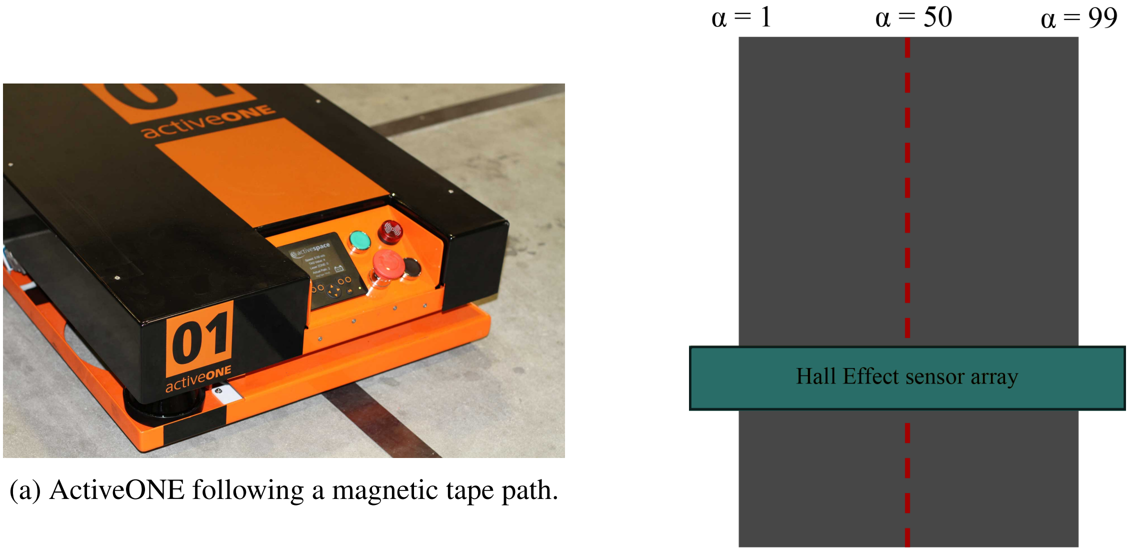
## 4.1 Industrial environment and inertial sensor prototype

As previously referred, the work of this dissertation involves redesigning an existing hardware in order to improve its capabilities, namely, the inclusion of inertial sensors to accomplish a orientation-based navigation of the vehicle in specific sections of a predefined path. Therefore, this section introduces the AGV used in this work as well as the method used for its navigation. Furthermore, the IGCS prototype is also detailed in this section. The choice of the IMU for this work was based on the results provided in [53].

### 4.1.1 Industrial Automated Guided Vehicle: ActiveONE

ActiveONE [1] is the AGV developed by *Active Space Technologies* company and used in the development of this work. It consists in a magnetic guided bi-directional AGV, capable of rotating around its own axis and with a RFID based positioning. This vehicle is mainly used in industry to transport heavy materials between different locations, for instance, from a warehouse to a production line.

The navigation of ActiveONE is similar to the navigation method illustrated in Figure 4.1b. An array of Hall Effect sensors mounted underneath the vehicle is used to detect the magnetic tape. Then, the microcontroller compute these sensors outputs to map the position of the MGS with respect to the tape, resulting in a value between 1 and 99, represented by $\alpha$ in Figure 4.1b. Posteriorly, a Programmable Logic Controller (PLC) reads this value and manipulates the speed of the vehicle's wheels in order to maintain the vehicle aligned with the center of the tape. The value of $\alpha$ can also be -1 when the sensors is not detecting the magnetic tape.



(a) ActiveONE following a magnetic tape path.

(b) Position of the sensor with respect to the tape.

Figure 4.1: ActiveONE in operational mode.

The IGCS prototype should be able to reproduce the same behaviour. The existent MGS composed of Hall Effect sensors along with the communication procedures between the hardware and the vehicle require a lot of the processing capacity and memory of the microcontroller. Therefore, the resources available to implement the IGCS are very limited. The orientation of the IGCS can be used to estimate the location of the sensor with respect to the tape. For a straight path in the ab-

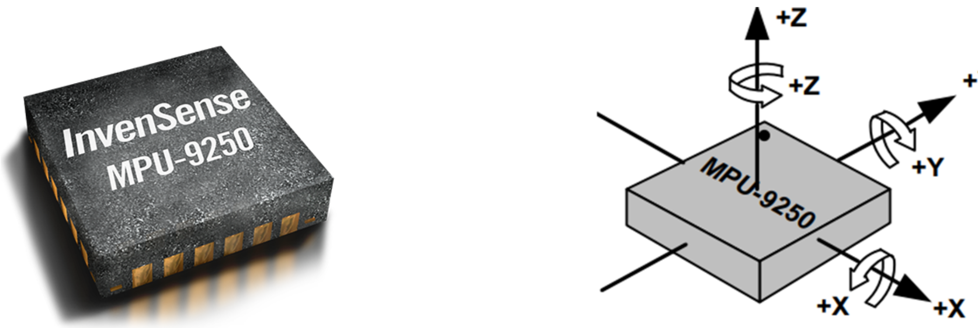
[1] For more details visit `http://www.activespaceautomation.com/agv/activeone/`.

sence of magnetic tape, the expected orientation should be around zero degrees which correspond to $\alpha = 50$. The algorithm to compute $\alpha$ from the orientation estimate is described later in this section.

### 4.1.2 Inertial Measurement Unit: MPU-9250

The MPU-9250 [2] is a 9-axis MEMS sensor designed by *InvenSense*, composed of gyroscopes, accelerometers and compasses. Since compasses are not suitable for this work due to magnetic tape disturbances, they are not referred in this dissertation. *InvenSense* guarantees that this device has a good noise performance combined with a low power-consumption and a small size. MPU-9250 is designed to be used in smartphones, tablets and wearable sensors as well as other motion-related applications.

The orientation of the axes of the 3-axis accelerometer and 3-axis gyroscope is represented in Figure 4.2b. According to this representation, the $z$ axis represents the yaw angle rotation which is the angle of interest in the navigation of these AGVs. The rotations around the $x$ axis and $y$ axis are represented by pitch and roll, respectively.



(a) MPU-9250 in a 3x3x1mm QFN package.

(b) Orientation of the axes for accelerometer and gyroscope.

Figure 4.2: MPU-9250 overview.

The MPU-9250 also features Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI) serial interfaces to allow a faster communication with the registers and the microcontroller. In this work, the redesign of the microcontroller ports is firstly required, since both I2C and SPI ports are not available to use. Since the I2C ports are occupied by an Universal Asynchronous Receiver-Transmitter (UART) port, used for debugging, they can be reassigned to another port easily. Therefore, I2C is the selected interface in the IGCS prototype.

[2] More information and the respective datasheet can be found in `https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/`.

### 4.1.3 Inertial Guidance Complementary System prototype

The hardware developed in this work consisted in the redesign of the existing PCB in order to include an IMU while simultaneously ensuring the normal execution of the Hall Effect sensors array. This task was achieved using the PCB design software *Altium*.

The selected sensor MPU-9250 communicates via I2C with a dsPIC33EV64GM104 16-bit microcontroller from *Microchip*[3], which has 64 KB Flash Memory and 8 KB of RAM. Unfortunately, the maximum supply voltage supported by the MPU-9250 is 3.6V, which is a much smaller value than the 5V operating voltage in the microcontroller. Hence, a voltage regulator is required to provide a regular 2.5V voltage from 5V. Furthermore, the I2C lines require a bi-directional level shifter to resolve this voltage incompatibility.

Summarizing, the inertial module consists of a voltage regulator, a bi-directional level shifter and a MPU-9250 sensor. The top-level hardware schematic of the inertial sensor is shown in Figure 4.3.

