Faculdade de Engenharia da Universidade do Porto

Human Motion Analysis Using Inertial Sensors for Rehabilitation Purposes

Beatriz dos Santos Neves de Oliveira

VERSÃO FINAL

Dissertação realizada no âmbito do
Mestrado Integrado em Bioengenharia
Ramo Engenharia Biomédica

Orientador: Jorge Belinha (PhD)
Orientador externo: Ana Clara Matos (Engª)

February, 2018

Aos meus pais,

For the past 33 years, I have looked in the mirror every morning and asked myself: 'If today were the last day of my life, would I want to do what I am about to do today?'

-Steve Jobs

Abstract

Musculoskeletal diseases are the major cause of disability, being responsible for almost 30 percent of all years lived with disability and affecting around 1.7 billion people worldwide. The combined annual incidence of the most common musculoskeletal disorders (low back pain, osteoarthritis and bone fractures) is about 3150/100 000 inhabitants. This translates into around 230 million new cases annually, with a corresponding high demand on rehabilitation services.

Evidence regarding motor rehabilitation highlights the need for high intensity, repetitive task-specific practice with feedback on performance, active patient involvement and adequate motivation. There is also evidence for the benefit of an early start in the rehabilitation process. Current rehabilitation models are highly dependent on specialized human resources. Whilst the clinical validity of this approach is clearly established, high economic costs and heavy logistics are major impediments in a day-to-day setting, limiting both the access to rehabilitation therapies and the intensity of these programs. Therefore, inertial sensors present a more suitable technique. They can be attached to different body segments to estimate joint kinematics and they represent a promising technology due to their usability in internal and external environments and fast donning becoming, then, an alternative to high-cost optical systems.

The study of joints using inertial sensors presents a few challenges, such as variability in sensor placement, accumulation of integration errors of sensors themselves. Therefore, and as in part of the dissertation project of the Integrated Master in Bioengineering, it was proposed by SWORD Health, the development of an algorithm that will allow quality and repeatability in the analysis of joints in the context of motor rehabilitation. The algorithm developed was based on the work written by Seel et al, with some adjustments, in order to respond to the needs of motor rehabilitation. The algorithm proved to be able to converge correctly to the axis of the knee joint, regardless of the placement of the sensors in the human body. In addition, another extra algorithm was developed to, based on the previous convergence, find the position of the joint.

Resumo

As doenças músculo-esqueléticas são a principal causa de incapacidade motora, sendo responsável por quase 30% de todos os anos vividos com deficiência e afetam cerca de 1,7 bilhão de pessoas em todo o mundo. A incidência anual combinada dos distúrbios músculo-esqueléticos mais comuns (dor lombar, osteoartrite e fraturas ósseas) é de cerca de 3150/100 000 habitantes. Isso traduz-se em cerca de 230 milhões de novos casos por ano, com alta procura aos correspondentes serviços de reabilitação.

Pesquisas efetuadas na área da reabilitação motora realçam a necessidade de uma prática específica de tarefas repetitivas e de alta intensidade com comentários sobre desempenho, envolvimento ativo do paciente e motivação adequada. Também há evidências para o benefício de um início precoce no processo de reabilitação. Os modelos atuais de reabilitação são altamente dependentes de recursos humanos especializados. Embora a validade clínica desta abordagem seja claramente estabelecida, os altos custos económicos e a logística são impedimentos no contexto do dia-a-dia, limitando o acesso às terapias de reabilitação e a intensidade desses programas. Portanto, os sensores inerciais apresentam uma técnica mais adequada. Estes podem ser anexados a diferentes segmentos do corpo para estimar a cinemática articular e representam uma tecnologia promissora devido à sua usabilidade nos ambientes internos e externos e a colocação rápida tornando-se, então, uma alternativa aos sistemas óticos de alto custo.

O estudo de articulações utilizando sensores inerciais apresenta alguns desafios, como variabilidade na colocação de sensores, acumulação de erros de integração dos próprios sensores. Assim, como parte do projeto de dissertação do Mestrado Integrado em Bioengenharia, foi proposto pela SWORD Health, o desenvolvimento de um algoritmo que permitisse a qualidade e repetibilidade na análise das articulações no contexto da reabilitação motora. O algoritmo desenvolvido teve por base o trabalho escrito por Seel at al, com alguns ajustes, de forma a responder às necessidades da reabilitação motora. O algoritmo provou ser capaz de convergir corretamente para o eixo da articulação do joelho, independentemente da colocação dos sensores no corpo humano. Além disso, um outro algoritmo extra foi desenvolvido para, com base na convergência anterior, encontrar a posição da articulação.

Agradecimentos

Este trabalho não teria sido possível sem a contribuição de algumas pessoas, as quais, de forma direta ou indireta, ajudaram-me neste percurso. A estas gostaria de expressar o meu mais sincero agradecimento.

Começo por agradecer ao meu orientador, Professor Jorge Belinha, pela orientação, disponibilidade, por todas as reuniões infinitas e por me desafiar a fazer mais e melhor. Apesar de um início turbulento, o apoio e dedicação do professor para me ajudar foi essencial para o trabalho. Por tudo isto e mais, o meu sincero obrigada Professor.

À Clara Matos e ao Luís Húngaro obrigada pelo desafio proposto, por toda a paciência que tiveram para me explicar todos os conceitos necessários e pelo voto de confiança para desenvolver este projeto.

A toda a SWORD Health que me acolheu ao longo destes meses de braços abertos. Obrigada por me proporcionarem uma experiência empresarial de preparação para o meu futuro profissional.

Aos meus amigos, Maria Beatriz, Pinto, Gustavo, Clara, Graça, Bárbara, Manuel e Malheiro. Obrigada por me mostrarem que, independentemente de onde estivermos, posso sempre contar com a vossa amizade para o resto da minha vida. Obrigada por todos os risos, todos os momentos épicos e aventuras. São memórias que levo comigo.