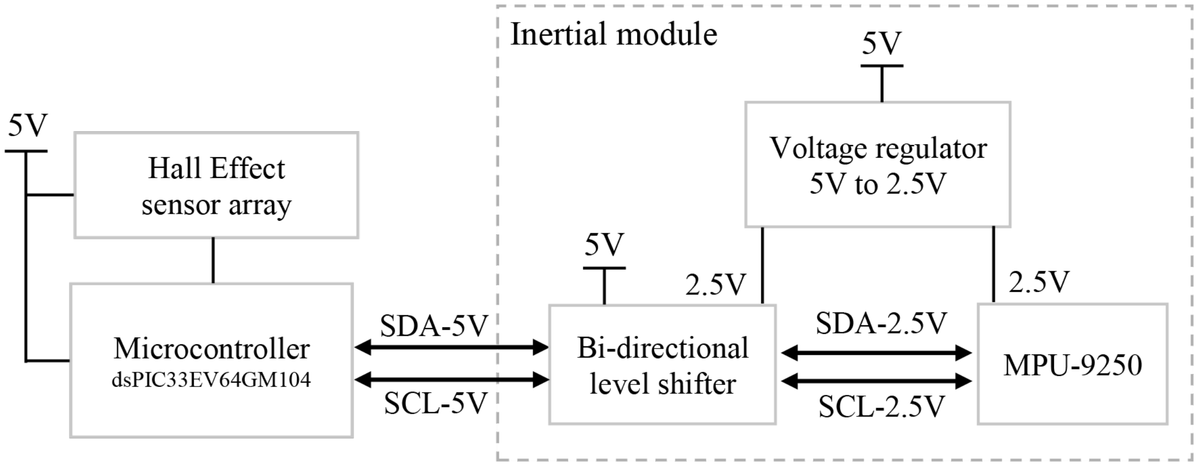


Figure 4.3: Top-level hardware schematic of the inertial sensor developed.

The I2C[4] interface is composed of two bi-directional lines: Serial Clock (SCL) and Serial Data (SDA). In the I2C bus, the devices are defined as masters or slaves. The master device is responsible for generating the clock signal and initiating the communication with slaves. The slave, which has a 7-bit unique address on the I2C bus, responds to the master when addressed.

[3] More information and the respective datasheet can be found in `http://www.microchip.com/wwwproducts/en/dsPIC33EV64GM104/`.

[4] More information about this serial interface can be found in the *Microchip* documentation `http://ww1.microchip.com/downloads/en/devicedoc/70000195f.pdf`.

## 4.2 Adaptations to the low-complexity fusion filters

The filters discussed in the previous chapter can be applied to the estimation of the orientation of a rigid body. However, they should be adapted to their specific working environment in order to achieve better results.

In this work, the main obstacle is the gyroscope drift over time produced by the accumulation of the white noise when integrating. This drift is usually mitigated using an external reference, i.e., another sensor or set of sensors that complement the IMU signals, for instance, a magnetometer or GPS. Nevertheless, the effect of the drift can be minimized by estimating the orientation only when the system is in motion as described in this section.

### 4.2.1 Motion detection

The motion of the IGCS can be computed using the data obtained from accelerometers and gyroscopes. Since these sensors detect linear and angular movements, respectively, the IGCS can be considered at a rest position (or at a constant linear speed) if the measurements returned correspond to very small values or lower than a established threshold.

The magnitude of the measurements is calculated using equations 4.1 and 4.2. The function $abs()$ represents the absolute value of a number.

$$Mag_a = abs\left(\sqrt{a_x^2 + a_y^2 + a_z^2} - 1\right) \tag{4.1}$$

$$Mag_\omega = abs\left(\sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}\right) \tag{4.2}$$

The calculated magnitudes are then submitted through a low-pass filter in order to attenuate the existing high frequencies. The discrete-time low-pass filter can be implemented through an exponentially weighted moving average as shown in equation 4.3. This straightforward method is compact and efficient for implementing in a microcontroller since only one past value needs to be maintained in every iteration. The initial conditions are considered null.

$$x_k = \lambda x_{k-1} + (1-\lambda)x_k \tag{4.3}$$

The coefficient $\lambda$ represent the smoothing factor which is restricted to the interval $0 \leq \lambda \leq 1$ and can be calculated using equation 4.4. $\Delta t$ corresponds to the sampling period and $W_n$ is the cutoff frequency parameter which should be previously tuned.

$$\lambda = \frac{\Delta t}{W_n + \Delta t} \tag{4.4}$$

Therefore, at a time step $t$, the low-pass filtered accelerometer and gyroscope magnitudes can be computed using equations 4.5 and 4.6, respectively.

$$A_t = A_{t-1} + \lambda(Mag_a - A_{t-1}) \tag{4.5}$$

$$\Omega_t = X_{t-1} + \lambda(Mag_\omega - X_{t-1}) \tag{4.6}$$

Stationary periods of the IGCS are computed using equation 4.7, where $\zeta_a$ and $\zeta_\omega$ are the thresholds of the accelerometer and gyroscope magnitudes, respectively.

$$stationary = \begin{cases} 1, & \text{if } A_t < \zeta_a \text{ and } \Omega_t < \zeta_\omega \\ 0, & else \end{cases} \tag{4.7}$$

This algorithm will be named as DOSP in this work. It is a simple approach to mitigate the effects of the gyroscope drift over time since the previous quaternion estimate is maintained when the IGCS is at rest. However, the constants $\zeta_a$ and $\zeta_\omega$ need to be tuned carefully, in order to prevent losing useful information, mainly when the AGV is moving at a constant speed.

## 4.3 Inertial navigation along straight lines

The first section of this chapter described the MGS used in the navigation of theAGV. The vehicle navigates using a value $\alpha$ which represents the position of the Hall Effect sensors array with relation to the magnetic tape. In this section, it is proposed a simple algorithm to reproduce the same behaviour using the IGCS output, i.e., the orientation around the perpendicular axis. Furthermore, a schematic of the full operational mode of the IGCS is also illustrated in this section.

### 4.3.1 Guidance algorithm

The goal of the algorithm is to find a function capable of computing $\alpha$ for a given orientation value, $\varepsilon$. The value of $\alpha$ should be equal to 50 when the orientation of the IGCS is null, $\varepsilon = 0$, which indicates that it is perfectly aligned with the magnetic tape. In this approach, only the orientation value is considered, and therefore any lateral errors are ignored. It is also considered that the vehicle is perfectly aligned with the magnetic tape in the initial position. The straightforward function to implement is the linear function which has the form $y = mx + b$, where $x$ is the independent variable, $y$ is the dependent variable, $b$ is the $y$ intercept and $m$ is the slope that gives the rate of change of the dependent variable.

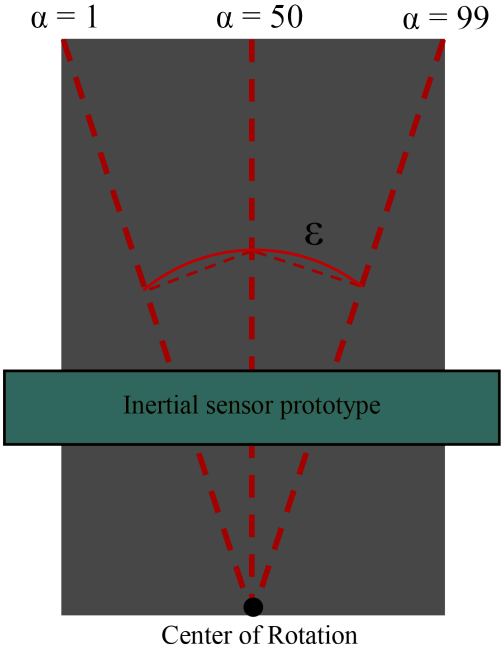


Figure 4.4: Guidance algorithm of the inertial sensor prototype.

Since $\alpha$ is a value confined to the interval of 1 to 99, the slope of the linear function can be calculated using equation 4.8.

$$m = slope = \frac{99-1}{\beta-(-\beta)} = \frac{98}{2\beta} \tag{4.8}$$

Considering an imaginary tape emulating the magnetic tape, the constant $\beta$ can be assumed as the minimum or maximum angle value which delineates the position of the IGCS on the tape. In other words, if the sensor has as orientation of $\beta$ or $-\beta$ degrees it is at the edge of the tape. The determination of this constant is done by reading the orientation of the IGCS while rotating the MGS in the presence of magnetic tape, provided that both systems are aligned in the same axis. In this work, the value of $\beta$ is assumed to be 20 degrees.

Finally, the position of the sensor with respect to the tape, $\alpha$, can be computed accordingly to equation 4.9.

$$\alpha = \begin{cases} 1, & \text{if } \varepsilon < -\beta \\ 99, & \text{if } \varepsilon > \beta \\ m \cdot \varepsilon + 50, & else \end{cases} \tag{4.9}$$

### 4.3.2 Firmware implementation

The microcontroller was programmed using the software program *MPLAB X IDE* from *Microchip*. The use of the IGCS for navigation can be selected through a command sent by the AGV. Thus, inertial guidance can be selected for some desired straight paths while maintaining the normal execution of the AGV in the rest of the circuit.

The flowchart of the firmware implementation in the IGCS is illustrated in Figure 4.5. This diagram does not concern to Hall Effect sensors data acquisition as well as other related firmware since it is not relevant for the purpose of this work. It is assumed that the Hall Effect sensors are previously calibrated and the desired circuit is already loaded in the AGV.

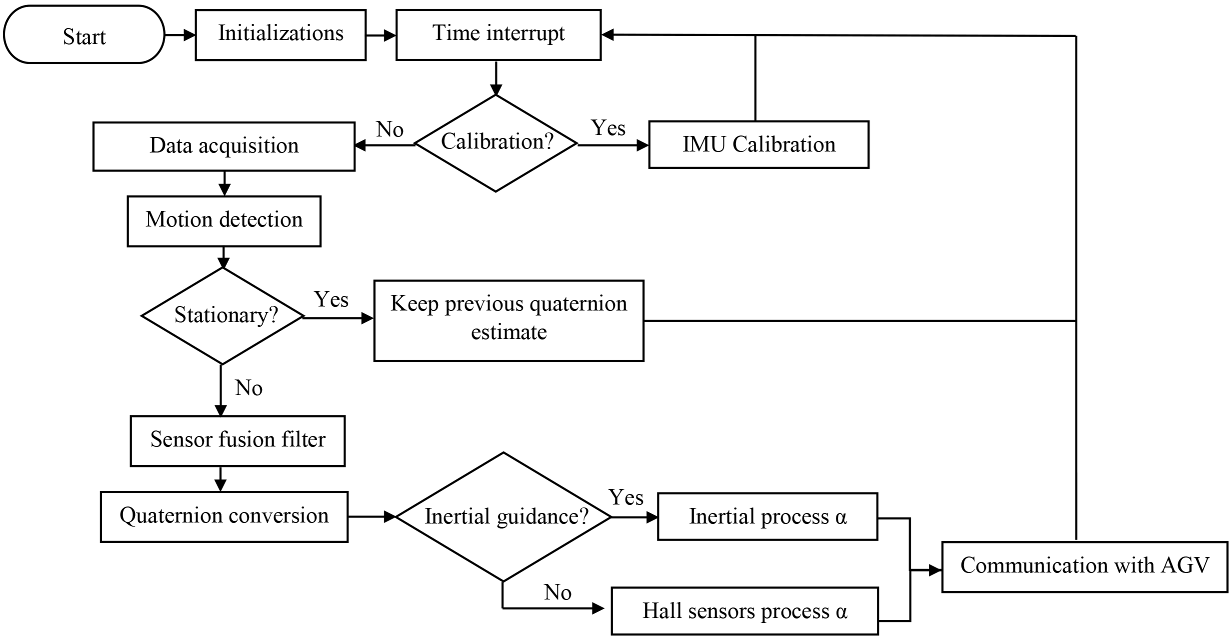


Figure 4.5: Flowchart of the firmware implementation.

The vehicle takes around 3 to 5 seconds to initiate after being turned on. However, the acquisition of measurements starts immediately after it is turned on. The entire process is composed by various steps which will be detailed below.

### Initializations

In this block of the diagram the Microcontroller and the sensors are initialised. This consists, for instance, in the definition of the gyroscope and accelerometer scales and the selection of the clock source, among many other initial settings. Furthermore, the configuration of timers, Analog-to-Digital Converters (ADCs) or UART ports are also included in this block.

### Time interrupt

In order to signal the microcontroller that an event needs immediate attention, an interrupt should be used. In this work, a temporal interrupt was already implemented in the MGS to obtain the measurements from the Hall Effect sensors at a sample rate of 100 Hz, and therefore it is also used for inertial sensors data acquisition.

### Calibration

The calibration of the inertial sensors is required in order to obtain reasonable measurements. The simple calibration algorithm implemented consists in calculating the mean of 100 measurements in order to remove the constant bias of these sensors, using equation 3.1 described previously. Because accelerometers measure the gravity as a constant bias, it has to be removed before this calculation. Therefore, the direction of the gravity must be known. A straightforward approach is to consider that the calibration is done with the sensor on a rest position and aligned with the ground, which is plausible in AGVs. The trigger of the calibration is done by the user or automatically, after a pre defined number of iterations.

### Data acquisition

The data acquisition block consists in the procedures developed to read the data from the registers and its manipulation to obtain the final data in their respective units of measurement. Most of the data acquisition information is explained in the register map of the MPU-9250 sensor[5]. In every iteration, the calibrated data is obtained by subtracting the calibration values from the data measured, i.e., $CalibratedData = MeasurementData - Calibration$. The calibrations values are null until the calibration procedure is performed. Any factor scale constants are also included in this step.

[5] The register map for MPU-9250 can be found in `https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/`.

#### Motion detection

In this block, the DOSP algorithm is implemented. It returns a boolean variable where the true value means that the previous quaternion estimate is maintained because the IGCS is stationary. On the other hand, a false output means that the IGCS is in motion and therefore a new orientation estimate is calculated.

#### Sensor fusion filter and quaternion conversion

The output of the fusion algorithm is a quaternion which represents a new estimated orientation. The quaternion estimate is calculated using the Madgwick filter and then it is converted to euler rotation angles. The angle rotated in degrees around the z-axis is calculated through equation 3.13.

#### Inertial Guidance

Finally, the guidance algorithm is computed in the inertial guidance block, accordingly to equation 4.8 and equation 4.9. This method will be used for guiding the vehicle, depending on the value of a flag, which can be changed through a command sent by the AGV. The output value of the algorithm is then sent to the controller to manipulate the speeds of the wheels.

# Chapter 5

# Tests and Results

This chapter presents the diverse tests performed to validate the sensor fusion algorithms studied as well as the results obtained using the IGCS for navigation.

As previously referred, the communications between the MPU-9250 sensor and the dsPIC microcontroller are made via I2C. The microcontroler receives the inertial measurements and processes them in order to obtain the orientation estimate of IGCS. During this procedure, the microcontroller can send the desired data through an available UART module so it can be manipulated later. In the tests performed and presented in this section the data were sampled at 100 Hz and various datasets were built through a serial terminal program. For analysis purposes, the program can save the desired data in a text file which can be posteriorly read by *MATLAB* software.

This chapter is organized in four sections. In the first section, a comparison between the studied complementary filtering approaches is shown. The second section presents the initial approach to the IGCS, which consisted in the detection of the different sections of a route during the normal navigation of the vehicle. The third section describes the results of the low-computational low-pass filter implementation as well as its use in detecting the motion of the AGV. In the fourth section, the inertial navigation results are presented along with a comparison between the MGS and the IGCS developed.

## 5.1 Complementary filters

In Section 3, two distinct complementary filtering approaches were presented. The performance of these two techniques is evaluated in this section, through a simple test that consisted in the manual rotation of the IGCS around the perpendicular axis for about 20 seconds. During this time, the measurements of the inertial sensors are collected and are later injected into the filters algorithms in order to calculate the estimated orientation of the IGCS. Unfortunately, the required hardware to obtain the ground truth signal was not available, and therefore it was not possible to evaluate the performance of the filters more accurately, for instance, by calculating the orientation error. Nevertheless, the results of this test are shown in Figure 5.1.

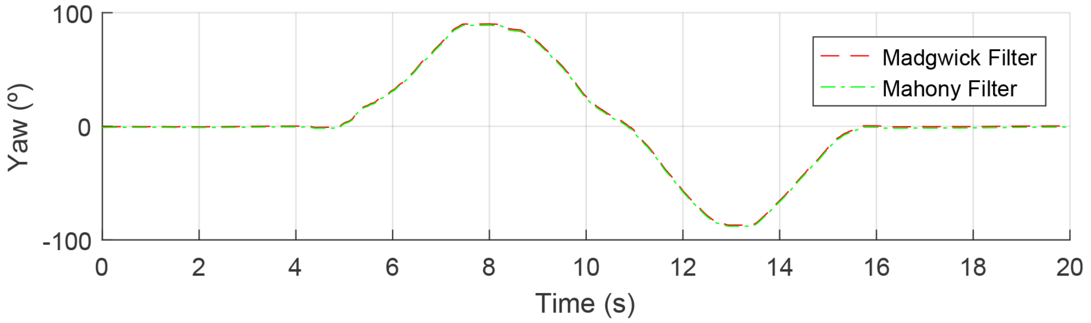


Figure 5.1: Comparison between the two complementary filtering approaches.

Since the two algorithms presented a very similar performance, any of them could be implemented in the microcontroller. In this work, the Madgwick filter was the selected approach due to time constraints, since many resources of this filter were available, namely, the filter implementation optimised in C, provided by the author. A filter gain $\beta$ of 0.1 was considered in the tests performed.