À Inês, Patrícia e Joana porque mais do que primas, são as minhas irmãs. Obrigada por me aceitarem tal como eu sou, incentivarem-me sempre a seguir o meu caminho e a ser livre.

Por fim, aos meus pais e irmão. João, obrigada por nestes últimos anos me incentivares à prática de desporto, tornou-se numa das minhas maiores paixões. Pai e Mãe, estou eternamente grata por tudo o que fizeram e ainda fazem por mim. Obrigada por confiarem em mim, me apoiarem nas minhas decisões e por estarem sempre ao meu lado. Não podia ter escolhido família melhor.

Contents

List of Figures

List of Tables

List of Abbreviations

2D	Bidimensional
3D	Tridimensional
DOF	Degree of freedom
IMU	Inertial Measurement Unit
MEMS	Microelectromechanical System
MSK	Musculoskeletal System
NHANES	National Health and Nutrition Examination Survey
PD	Parkinson's Disease
WHO	World Health Organization

Chapter 1

Introduction

Within the scope of the dissertation project of the Integrated Master in Bioengineering, Biomedical Engineering, the student will carry out an internship at SWORD Health. This project aims to study and enhance/develop new algorithms that will ensure quality and repeatability in joint analysis using inertial sensors. Furthermore, these algorithms must be developed regarding human rehabilitation and they must give a real-time feedback.

1.1- Motivation and Objectives

Approximately 15% of the world population lives with a disability condition, of which 2-4% suffer substantial functional problems [7]. Hence, one of the major goals of neuromuscular rehabilitation is to regain function in order to promote more independent lives. The assessment of functional activities can help practitioners to determine the autonomy level of the patient and the optimal care they should receive [8]. Therefore, it is important to understand and characterize human motion as a mean to improve the diagnosis, enhance treatments and follow patients' evolution. A fundamental part of this analysis is the estimation of joint angular displacement, which involves the detection of joint position and spatial orientation [9].

There are many available techniques to detect human movement and they can be divided in visual (camera-based) or non-visual (inertial sensors, for example). In contrast to camera-based laboratory systems, wearable sensors, like inertial sensors, present many advantages, such as their lower cost, higher flexibility, portability and adaptability [10]. Inertial sensors are a multi-axial combination of accelerometers, gyroscopes and, in some cases, magnetometers. They can be attached to different body segments to estimate joint kinematics and they represent a promising technology due to their usability in internal and external environments and fast donning becoming, then, an alternative to high-cost optical systems [5].

The study of joints using inertial sensors presents a few challenges, such as the variability in sensor placement and the accumulation of integration errors of sensors themselves. These errors may cause a significant decrease in quality of the measures, thus, biomechanical models

are a way to overcome this issue. These models take into consideration the restrictions associated with degrees of freedom in joints and allow correcting observed deviations. Nevertheless, there are also other challenges inherent in the context of motor rehabilitation, for example, patient with limited mobility may difficult the calibration movements or other techniques that facilitate the implementation of these models.

Therefore, the objectives of this thesis include the study of algorithms and biomechanical models for human motion analysis. Also, the development of algorithms that will allow quality and repeatability in the analysis of joints in the context of motor rehabilitation.

1.2- SWORD Health

SWORD Health is a healthcare startup that is reinventing physical rehabilitation through the combination of science-driven therapeutic methods with effective technologies that facilitate dissemination of care. To solve the dependence of current rehabilitation models on specialized human resources, which are costly and scarce, SWORD Health has developed SWORD Phoenix, a system that combines small, portable motion sensors and powerful software allowing a more efficient and cost-effective approach to rehabilitation. SWORD Health has been endorsed by the European Commission as one of the most innovative companies in Europe, and has already established partnerships with several major rehabilitation institutions in Europe and in the US.

In the latter, SWORD Health has partnered with Genesis Rehab Services, the largest rehabilitation chain in the US. As a result, SWORD Phoenix is already being piloted in several institutions in Europe and China, with pilots in the US about to start.

1.3- Document Structure

The following documented is organized in 7 chapters, each one referring to the different parts of the problem resolution and validation.

In the first one, is presented the main goals of the study as well as a description of the document.

The second chapter, gives a notion of the human anatomy and explains the meaning of human rehabilitation. Furthermore, this chapter provides a socio-economical perspective of human rehabilitation in today's society and two different scenarios are specified: stroke and Parkinson's disease. For each, rehabilitation techniques and prevalence and incidence are provided, in order to raise the reader's awareness of human disabilities and how they affect daily lives. Further, a detailed description of constitution and principles of operation of inertial sensors is given and the usage of such sensors for human motion capture. Besides, it is presented an historical reference of the first work done in this context. Then, a survey on human motion capture techniques is provided as well as their advantages and disadvantages regarding the project's intent.

In the third chapter is presented the algorithm that will support this thesis. First, a brief explanation of the same is given, with the essential equations and assumptions and then a step by step tutorial of the algorithm is specified.

Chapter four describes the method used for the simulated data validation and then the results obtained, as well as a discussion of such results.

The fifth chapter shows the results obtained for the experimental trials performed by a subject and provides its corresponding discussion.

The initial objective of this thesis was to develop of a robust algorithm capable to identify the vector defining the knee articulation joint (its axis). However, this work went further and a new algorithm was is developed to find the joint position based on the joint axis achieved with the algorithm described in the fifth chapter. This extra work can be found in chapter six, in which the new algorithm is described with detail and its validation is shown and discussed.

Lastly, in chapter seven a summary of the work produced is provided, along with the final relevant conclusions and remarks. This final chapter also mentions some limitations encountered through this work and some possibilities for future works.