## 5.2 Detection of sections along the route

In order to obtain inertial measurements during the normal navigation of the AGV a simple test was made. The vehicle was placed in the magnetic tape circuit illustrated in Figure 5.2 and set in motion using the MGS for navigation. Through an appropriate software, the forward and backward routes of the vehicle were configured. This consists in defining the operations of the vehicle (for instance: Move, Stop or Change Speed) for each RFID tag present in the circuit. Table 5.2 describes the operations for each RFID tag of the forward route of the AGV.

Finally, the IGCS was fixed near the center of rotation of the vehicle to acquire the estimated orientation angle during the motion of the vehicle. The vehicle stopped after navigating through the entire circuit three times.

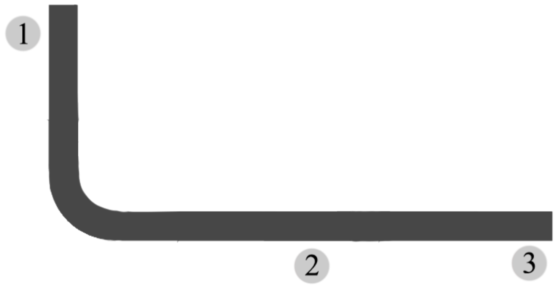


Figure 5.2: Magnetic tape circuit used for initial tests. The numbers represent the position of each RFID tag.

| Tag | State |
|---|---|
| 1 | Speed = 0% (Stop)<br>Wait 5 seconds<br>Speed = 50% (Move) |
| 2 | Speed = 0% (Stop)<br>Wait 3 seconds<br>Speed = 50% (Move) |
| 3 | Speed = 0% (Stop)<br>Wait 5 seconds<br>Go to the next route |

Table 5.1: Different states of the AGV during the forward route for each RFID tag.

The calculated orientation angle of the IGCS during this test is shown in Figure 5.3. Through the analysis of this figure, it is simple to identify the state of the vehicle over time, which matches the states represented in Table 5.2.

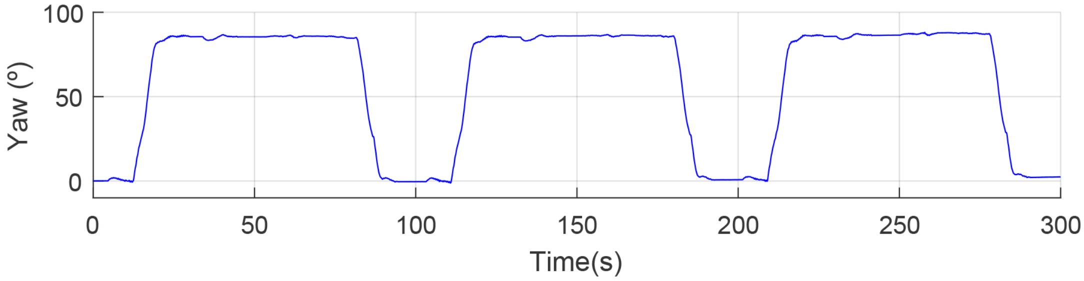


Figure 5.3: Orientation of the IGCS during the navigation of the vehicle using the MGS.

This test consisted in the motion of the vehicle using the MGS for around 300 seconds. Ideally, the calculated orientation of the IGCS would be null whenever a new lap of the circuit starts. However, due to the gyroscope drift over time, the orientation error increases over time. To measure this drift, the sensor was placed on a flat surface for several seconds and the measurements of the inertial sensors were collected to build a dataset. This results are shown in Figure 5.4.

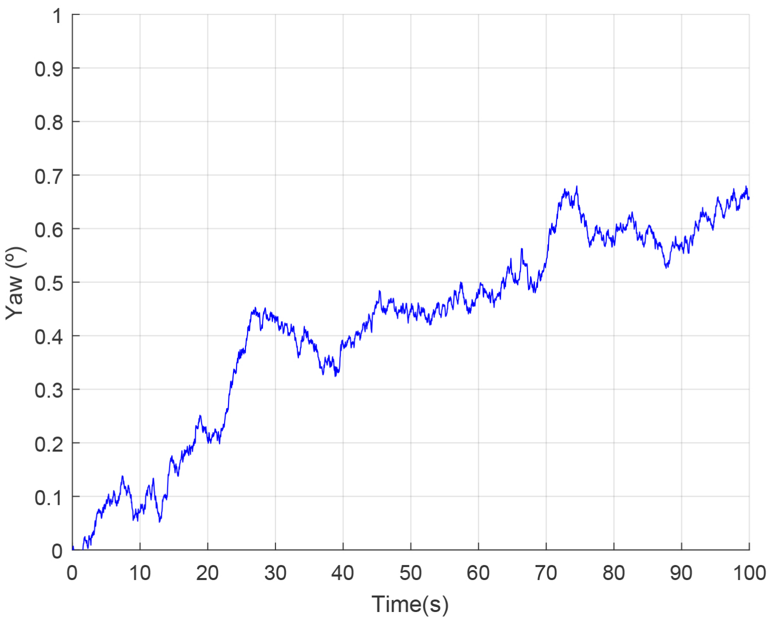


Figure 5.4: Gyroscope drift over time.

Accordingly to these results, the gyroscope drift over time can introduce an error in the orientation estimate of approximately 0.7 degrees after 100 seconds of operation. In fact, the main problem of the IMU sensor approach is the gyroscope bias drift over time around Z axis, since the accelerometer measurements only compensate X and Y axes. Nevertheless, this effect can be reduced by calibrating the inertial sensors before each desired inertial path or by computing new orientation estimates only when the IGCS is in motion, discarding the remaining measurements. This last procedure is described in the next section.

## 5.3 Motion detection

The DOSP algorithm described the use of low-computational Low-Pass Filters (LPFs) to attenuate the existing high frequencies. The implemented low-computational filter demonstrated similar results to the first-order Butterworth LPF from MATLAB library. This results are shown in Figure 5.5 for a cutoff frequency, $W_n$, of 0.1 Hz. It can be concluded that the implemented LPF has a similar performance in relation to the Butterworth LPF filter from MATLAB. Furthermore, its computational weight and easy implementation makes it suitable for implementing in the microcontroller.

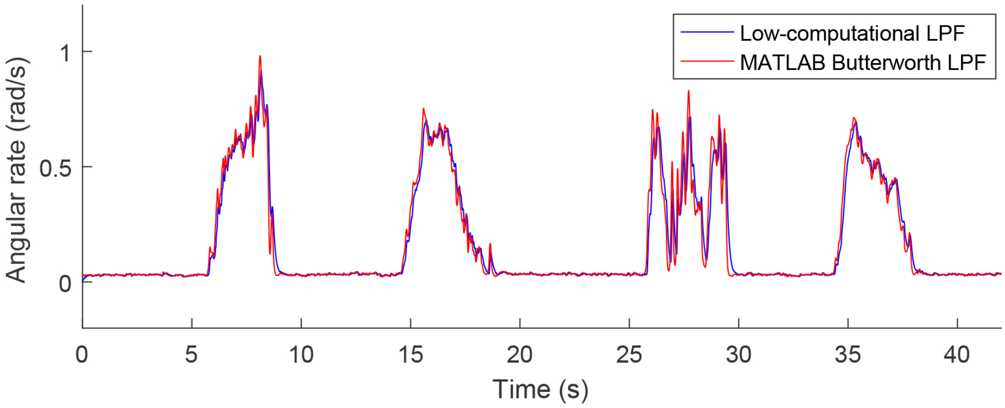


Figure 5.5: Comparison between two different low-pass filters.

The motion detection test consisted in successive rotations of the IGCS. The magnitude of the gyroscope and accelerometer measurements are exposed to the low-computational LPF implemented and then, the stationary condition is verified accordingly to Equation 4.7. Considering the thresholds $\zeta_a$ and $\zeta_\omega$ equal to 0.05, the stationary periods for the accelerometer and gyroscope are shown in Figure 5.6 and Figure 5.7, respectively.

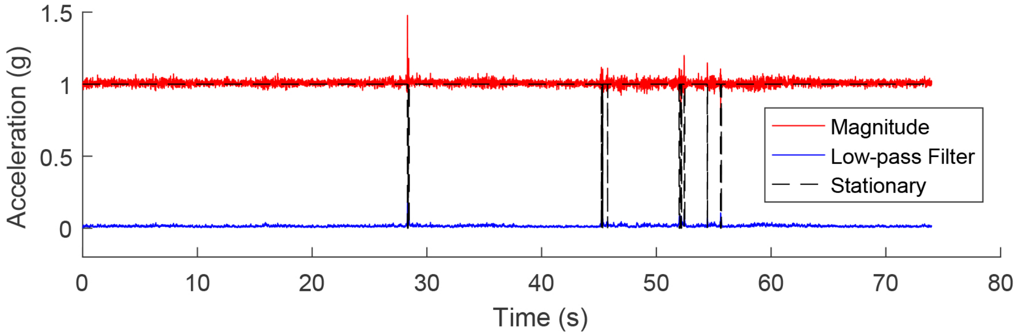


Figure 5.6: Stationary accelerometer periods during the test.

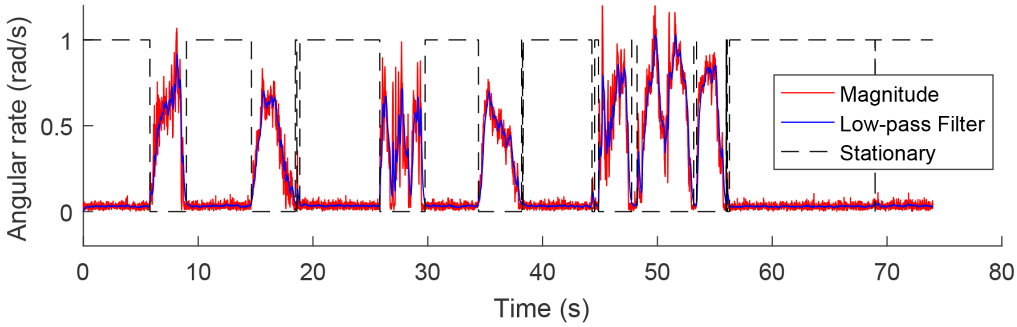


Figure 5.7: Stationary gyroscope periods during the test.

The calculation of the stationary periods of the IGCS is done through the product of both stationary signals, as shown in Figure 5.8. The IGCS is considered as stationary only when the accelerometer and gyroscope signals are both stationary, i.e. when the IGCS is not subject to any translational or angular movements. Since this test mostly consisted in angular movements, the accelerometers only measure the gravitational force, and therefore accelerometers are considered as stationary most of the time.

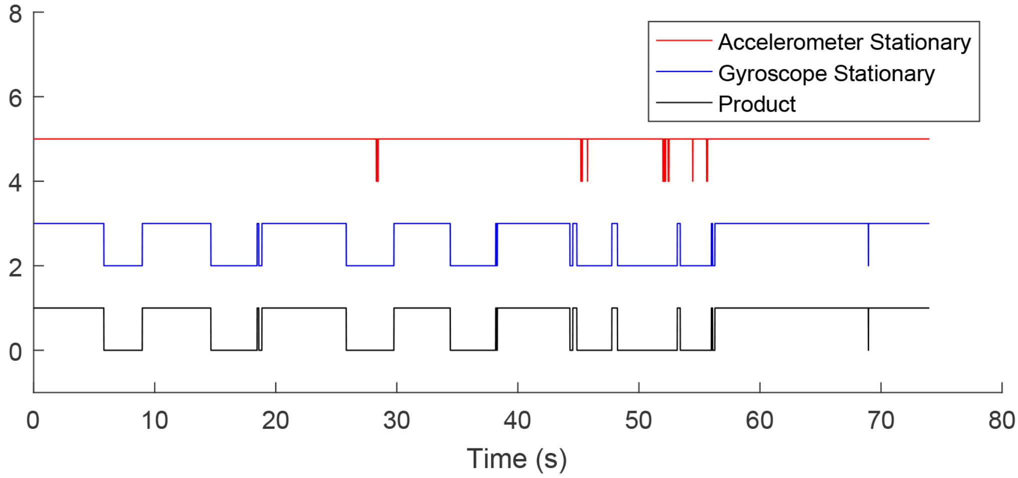


Figure 5.8: Final stationary signal during motion.

The tuning of the thresholds $\zeta_a$ and $\zeta_\omega$ must be done carefully, otherwise, if the IGCS is moving with a constant speed while not subject to angular velocities it will be considered as stationary, even though it is not. However, this approach can be assumed for these AGVs since the motion of these vehicles is easily detected by accelerometers.

The results of the DOSP method are shown in Figure 5.9. The IGCS was calibrated and then subject to 90 degrees rotations around the z-axis, returning to the initial position in the end.

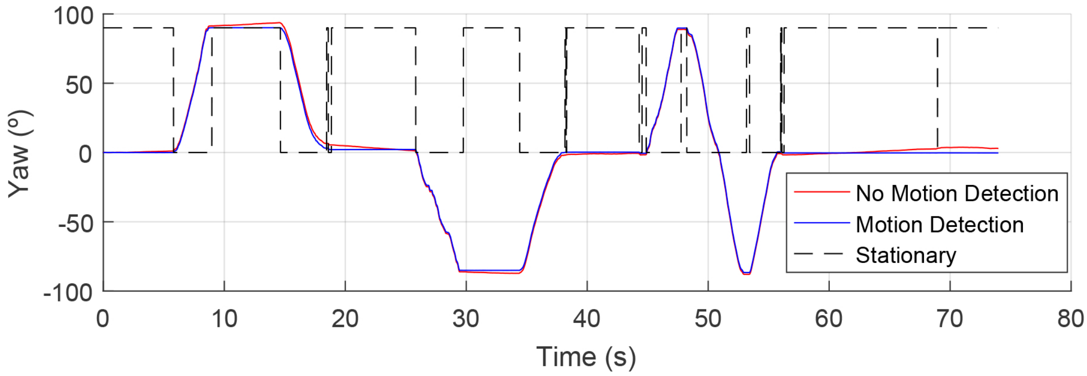


Figure 5.9: Gyroscope drift over time reduced by the DOSP method.

This method attenuates the error accumulated in the calculated orientation of the IGCS by maintaining the previous orientation estimate when the IGCS is at a stationary state. This results are particularly interesting in the navigations of these AGVs. For instance, this method prevents the accumulation of errors in the calculated orientation when the vehicle is charging the batteries, at a resting position.

The mitigation of the drift over time problem can be solved by adding a sensor to the system with the capability of compensating this drift. In other words, since the accelerometer data does not compensate the z-axis drift, another external reference would be necessary. Nevertheless, for this work, the implemented solution is sufficient for navigating along straight paths using the IGCS.

## 5.4 Inertial navigation

The goal of the inertial navigation method is to emulate the behaviour of the MGS using the IGCS developed. The tests and results for the inertial navigation, using the inertial algorithm previously described, are demonstrated in this section. These tests consisted in the movement of the vehicle at a low speed, along a straight path of approximately 20 meters .

Firstly, the vehicle was set in motion along a straight path using the MGS, in order to collect the typical $\alpha$ values during the normal execution of the AGV navigation. Then, the same procedure was executed after including the IGCS in the AGV system. Due to limited resources it was not possible to obtain both signals simultaneously, compromising the acquisition of accurate measurements.

Nevertheless, in this test, the calculated orientation of the IGCS was collected in addition to the navigation value $\alpha$, during the navigation of the vehicle using the MGS. The angle obtained through the IGCS along a straight line is shown in Figure 5.10.

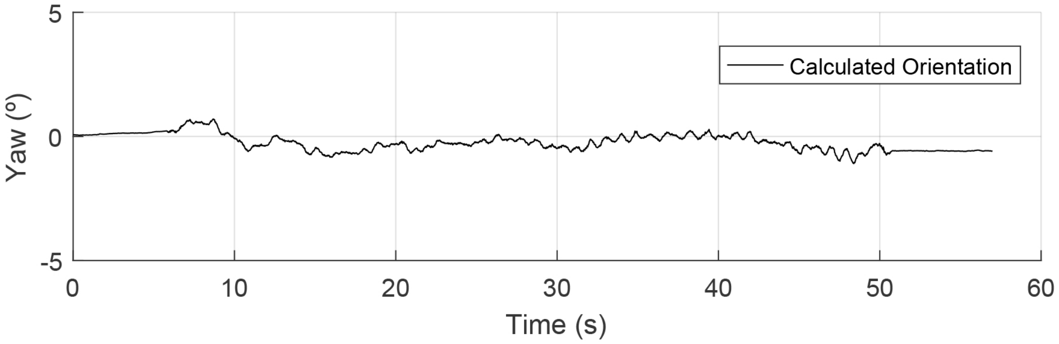


Figure 5.10: Yaw angle obtained using the IGCS.