Chapter 2

Background and Literature Review

2.1- Human Motion

Biological musculoskeletal system (MSK), composed of numerous bones, cartilages, skeletal muscles, tendons, ligaments etc., provides form, support, movement and stability for human or animal body. As the result of million years of selection and evolution, the biological MSK evolved to be a nearly perfect mechanical mechanism to support and transport the human body. Additionally, it provides enormously rich resources inspiring engineers to innovate new technology and methodology and to develop robots and mechanisms as effective and economical as the biological systems [11]. In addition, development of predictive musculoskeletal models, capable of predicting body kinematics and kinetics, will be one of the major future research directions due to their great potential in clinical diagnosis, rehabilitation engineering and surgical planning [12].

2.1-1. General Anatomic Description

The skeleton, formed by bones and joints, is the supporting structure of the human body. The bones give rigid support to the soft tissues of the body and form levers which move due to muscle contractions, whereas the muscles form the locomotor and weight-bearing apparatus together with the bones and their joints (figure 2.1)[13].

Regarding the skeletal system, it includes over 200 bones, 85 of which are paired. Bone consists of cells and extracellular matrix, and its specific composition allows mineralization, which is a specific feature of bone[14]. According to shape, bones can be divided in long bones and flat [13]. The first ones, consist of the epiphyses, which are protrusions at the ends of the long bones [15], the diaphysis (constituting its shaft [16]) and the metaphyses, which are located between the epiphysis and the diaphysis [17]. The external shape of bones is formed by a dense cortical shell (cortical or compact bone), which is particularly strong along the diaphysis, where the bone marrow is located. On the other hand, trabecular bone (or cancellous bone) is a sponge-like network consisting of countless highly interconnected bony trabeculae

[14]. Although cortical and trabecular bone are composed of the same cells and same matrix components, cortical bone almost exclusively consists of mineralized tissue (up to 90%), allowing it to fulfill its mechanical requirements. In contrast, only 20% of trabecular bone is mineralized tissue. As a consequence, this bone shares a vast surface with the nonmineralized tissue, which is the basis for the metabolic function of bone, necessitating a high level of communication between the bone surface and the nonmineralized tissue [14].

Human joints provide the structures by which bones join with each other, and are classified according to histology features of the union and range of joint motion: fibrous joints and synovial joints. These two types of joints differ in the degree of mobility that they provide for the bones and in the capacity of bearing mechanical loads [13]. Synovial joints, which articulate with free movement, have a synovial membrane lining the joint cavity and containing synovial fluid [18]. Furthermore, this joint can also be classified according to their shapes, including ball-and-socket (hip), hinge (interphalangeal), saddle, and plane joints. The various designs permit flexion, extension, abduction, adduction or rotation. Certain joints can act in one (humeroulnar), two (wrist) or three (shoulder) axes of motion [14]. Fibrous joints provide uninterrupted union of bones, by means of connective tissues, permitting limited motion [13]. Ligaments play a major role in the passive stabilization of joints, aided by the capsule and, when present, the menisc. Moreover, they provide a stabilizing bridge between bones, permitting a limited range of movement [19].

The muscle form the locomotor and weight-bearing apparatus together with bone and respective joints. There are more than 660 skeletal muscles in the human body and they constitute up to 40% of the body mass [14]. Functionally, muscles are united to form a working group, responsible for motor activity. Each axis of rotation at a joint possesses its own pair of functional working groups: uniaxial joints have one pair, diaxial joints have two pairs and triaxial joints have three pairs of working groups [13]. Muscles accomplishing a single type action are called synergists, and the ones that act in opposition to another are called antagonists. Muscle is a use-dependent tissue, meaning that increasing muscle activity through physical exercise leads to adaptation in muscle fibers, fortifying the muscle. On the other hand, when activity is reduced, the cross-section of the fiber decreases, leading to muscle weakening. After significant periods of disuse, loss of muscle strength and endurance can progress to the point that patients are unable to accomplish daily activities. Although younger and healthy ones tend to regain muscle size and function with exercise rehabilitation, older individuals and those with chronic diseases cannot fully recover from muscle disuse [20]. Normal individuals exhibit an approximately 30% decrease in total muscle mass between the ages of 30 and 80 [14].

Figure 2.1- Representation of the human musculoskeletal system [21].

2.1-2. Motor Rehabilitation

Rehabilitation, from the Medieval Latin root word *rehabilitare*, literally means "to restore to a rank, it is similar to other types of physical conditioning. Rehabilitation is the process of applying stress to healing tissue in accordance with specific stresses that the tissue will face on returning to a particular activity. Thus, rehabilitation involves reconditioning injured tissue, and, essentially, body systems respond to physical stress by undergoing adaptations that ultimately improve their functioning [22]. When the healing tissue is mature, the emphasis moves to more aggressive conditioning in preparation to a specific task.

Musculoskeletal conditions are without doubt a major burden on individuals, health systems and social care systems [23]. In 2011, an estimated 37.9 million people, 12.2% of the U.S. population, were living with a disability [24]. Furthermore, according to the World Health Organization (WHO), there are 4 million people with Parkinson's disease (PD) worldwide, specifically, in Europe, prevalence rate estimates range from 65.6 per 100,000 to 12,500 per 100,000 [25]. According to the National Heart, Lung, and Blood Institute, annually in USA, 795,000 strokes occur, with 610,000 being first-time strokes and 185,000 related to a recurrence [26]. The American Heart Association also estimated an overall stroke prevalence of 6.8 million Americans, accounting for 2.8% of the population, based on National Health and Nutrition Examination Survey (NHANES) data from 2007 to 2010 [27].

Stroke

Stroke occurs due to interruption of blood supply to the brain or a result of ischemia bleeding and has a prevalence of approximately 795,000 new or recurrent events in the United States each year [28]. A common problem experienced by these populations is impaired upper extremity function. About 70% of stroke survivors lose motor skills of the paretic arm and hand, even mild impairment results in significant daily function limitations and has a negative impact

on the quality of life [29][30]. During rehabilitation, a process of relearning how to move to carry out their needs successfully, patients improve their activity by either developing compensatory strategies or by reacquisition of the pre-lesion patterns, defined as recovery [31].