As expected, the calculated orientation of the IGCS oscillates around zero degrees during this test, provided that the initial orientation of the IGCS is null. Similarly, the value of $\alpha$ which is used to control the speed of each wheel, should oscillate around 50, representing that the IGCS is in the center of the tape. This results are shown in Figure 5.11.

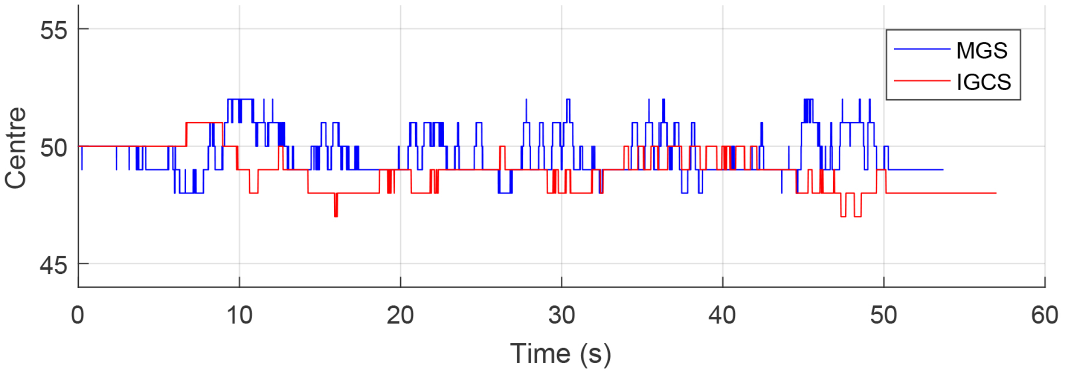


Figure 5.11: Centre value retrieved using the two approaches.

Since in this test the IGCS is calibrated before the inertial path, the initial value of the orientation angle is 0 degrees, and therefore the value of $\alpha$ is 50. This approach must consider that the vehicle starts aligned with the desired straight path. On the other hand, in magnetic tape guidance the AGV can be in a different orientation, provided that the MGS is reading the magnetic tape. The results obtained with the MGS oscillate more than the results using the IGCS since the former is based on Hall Effect sensor measurements which are influenced by small magnetic variations.

# Chapter 6

# Conclusions and Future Work

The work developed in this dissertation consisted in developing an inertial-based navigation method to complement a magnetic-guided navigation method for industrial AGVs. The resources available were very limited, mainly in the microcontroller, since the normal execution of the Hall Effect sensors had to be maintained. Furthermore, the communication between the system and the vehicle had very strict times to accomplish which also made it difficult to implement new features. Nevertheless the entire work described in this dissertation was implemented in the IGCS. More complex and accurate sensor fusion approaches, for instance Kalman approaches, have higher computational needs, and therefore they were not implemented in the IGCS. However, the accuracy of complementary filters proved to sufficient for this task. The IGCS developed consists in a relative orientation method, since the data is only obtained from gyroscopes and accelerometers. Even though the disturbances present in the measurements of these sensors can be identified, their compensation is a huge challenge, since the gyroscopes are subject to drift over time and accelerometers measure not only the gravity but also the translational accelerations. Nevertheless, considering the industrial environment characteristics and the type of data measured by these sensors, it was possible to implement a methods to reduce these disturbances during stationary periods. For a more flexible method, other sensors must be included and more rigid calibration techniques must be implemented. The drift over time of the gyroscope during the motion of the vehicle requires another external reference to be compensated, since the accelerometers only compensate two axes. This task is usually accomplished by magnetometers which were not considered in this dissertation due to the disturbances present in the environment. Therefore, the future work may consist in the study of other techniques to complement the inertial approach. Taken this into account, it will open the path for the navigating of the AGV along curved trajectories, which is a much more complicated task.

*This page is intentionally left blank.*

# Bibliography

[1] L. Schulze and A. Wullner, “The approach of automated guided vehicle systems,” in *2006 IEEE International Conference on Service Operations and Logistics, and Informatics*, June 2006, pp. 522–527.

[2] J. Sankari and R. Imtiaz, “Automated guided vehicle(agv) for industrial sector,” in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, Jan 2016, pp. 1–5.

[3] G. Ullrich, *Automated Guided Vehicle Systems - A Primer with Practical Applications*. Springer, 2015.

[4] V. M. Nivas, P. G. Krishnan, and A. C. Fredrhic, “Automated guided car (agc) for industrial automation,” in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, Feb 2016, pp. 1–6.

[5] K. Wroble, “Performance analysis of magnetic indoor local positioning system,” Master’s thesis, Western Michigan University, 2015.

[6] H. R. Park, D. J. Hyun, H. S. Yang, and H. S. Park, “A dead reckoning sensor system and a tracking algorithm for mobile robot,” in *2009 ICCAS-SICE*, Aug 2009, pp. 5559–5563.

[7] D. Hyun, H. S. Yang, H.-S. Park, and H.-J. Kim, “Dead-reckoning sensor system and tracking algorithm for 3-d pipeline mapping,” *Mechatronics*, vol. 20, no. 2, pp. 213 – 223, 2010.

[8] D. Hess, F. Kuenemund, and C. Roehrig, “Simultaneous calibration of odometry and external sensors of omnidirectional automated guided vehicles (agvs),” in *Proceedings of ISR 2016: 47st International Symposium on Robotics*, June 2016, pp. 1–8.

[9] J. Borenstein and L. Feng, “Correction of systematic odometry errors in mobile robots,” in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 3, Aug 1995, pp. 569–574 vol.3.

[10] T. Tsumura, “Agv in japan-recent trends of advanced research, development, and industrial applications,” in *Intelligent Robots and Systems ’94. ’Advanced Robotic Systems and the Real World’, IROS ’94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 3, Sep 1994, pp. 1477–1484 vol.3.

[11] I. Skog and P. Händel, “A low-cost gps aided inertial navigation system for vehicle applications,” in *2005 13th European Signal Processing Conference*, Sept 2005, pp. 1–4.

[12] T.-H. Hong, T. Chang, C. Rasmussen, and M. Shneier, “Feature detection and tracking for mobile robots using a combination of ladar and color images,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, 2002, pp. 4340–4345 vol.4.

[13] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (slam) problem,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, Jun 2001.

[14] C. Ramer, J. Sessner, M. Scholz, X. Zhang, and J. Franke, “Fusing low-cost sensor data for localization and mapping of automated guided vehicle fleets in indoor applications,” in *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sept 2015, pp. 65–70.

[15] J. Moras, V. Cherfaoui, and P. Bonnifait, “A lidar perception scheme for intelligent vehicle navigation,” in *2010 11th International Conference on Control Automation Robotics Vision*, Dec 2010, pp. 1809–1814.

[16] F. Santoso, M. A. Garratt, and S. G. Anavatti, “Visual-inertial navigation systems for aerial robotics: Sensor fusion and technology,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 260–275, Jan 2017.

[17] V. Malyavej, W. Kumkeaw, and M. Aorpimai, “Indoor robot localization by rssi/imu sensor fusion,” in *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, May 2013, pp. 1–6.

[18] J. Lategahn, M. Müller, and C. Röhrig, “Global localization of automated guided vehicles in wireless networks,” in *2012 IEEE 1st International Symposium on Wireless Systems (IDAACS-SWS)*, Sept 2012, pp. 7–12.

[19] O. Maklouf, A. Ghila, A. Abdulla, and A. Yousef, "Low cost imu  gps integration using kalman filtering for land vehicle navigation application," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 7, no. 2, pp. 184 – 190, 2013.

[20] Y. Tian, J. Zhang, and J. Tan, "Adaptive-frame-rate monocular vision and imu fusion for robust indoor positioning," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 2257–2262.

[21] O. J. Woodman, C. O. J. Woodman, and O. J. Woodman, "An introduction to inertial navigation," 2007.

[22] P. Neto, J. N. Pires, and A. P. Moreira, "3-d position estimation from inertial sensing: Minimizing the error from the process of double integration of accelerations," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, Nov 2013, pp. 4026–4031.

[23] R. Yudanto, A. P. Ompusunggu, and A. Bey-Temsamani, "On improving low-cost imu performance for online trajectory estimation," in *Smart Sensors, Actuators, and MEMS VII; and Cyber Physical Systems*, vol. 9517, 2015, pp. 95 172H–95 172H–12.

[24] R. G. Yudanto and F. Petré, "Sensor fusion for indoor navigation and tracking of automated guided vehicles," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2015, pp. 1–8.

[25] Y. Wu, "Versatile land navigation using inertial sensors and odometry: Self-calibration, in-motion alignment and positioning," in *2014 DGON Inertial Sensors and Systems (ISS)*, Sept 2014, pp. 1–19.

[26] E. Nebot and H. Durrant-Whyte, "Initial calibration and alignment of low-cost inertial navigation units for land vehicle applications," *Journal of Robotic Systems*, vol. 16, no. 2, pp. 81–92, 1999.

[27] F. L. M. J. L. Crassidis and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[28] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[29] H. W. Sorenson, "Least-squares estimation: from gauss to kalman," *IEEE Spectrum*, vol. 7, no. 7, pp. 63–68, July 1970.

[30] S. Särkkä, *Bayesian Filtering and Smoothing*. New York, NY, USA: Cambridge University Press, 2013.

[31] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006.

[32] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep., 1995.

[33] R. Negenborn, "Robot localization and kalman filters on finding your position in a noisy world," Master's thesis, UTRECHT UNIVERSITY, 2003.

[34] F. Olsson, J. Rantakokko, and J. Nygårds, "Cooperative localization using a foot-mounted inertial navigation system and ultrawideband ranging," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2014, pp. 122–131.

[35] N. Abbate, A. Basile, C. Brigante, and A. Faulisi, "Development of a mems based wearable motion capture system," in *2009 2nd Conference on Human System Interactions*, May 2009, pp. 255–259.

[36] C. M. N. Brigante, N. Abbate, A. Basile, A. C. Faulisi, and S. Sessa, "Towards miniaturization of a mems-based wearable motion capture system," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 8, pp. 3234–3241, Aug 2011.

[37] A. M. Sabatini, "Kalman-filter-based orientation determination using inertial/magnetic sensors: Observability analysis and performance evaluation," *Sensors*, vol. 11, no. 12, p. 9182–9206, Sep 2011.

[38] X. Yun, E. R. Bachmann, H. Moore, and J. Calusdian, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 2526–2533.

[39] P. S. Marinushkin and I. A. Podshivalov, "Mems-based non-orthogonal redundant inertial measurement unit for miniature navigation systems," in *2015 International Siberian Conference on Control and Communications (SIBCON)*, May 2015, pp. 1–3.

[40] E. A. Wan and R. V. D. Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[41] S. Pourtakdoust and H. G. Asl, “An adaptive unscented kalman filter for quaternion-based orientation estimation in low-cost ahrs,” *Aircraft Engineering and Aerospace Technology*, vol. 79, no. 5, pp. 485–493, 2007.

[42] P. Balzer, T. Trautmann, and O. Michler, “Epe and speed adaptive extended kalman filter for vehicle position and attitude estimation with low cost gnss and imu sensors,” in *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 01, Sept 2014, pp. 649–656.

[43] Z. Chen *et al.*, “Bayesian filtering: From kalman filters to particle filters, and beyond,” *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[44] J. J. LaViola, “A comparison of unscented and extended kalman filtering for estimating quaternion motion,” in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 3, June 2003, pp. 2435–2440 vol.3.

[45] E. Kraft, “A quaternion-based unscented kalman filter for orientation tracking,” in *Sixth International Conference of Information Fusion, 2003. Proceedings of the*, vol. 1, July 2003, pp. 47–54.

[46] W. T. Higgins, “A comparison of complementary and kalman filtering,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-11, no. 3, pp. 321–325, May 1975.

[47] R. Valenti, I. Dryanovski, and J. Xiao, “Keeping a good attitude: A quaternion-based orientation filter for imus and margs,” *Sensors*, vol. 15, no. 8, p. 19302–19330, Aug 2015.

[48] A. Cavallo, A. Cirillo, P. Cirillo, G. D. Maria, P. Falco, C. Natale, and S. Pirozzi, “Experimental comparison of sensor fusion algorithms for attitude estimation,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7585 – 7591, 2014.

[49] Y. Tian, H. Wei, and J. Tan, “An adaptive-gain complementary filter for real-time human motion tracking with marg sensors in free-living environments,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 2, pp. 254–264, March 2013.

[50] M. S. Karunarathne, S. W. Ekanayake, and P. N. Pathirana, "An adaptive complementary filter for inertial sensor based data fusion to track upper body motion," in *7th International Conference on Information and Automation for Sustainability*, Dec 2014, pp. 1–5.

[51] R. Mahony, T. Hamel, and J. M. Pflimlin, "Complementary filter design on the special orthogonal group so(3)," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec 2005, pp. 1477–1484.

[52] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Report x-io and University of Bristol (UK)*, vol. 25, 2010.

[53] M. A. C. Rasteiro, "Motion-based remote control device for interaction with multimedia content," Master's thesis, Instituto Politécnico de Leiria, 2015.

[54] D. K. Shaeffer, "Mems inertial sensors: A tutorial overview," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 100–109, April 2013.

[55] C. Hirt, S. Claessens, T. Fecher, M. Kuhn, R. Pail, and M. Rexer, "New ultrahigh-resolution picture of earth's gravity field," *Geophysical Research Letters*, vol. 40, no. 16, pp. 4279–4283, 2013. [Online]. Available: http://dx.doi.org/10.1002/grl.50838

[56] J. Wu, G. K. Fedder, and L. R. Carley, "A low-noise low-offset capacitive sensing amplifier for a 50- mu;g/ radic;hz monolithic cmos mems accelerometer," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 5, pp. 722–730, May 2004.

[57] J. A. Geen, S. J. Sherman, J. F. Chang, and S. R. Lewis, "Single-chip surface micromachined integrated gyroscope with 50 deg;/h allan deviation," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1860–1866, Dec 2002.

[58] K. Narayanan, "Performance analysis of attitude determination algorithms for low cost attitude heading reference systems," Ph.D. dissertation, Auburn University, 2010.

[59] A. M. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, July 2006.

[60] M. Zhang, J. D. Hol, L. Slot, and H. Luinge, "Second order nonlinear uncertainty modeling in strapdown integration using mems imus," in *14th International Conference on Information Fusion*, July 2011, pp. 1–7.

[61] S. D. G. G. Brás, “Deterministic position and attitude estimation methods,” Ph.D. dissertation, Universidade de Lisboa - Instituto Superior Técnico, 2015.

[62] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, “Estimation of imu and marg orientation using a gradient descent algorithm,” in *2011 IEEE International Conference on Rehabilitation Robotics*, June 2011, pp. 1–7.

[63] R. Mahony, T. Hamel, and J. M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, June 2008.

*This page is intentionally left blank.*

# Appendix A

# Attitude representations

The problem of estimating the orientation of a rigid body in the Euclidean space is formulated using two reference frames [58], namely, a non-rotating (inertial) global frame and a rotating (body-fixed) frame, usually represented by $A$ and $B$, respectively. This is demonstrated in Figure A.1 [21], where $\{x_g, y_g, z_g\}$ represent the axes of the inertial frame and $\{x_b, y_b, z_b\}$ the axes of the body frame.

As previously mentioned in this dissertation, there are several methods of representing the attitude of a rigid body. Therefore, the orientation of the body-fixed frame with respect to the inertial frame can be represented though different parameters, depending on the representation method adopted. In this appendix, the advantages and disadvantages of the different methods are discussed as well as some relevant conversions among them.

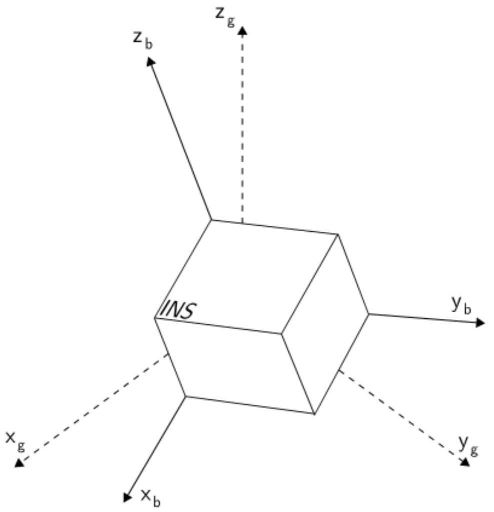


Figure A.1: The inertial and body frames of reference.