Motor learning is fundamental for improvement of affected motor skills following brain lesion. When designing a rehabilitation training, it is very import to select appropriate outcome measures for patients' evaluation. According to the stroke survivors, the additional practice they seek must be meaningful. In some cases, the problem may simply be tasks for which the explanation of the biological basis of the therapy or the training intent is inadequate [32]. Goal setting in stroke rehabilitation has been identified as important for recovery, positive influence on patients' perceptions of self-care ability and engagement in rehabilitation [33].

The economic burden of stroke is impacted by initial hospitalization, medications, continuing medical care, including rehabilitation, and work limitations. In the United States, the average cost of a stroke hospitalization in 2005 was 8600 € per patient [34]. Over a lifetime, the cost of an ischemic stroke in the same country is more than 127,000 € per patient, including inpatient care, rehabilitation, and long-term care for lasting deficit [35]. Whereas, in Europe, the cost per subject are illustrated in the figure below, where direct healthcare costs include hospitalization, pharmaceuticals and rehabilitation, direct non-medical costs include social services and special accommodation and indirect costs concern lost production due to work absence or early retirement [36].

Table 2.1- Annual cost per subject with stroke in Europe, in euros (n.a: not available) [36].

Country	Direct healthcare costs	Direct non-medical costs	Indirect costs	Year
Germany	18,518	n.a.	n.a.	2004
Ireland	18,571	710	2821	2007
Italy	12,222	9012	1982	2005
Netherlands	19,511	n.a.	n.a.	2000

Parkinson's Disease

Parkinson's disease (PD) is among the most common neurodegenerative diseases. It is characterized by bradykinesia, tremor, rigidity and postural instability. Comorbidities such as mental disorders, autonomic dysfunction, difficulties in swallowing and speech as well as sleep impairment may occur during the course of the disease. In Europe, the prevalence is approximately 160 per 100,000 among those aged 65 and older and this number will considerably increase in the coming years. The number of persons and subsequently the burden of Parkinson's disease is expected to increase over time due to the aging European population [37].

Growing evidences suggest that exercise has positive effects on the quality of life of elderly people and subjects with neurodegenerative disorders. Exercise has been consistently shown to improve both motor and non-motor features of PD [38]. Moreover, exercise has been associated with a reduced risk of developing Parkinson's disease [39]. General physiotherapy (stretching, muscle strengthening, balance and postural exercises), occupational therapy, and treadmill training, are frequently adopted to improve specific aspects of mobility [40]. When designing a rehabilitative program for PD, exercise should be "goal based" that is targeted to practicing and learning specific activities in these core areas that are impaired (for instance, balance and gait control) thus leading to improved performance in daily living activities [41].

The admission rate ranged from 6% to 21% of patients, with highest rates found in Austria. Costs for inpatient days (hospitalization and rehabilitation) ranged from EUR 100 (95% CI: 50–150) to EUR 1600 (95% CI: 890–3020) [37]. In Portugal, inpatient costs represented a major cost component of the direct costs, including hospitalization and rehabilitation, consisted of EUR 770 (95% CI: 230–1940) for hospitalization and EUR 90 (95% CI: 6–430) for rehabilitation. The lowest inpatient costs were found in Eastern European countries where they comprised 5% of the total direct costs as compared to 13%–42% in Western European countries.

Table 2.2- Selected unit costs adjust to 2008, per country [37].

		Austria	Czech Republic	Germany	Italy	Portugal	Russia
	Type of costs (per drug package/per visit)	€	€	€	€	€	€
Medication	Levodopa/ benserazide 250 mg	40	58	80	18	7	40
	Dopamine-agonist	240	292	356	66	168	72
Inpatient Care	Rehabilitation (daily charge)	253	10	172	496	268	410
	Hospitalization	298	70	441	3773	638	410

2.2- Inertial Sensors

Inertial sensors were mainly used in aeronautics and maritime applications until the nineties because of the high cost associated with the high-accuracy requirements. With developments in micro-electro-mechanical systems (MEMS), the availability of small, lower-cost, medium-performance inertial sensors has opened up new possibilities for their use, such as the recognition of daily activities [42] physical therapy and home-based rehabilitation [43],

biomechanics [44], detecting and classifying falls, shock and vibration analysis, navigation of unmanned vehicles, and state estimation and dynamic modeling of legged robots [45].

Inertial measurement units (IMUs) typically contain gyroscopes and accelerometers, sometimes used in conjunction with magnetometers. Each device can be sensitive around a single axis or multiple axes (usually two or three). An accelerometer detects specific force, which is proportionate to the acceleration of the sensor relative to an inertial reference frame along its axis of sensitivity. A gyroscope senses the angular rate about an axis of sensitivity with respect to an inertial reference frame[46]. Magnetometers measure the magnetic field strength at a given location superposed with the Earth's magnetic field [47].

Currently, there are many inertial sensors available on the market with different specifications. In order to choose the best one that fits the problem's needs, there are certain specifications that need special attention, such as, range, interface, axes, power requirements and price. The price of an inertial sensor may vary from 10€ up to 400€, this discrepancy is reflected in the already mentioned specifications.

2.2-1. Accelerometers

Accelerometer is a sensor dedicated to measure its own linear acceleration. There are several classes of commercial accelerometers such as fluid, reluctive, servo, and magnetic [48]. However, classes that are more common to the investigation of human motion are strain gauge, piezoresistive, capacitive, and piezoelectric. The basic mechanism underlying acceleration measurement is often described in terms of a mass–spring system, which operates under the principles of Hooke's law, $F = kx$, and Newton's 2nd law of motion, $F = ma$. When a mass-spring system is submitted to a compression or stretching force due to movement, the spring will generate a restoring force proportional to the amount of compression or stretch. Given that mass, and the stiffness of the spring can be controlled, the resultant acceleration of the mass element can be determined from characteristics of its displacement:

$$F = kx = ma \tag{2.1}$$

Thus,

$$a = \frac{kx}{m} \tag{2.2}$$

In practice, acceleration is quantified using a number of techniques depending on the accelerometer class. For example, in a capacitive accelerometer, a silicon mass element is surrounded by an array of paired capacitors on opposite sides of the accelerometer housing. As the mass element reacts to movement, an imbalance is created between opposing capacitors that produce an electrical output signal proportional to the magnitude of the applied

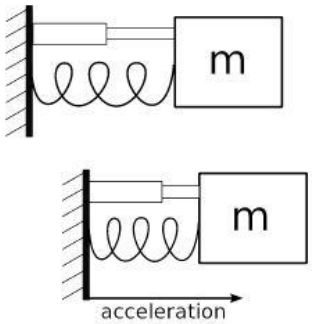

Figure 2.2 – Accelerometer principle[49].

acceleration. Regardless of accelerometer class, the relationship between the electrical output and a corresponding reference value of acceleration must be determined by calibration procedures performed under specific conditions [50].

In the literature, two categories of measurement errors are distinguished: stochastic errors (noise) and deterministic errors (calibration defects). Firstly, electronic sensors are disturbed by noise [51]. Thermal noise has the main influence on data collected from electronic sensors. As it is usually modeled with a Gaussian white noise, it has an impact over the entire frequency domain such that it cannot be filtered. Furthermore, the definition of the link between raw output signals and estimated acceleration can be a factor of loss of accuracy. Unlike noise, this error can be corrected by defining an adapted measurement model and following an accurate calibration process [52].

Usually, accelerometers are calibrated by using angular position-control or centrifuge machines. In the first, the accelerometer is placed and held stationary at various known reference orientations throughout the test. This is known as the multi-position or the 1g test. Calibration parameters are estimated based on the acquired sensor measurements and the reference accelerations associated with the reference orientations (i.e., the local gravity vector $\vec{g}$) [47]. However, this procedure has a limitation: the reference acceleration inputs applied to the sensors are restricted to the [−g, +g] interval, which may result in inaccurate calibration outside this interval. On the other hand, when centrifuge machines are used, reference acceleration inputs are not necessarily restricted to the [−g, +g] interval and higher acceleration values are sustainable [53]. Deterministic error components can then be identified in the same way as in the former procedure.

2.2-2. Gyroscope

A gyroscope is a device that measures angular rate around a certain axis of rotation. In order to do this, they rely on the physical principle of the Coriolis force. An object with a certain velocity v and an angular rate Ω around an axis orthogonal to the vector v, is subject to a Coriolis acceleration [54]:

$$a_{cor} = 2\Omega \times v \tag{2.3}$$

A corresponding Coriolis force acts on the object, with a direction orthogonal to the plane of both the axis of rotation and the direction of velocity, and with the following modulus:

$$F_{cor} = 2m\Omega v \tag{2.4}$$

Typically, MEMS gyroscopes are specialized vibrating accelerometers that measure Coriolis forces (figure 2.3). A basic vibratory gyroscope consists of a proof mass mounted on a suspension that allows the proof mass to move in two orthogonal directions. To generate a Coriolis force, the proof mass must be in motion. To this end, the proof mass is electronically forced to oscillate in a direction parallel to the chip surface. If the gyroscope chip is rotated about the axis perpendicular to the chip surface, then a Coriolis force causes the proof mass to be deflected in the second direction.

The amplitude of this oscillatory deflection is proportional to the rate of rotation, so that capacitive sensing, as in the case of the accelerometer discussed above, can be used to produce a voltage proportional to the angular rotation rate [55].

An angular velocity sensor with two or three axes is also possible. The triaxial micro-angular velocity sensor is mainly fabricated by the silicon-on-insulator technique, and it operates to detect the three-axes angular velocities. The outer ring is driven by the rotational comb electrodes to rotate, counterclockwise and clockwise alternatively, around the z-axis. Once the gyroscope is perturbed by Coriolis acceleration resulting from external rotation excitation around the y-axis, the outer ring responds to tilt in the direction of the x-axis. On the other hand, the inner-disc is forced to oscillate about the y-axis if the external rotation excitation is about the x-axis. All the tilts along x-axis or y-axis will result in the change of voltage output across the corresponding capacitors [56].

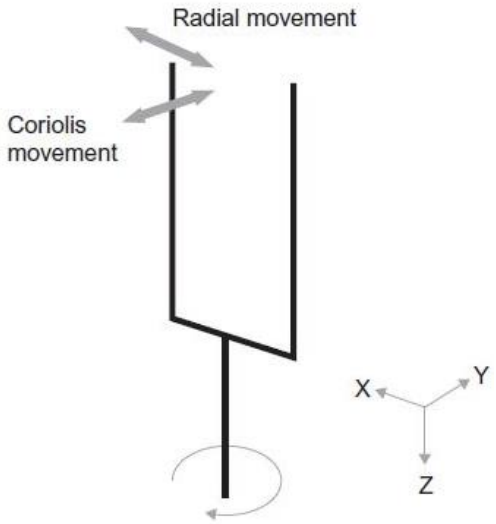

Figure 2.3 – The Coriolis force [57].

The calibration of gyroscopes depends on its precision. High-precision gyroscopes are capable of measuring the Earth's angular velocity, enabling the use of multi-position tests [46]. Gyroscopes can be positioned at reference orientations and the calibration parameters can be estimated by comparing the sensor measurements with the reference angular rate at these positions [47]. However, lower-grade MEMS gyroscopes need to be exposed to different reference angular velocities that can be provided by an angular position-control machine or a single-axis rate table [58]. The latter allows accurate calibration over a broader operation range compared to the former. The calibration parameters can be estimated by comparing the sensor measurements with the reference angular rates [59]. If the reference angular rates are not available, alternative techniques are required:

i. Calibrated accelerometers and/or magnetometers, embedded in the sensor unit together with the gyroscope, can be used to provide the reference information. If accelerometers are employed, measurements at stationary positions are compared with the gravity projected onto the sensor sensitivity axes by an angular transformation computed using gyroscopic measurements [60]. However, accelerometers only detect the two angles between the sensor and the local horizontal direction. Therefore, their measurements cannot resolve the rotation about the local vertical direction. Alternatively, it can be used embedded magnetometers that project the Earth's magnetic field onto the sensor's sensitivity axes as reference information [61]. Calibration is commonly performed by simply rotating the sensors by hand on a flat surface.

ii. When there is no additional sensor embedded in the sensor, the gyroscope can be fixed to one of the contact surfaces of a right-angled plate and rotated by hand on a flat surface. In this case, the attitude computed by gyroscope measurements is compared with the reference attitude associated with the plate's configuration [62].