## A.1 Euler angles

The most popular method of representing the orientation of a rigid body and also the simplest and easy to visualize is the Euler method [58]. In this representation, the orientation of the B-frame with respect to the A-frame is defined by a sequence of three rotations, represented by three parameters that correspond to the angles of each rotation. Hence, there are several conventions [1] depending on the axes about which the rotations are performed and their order.

For instance, in the aeronautical convention, these angles are designated as Roll-Pitch-Yaw (RPY) angles, where roll ($\phi$) is around the $x$ axis, pitch ($\theta$) is around the $y$ axis and yaw ($\psi$) around the $z$ axis, as shown in Figure A.2.

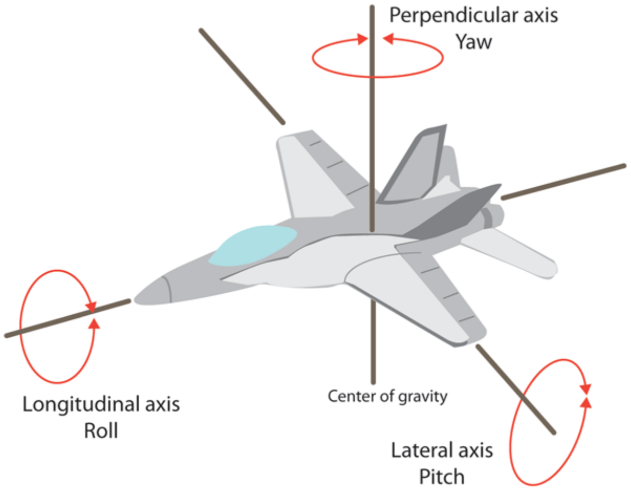


Figure A.2: RPY angles used in the aeronautical convention.

The Z-Y-X Euler angles convention starts by rotating the B-frame about its $z$ axis by an angle $\gamma$, then it is rotated about its $y$ axis by an angle $\beta$ and lastly it is rotated about its $x$ axis by an angle $\alpha$. The rotation matrix can be obtained from Euler angles using equation A.1.

$$^{A}_{B}\boldsymbol{R} = \boldsymbol{R}_z(\gamma)\boldsymbol{R}_y(\beta)\boldsymbol{R}_x(\alpha) \tag{A.1}$$

where

$$\boldsymbol{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix}, \boldsymbol{R}_y(\beta) = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix}, \boldsymbol{R}_z(\gamma) = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[1] In fact, the number of different combinations can be easily calculated. The first rotation can be around any axis (X,Y,Z). The second rotation is about either of the two axes not used in the first rotation. Finally, the last rotation is about one of the two axes not used in the second rotation. Hence, there are twelve different possible sets for Euler angles.

## A.2 Direction Cosine Matrix

A convenient way to represent the relative orientation between two coordinate frames is the DCM. In this representation, a vector $\vec{u}_B$ in the body frame is transformed to a vector $\vec{u}_A$ in the inertial frame using the orthogonal rotation matrix ${}^{A}_{B}\boldsymbol{R}$.

$$\vec{u}_A = {}^{A}_{B}\boldsymbol{R}\, \vec{u}_B = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \vec{u}_B \tag{A.2}$$

where $R_{ij} \equiv \vec{u}_A \cdot \vec{u}_B$ is the cosine of the angle between $\vec{u}_A$ and $\vec{u}_B$.

Euler angles can be obtained from the rotation matrix using the equations in A.3.

$$\begin{aligned} \beta &= atan2\left(-R_{31}, \sqrt{R_{11}^2 + R_{21}^2}\right) \\ \alpha &= atan2\left(\frac{R_{32}}{cos(\beta)}, \frac{R_{33}}{cos(\beta)}\right) \\ \gamma &= atan2\left(\frac{R_{21}}{cos(\beta)}, \frac{R_{11}}{cos(\beta)}\right) \end{aligned} \tag{A.3}$$

This relation becomes ill-defined when $\beta = \pm\frac{\pi}{2}$ rad which can be avoided by computing $\alpha = atan2(R_{11}, R_{12})$ when $\beta = \frac{\pi}{2}$ rad and its symmetric when $\beta = -\frac{\pi}{2}$ rad, for $\gamma = 0$ rad.

## A.3 Angle-axis representation

Another alternative representation for the rotation between two frames is the Euler-axis representation. This method require two parameters: the rotation angle and the Euler axis $\boldsymbol{\lambda} \in \mathbb{R}^3$ . The latter represents the direction normal to the rotation plane.

The rotation angle satisfies:

$$\phi = arccos\left(\frac{tr({}^{A}_{B}\boldsymbol{R}) - 1}{2}\right) \in [0, \pi] rad \tag{A.4}$$

and the Euler axis is obtained using equation A.5.

$$\boldsymbol{\lambda} = \frac{\theta}{2sin(\theta)}\left({}^{A}_{B}\boldsymbol{R} - {}^{A}_{B}\boldsymbol{R}^T\right) \tag{A.5}$$

This representation is not easy to visualize but is more compact than using rotation matrices. However, the Euler axis becomes ill-defined when $\theta = 0$ or $\theta = \pi$.

The rotation matrix is obtained from Euler angle-axis representation using equation A.6.

$$\boldsymbol{R} = rot(\theta, \boldsymbol{\lambda}) = cos(\theta)\boldsymbol{I}_3 + sin(\theta)(\boldsymbol{\lambda})_\times + (1 - cos(\theta))\boldsymbol{\lambda}\boldsymbol{\lambda}^T \tag{A.6}$$

where $(.)_\times$ is the cross-product matrix represented in A.7.

$$(\boldsymbol{e})_\times = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix} \tag{A.7}$$

## A.4 Quaternions

Quaternions are an extension of complex numbers in the 3D world. A quaternion is defined by a scalar component and a vector component, constructed using the standard vectors $\vec{i}$, $\vec{j}$ and $\vec{k}$. The scalar part of the quaternion represents the magnitude of the rotation and the vectorial part represents the axis of rotation. The generic quaternion equation is represented in equation (A.8).

$$q = s + x\hat{i} + y\hat{j} + z\hat{k} \tag{A.8}$$

The four dimensional vector representation of a quaternion is shown in (A.9).

$$\boldsymbol{q} = \begin{bmatrix} q_s & \boldsymbol{q_v} \end{bmatrix} = \begin{bmatrix} q_s & q_x & q_y & q_z \end{bmatrix} \tag{A.9}$$

The scalar and vectorial parts of a quaternion can be obtained from Euler angle-axis representation using equation A.10.

$$q_s = cos\left(\frac{\theta}{2}\right),\ \boldsymbol{q}_v = sin\left(\frac{\theta}{2}\right)\boldsymbol{\lambda} \tag{A.10}$$

The rotation matrix that corresponds to a given quaternion can be computed from

$$\boldsymbol{R}(\boldsymbol{q}) = (q_s^2 - ||\boldsymbol{q}_v||^2)\boldsymbol{I}_3 + 2\boldsymbol{q}_v\boldsymbol{q}_v^T + 2q_s(\boldsymbol{q}_v)_\times \tag{A.11}$$

## A.5 Advantages and disadvantages

The best method to represent the orientation of a rigid body depends on the application. In embedded systems, the choice of the attitude representation method should ensure a low computational burden. Therefore, the best method should require a reduced number of parameters to represent the attitude and also be straightforward when propagating successive rotations. For this reason, quaternions are usually chosen as the attitude representation method in embedded systems.

Euler angles are the most popular method because they are immediately intuitive. Usually, attitude systems require the calculation of these angles at some stage. However, they can be ambiguous due to their twelve possible formulations and they are difficult to manipulate mathematically, usually requiring the computation of the respective DCM in order to transform vectors between frames. Furthermore, they suffer from the Gimbal Lock, i.e., the loss of one degree of freedom due to the singularity present when $\beta = \pm\frac{\pi}{2}$, as previously discussed.

DCM representations can overcome the drawbacks of Euler angles representations. The time propagation equation used to propagate the vehicle's attitude from the previous time step to the current time step is a straightforward function, since the matrix already exists. Furthermore, the DCM is an orthonormal matrix, which means that its transpose and its inverse are the same. This property is useful to reduce the computational effort when inverting rotations. The main drawback of this method is the necessity of nine parameters to describe three angles. Hence, the normalization and orthogonalization of the DCM must be ensured in every time step, which increases the computational overhead. Moreover, it is complicated to intuitively obtain the rotation from a DCM representation.

Quaternions are simple to use and have no singularities. Since they are represented by four parameters, they are more efficient than the DCM representation. The time propagation equation is also computationally straightforward.