2.2-3. Magnetometer

There are many different methods to sense a magnetic field, depending on the intensity of the signal to be measured. Given the large number of potential applications, there is a growing interest in developing miniaturized and low power magnetic field sensors [63].

The most promising approaches for the implementation of a magnetometer in MEMS technology, is based on the Lorentz force principle with capacitive readout [63], because it does not require the integration of magnetic materials in the realization process [54].

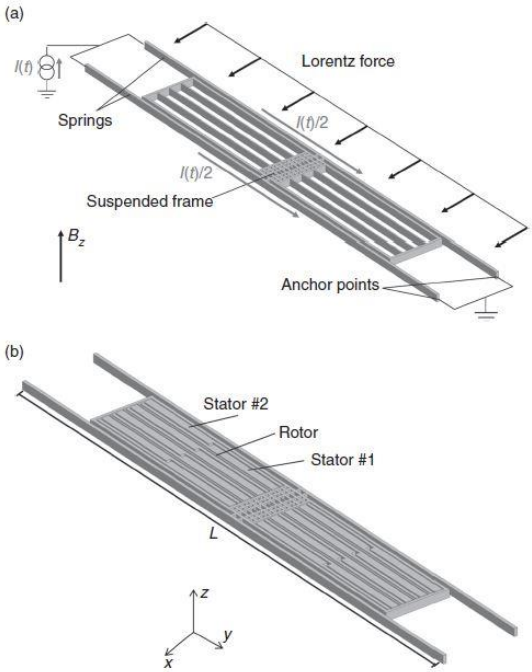

Figure 2.4 - Representation of a MEMS magnetometer. (a) View of the suspended frame with a schematic of the working principle based on the Lorentz force arising in presence of a current $I(t)$ and a magnetic field oriented in the z direction; (b) view of the full device [54].

As can be seen in figure 2.4, a pair of longitudinal springs of length L holds a suspended shuttle that, with suitable stators, forms a set of differential capacitors C_1 and C_2. Suppose that a current $I(t)$ is induced in both springs. In the presence of a component B_Z of the magnetic field in the direction orthogonal to the plane of the figure, a force arises which has a total intensity:

$$|F_L(t)| = B_Z I(t) L \tag{2.5}$$

The force is orthogonal to the plane of both B_Z and $I(t)$. As a mean to amplify the displacement obtained from such a force, the device can be operated at resonance, with a displacement amplification given by the quality factor. Assuming that the current has the expression: $I(t) = I_0 \sin(2\pi f_0 t)$, where f_0 is the device resonance frequency. If the device is packaged at low pressure, the displacement is amplified by the quality factor Q and is:

$$x(t) = \frac{|F_L|}{2k} Q = \frac{B_Z I(t) L}{2k} Q \tag{2.6}$$

Other advantage of this modulation is that it modulates the signal at a large frequency, possibly out of the $1/f$ noise of the readout electronics. Consideration should be given here to the maximum Q that can be used; as the measurement bandwidth after the signal demodulation turns out to be equal to $BW = f_0/2Q$, in order to cope with typical required bandwidths for

consumer and automotive applications, the quality factor should not exceed a value which is 100 times lower than the resonance frequency.

Therefore, there are only a few ways to increase, the mechanical sensitivity of Lorentz force capacitive magnetometers: the maximization of the device length L; and increasing the driving current $I(t)$, which is however limited by power dissipation constraints.

Magnetometers are also calibrated based on data acquired during multi-position tests. The reference input is the Earth's local magnetic field vector [64][65][66]. These devices need to go through testing on the platform on which they will be used, and their calibration parameters need to be estimated specifically for this platform. The orientation of the platform is often controllable and it is capable of producing the motions required for the multi-position test [64][67]. When this is the case, the interior effects of the platform can be modeled as constant time-invariant distortion since there is no relative motion between the platform and the magnetometer. However, the platform and the magnetometer are usually not isolated from the environment. External magnetic sources, such as electric motors or transformers, may affect the magnetometer measurements [64] and contribute additional time-varying distortion components [68]. Nevertheless, in most of the earlier studies, external distortion components are assumed to be constant for a given calibration platform.

2.3- Segment-to-sensor Alignment

When estimating joint kinematics using inertial sensors, a sensor-to-segment axis alignment is a crucial factor that has to be taken into account and it is needed for the sake of the functional readability of the measured or derived information content. In order to be called "joint kinematics", it has to be the relative orientation between the anatomical axes of two adjacent body segments rather than solely the relative orientation between the axes of two adjacent body-fixed sensors [69]

To estimate a functionally 3D joint kinematics using inertial sensors, for each involved body segment, the orientation of the axes of the anatomical reference system representing the orientation of the body segment has to be known with respect to the orientation of the sensor-embedded reference system. This relationship is assumed to be time-invariant and once it is known, it will be sufficient solely to record the time-varying orientation of the sensor-embedded system of reference [69]. In the next chapter, a deeper development of this topic will be exposed.

2.4- Human Motion by Inertial Sensors

Cappozzo and colleagues defined human movement analysis as "gathering quantitative information about the mechanics of the musculo-skeletal system during the execution of a motor task"[70]. A large variety of kinematic variables and parameters may be used for

characterizing this motor task [71]. Furthermore, a number of indices can be retrieved from joint kinematics for characterising what is called interjoint or interlimb coordination: an altered coordination pattern indeed characterises numerous musculoskeletal [72][73] and neurological diseases [74][75].

The human movement analysis consists in sensor-based measurement techniques aimed to objectively describe and quantitatively assess the motor functions and the motor abilities of a subject. Usually, a standard laboratory of human movement analysis is composed of multi-camera motion capture systems via the kinematics and kinetics of body joints. However, this technique requires specialized laboratories, expensive equipment, and long set-up and post-processing times. Nevertheless, technology advances in the field of motion measurement techniques have allowed to measure the kinematics of body segments with wearable inertial sensors such as accelerometers and gyroscopes instead of optoelectronic systems. This technique has many advantages over the first one, such as: low cost, small dimensions and light weight, and the absence of limitation of the testing environment to a laboratory[57].

2.4-1. A Brief “Historical” Perspective

The first studies that estimated segment orientation date back to 1973, when Morris used six uniaxial accelerometers coupled on a rigid bar for solving the equation regarding the motion of a rotating rigid body and determining its angular acceleration [76]. Then, angular velocity and displacement were computed by first and double integration of angular acceleration, respectively. Actually, the direct measurement of the angular velocity was carried out by Bortz two years before, using a gyroscope, while the orientation of the object was estimated by the numerical integration of the measured angular rate [77].

Regarding joint kinematics using inertial sensors, the first study took place in 1990 by Willemsen [78]. Since then, over the past 25 years, a variety of methodological approaches have been presented to estimate 2D and 3D joint kinematics using wearable inertial sensors. A list of methodological contributions and their relevant specifications can be found in table 2.3.

Table 2.3 - Main Characteristics of the methodological approaches, in chronological order, proposed to estimate joint angular kinematics.

Study	What	Devices/ Segment	How	Drift Correction	Alignment
Willemsen 1990	Knee flex-ext	2 (biaxial accel)	By comparing the equivalent accelerations of proximal and distal body segment at the connecting hinge joint; joint's acceleration from rigid body angular motion equation different between the planar orientation of 3 adjacent body segments; segment's orientation computed by numerical of angular velocity as Willemsen	Not required	Manual
Tong 1999	Knee flex-ext	1 (uniaxial gyro)		Kinematic reset	Not necessary
Dejnabadi 2005	Knee flex-ext	1 (biaxial accel + monoaxial gyro)		Not required	Picture
O'Donovan 2007	3D ankle kinematics	1 (mag-aided IMU)	Relative orientation between the proximal and distal segment's 3D frames	Sensor fusion	Functional
Findlow 2008	Hip knee ankle flex-ext	1 (IMU)	Neural networks-based prediction from measured segmental linear acceleration and angular velocities	Not required	Not necessary
Picerno 2008	Hip knee ankle 3D kin.	1 (mag-aided IMU)	Relative orientation between the proximal and distal segment's 3D frames	Sensor fusion	Anatomical
Favre 2009	3D knee kin.	1 (IMU)		Sensor fusion	Anatomical
Takeda 2009	Knee flex-ext & abd-add	1 (IMU)	As Dejnabadi	Not required	Functional
Cooper 2009	Knee flex-ext	1 (IMU)	As Tong	Sensor fusion + joint constrains	Not necessary
Cutti 2010	Hip knee ankle 3D kin.	1 (mag-aided IMU)	Relative orientation between the proximal and distal segment's 3D frames	Sensor fusion	Functional
Djuric-Jovicic 2011	Knee flex-ext	2 (biaxial accel)	As Tong; segment's orientation is computed by double numerical integration of the angular acceleration determined from rigid body angular motion equation	High-pass filtering	Not necessary
Seel 2014	Knee flex-ext	1 (IMU)	Knee flexion-extension angles	Weighted average of the two estimates	Functional

2.4-2. Human Motion Tracking Techniques

Human motion tracking systems are expected to generate real-time data that dynamically represents changes of human segments, or a part of it, based on well-developed motion sensor technologies [79]. In general, a tracking system can be non-visual, visual-based or a

combination of both. A classification of the available sensor techniques is illustrated in figure 2.5.

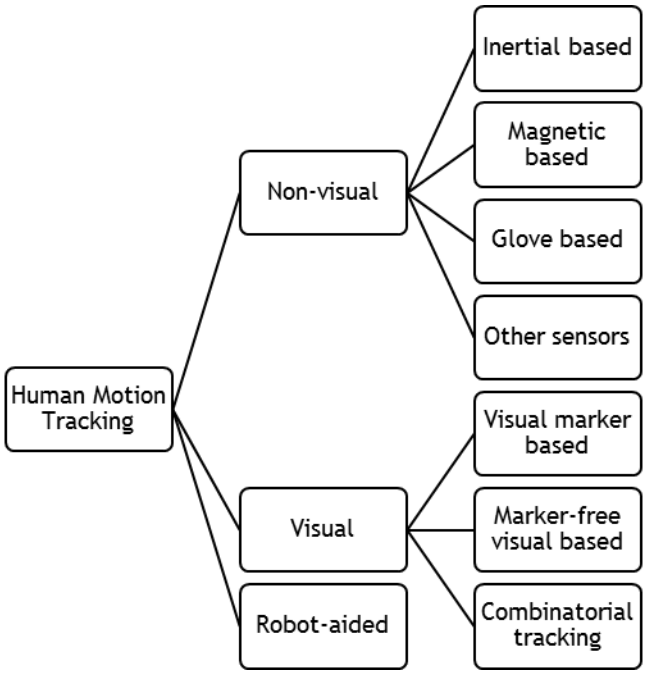

Figure 2.5 - Classification of human motion tracking using sensors technology [80].

Table 2.4 - Performance comparison of different motion tracking systems according to figure 4.2 [80].

Systems	Accuracy	Compactness	Computation	Cost	Drawbacks
Inertial	High	High	Efficient	Low	Drifts
Magnetic	Medium	High	Efficient	Low	Ferromagnetic materials
Ultrasound	Medium	Low	Efficient	Low	Occlusion
Glove	High	High	Efficient	Medium	Partial posture
Marker	High	Low	Inefficient	Medium	Occlusion
Marker-free	High	High	Inefficient	Low	Occlusion
Combinatorial	High	Low	Inefficient	High	Multidisciplinary
Robot	High	Low	Inefficient	High	Limited motion

For non-visual tracking, sensors engaged with these systems adhere to the human body in order to collect movement information. They are commonly categorized as mechanical, inertial, acoustic, radio, or microwave and magnetic based, and some of them have such small footprints that they can detect small amplitudes, for example finger or toe movements [80].

Regarding visual based tracking systems, they use optical sensors, for example cameras, to improve accuracy in position estimation. The further classification as marker or marker-free depends on whether or not indicators need to be attached in the body. Visual marker based systems, like VICON or Optotrack, use cameras to track the movement with identifiers placed upon the human body. On the other hand, marker-free tracking systems only exploit optical sensors to measure movements.

Robot-aided tracking systems are a subset of therapeutic robots. The rendering position/orientation of limbs is a necessary requirement in order to guide limb motion. There are a wide variety of rehabilitation systems driven by this strategy, such as MIT-MANUS [81], MIME [82] and ARM Guide [83].

In the recent years, inertial sensors have been an alternative to visual capture systems regarding motion tracking research. These devices are cost-effective and can be successfully used for accurate, non-invasive and portable motion tracking [84]. The big interest in these devices is mainly motivated by the fact that they overcome many issues raised by optical systems and mechanical trackers. Inertial sensors, indeed, do not suffer from occlusions and have theoretically unlimited workspace compared to optical motion tracking systems, and despite the accuracy of mechanical trackers, IMUs are much more affordable and far less intrusive. However, a fundamental problem of the IMUs is how to define an appropriate measurement protocol and provide a sensor-to-body calibration procedure [84]. Because IMUs' local frames are not aligned with anatomically defined frames, different approaches in the literature have presented different methods to determine the sensor frame's orientation with respect to the body segment frame.

2.4-3. Quaternions

A quaternion is a four-dimensional complex number that can be used to represent the orientation of a rigid body or coordinate frame in three-dimensional space. An arbitrary orientation of frame B relative to frame A can be achieved through a rotation of angle θ around and axis ${}^{A}\hat{\boldsymbol{r}}$ defined in a frame A. This is presented in figure 2.6, where the mutually orthogonal unit vectors $\hat{\boldsymbol{x}}_A$, $\hat{\boldsymbol{y}}_A$ and $\hat{\boldsymbol{z}}_A$, and $\hat{\boldsymbol{x}}_B$, $\hat{\boldsymbol{y}}_B$ and $\hat{\boldsymbol{z}}_B$ define the main axis of coordinate frames A and B, respectively. Equation 4.1 describes the quaternion orientation ${}^{A}_{B}\hat{\boldsymbol{q}}$, in which r_x, r_y and r_z define the components of the unit vector ${}^{A}\hat{\boldsymbol{r}}$ in frame A axis x, y and z, respectively. A notation system of leading super-scripts and sub-scripts is used to denote the relative frames of orientations and vectors. Quaternion arithmetic often requires that a quaternion describing an orientation is first normalized. It is therefore conventional for all quaternions describing an orientation to be of unit length. A quaternion with unity norm is referred to as unit quaternion and can be used to represent the attitude of a rigid body [85]. Therefore, the unit quaternion is represented by:

$$ {}^{A}_{B}\hat{\boldsymbol{q}} = [q_1 \quad q_2 \quad q_3 \quad q_4] = \left[\cos\frac{\theta}{2} \quad -r_x\sin\frac{\theta}{2} \quad -r_y\sin\frac{\theta}{2} \quad -r_z\sin\frac{\theta}{2}\right] \tag{2.7} $$

The quaternion conjugate, denoted by *, can be used to swap the relative frames described by an orientation. Thus, the conjugate of ${}^{A}_{B}\hat{\boldsymbol{q}}$ is:

$$ {}^{A}_{B}\hat{\boldsymbol{q}}^{*} = {}^{B}_{A}\hat{\boldsymbol{q}} = [q_1 \quad -q_2 \quad -q_3 \quad -q_4] \tag{2.8} $$

The quaternion product, denoted by $\otimes$, can be used to define compound orientations. For example, for two orientations described by ${}^{A}_{B}\widehat{\boldsymbol{q}}$ and ${}^{B}_{C}\widehat{\boldsymbol{q}}$, the computed orientation ${}^{A}_{C}\widehat{\boldsymbol{q}}$ is defined by:

$$ {}^{A}_{C}\widehat{\boldsymbol{q}} = {}^{B}_{C}\widehat{\boldsymbol{q}} \otimes {}^{A}_{B}\widehat{\boldsymbol{q}} \tag{2.9} $$

For two quaternions, $\boldsymbol{a}$ and $\boldsymbol{b}$, the quaternion product can be determined using the Hamilton rule and defined as equation (2.10). Furthermore, a quaternion product is not commutative, that is $\boldsymbol{a} \otimes \boldsymbol{b} \neq \boldsymbol{b} \otimes \boldsymbol{a}$. Being,

$$ \boldsymbol{a} \otimes \boldsymbol{b} = [a_1 \quad a_2 \quad a_3 \quad a_4] \otimes [b_1 \quad b_2 \quad b_3 \quad b_4] = \begin{bmatrix} a_1 b_1 - a_2 b_2 - a_3 b_3 - a_4 b_4 \\ a_1 b_2 + a_2 b_1 + a_3 b_4 - a_4 b_3 \\ a_1 b_3 - a_2 b_4 + a_3 b_1 + a_4 b_2 \\ a_1 b_4 + a_2 b_3 - a_3 b_2 + a_4 b_1 \end{bmatrix} \tag{2.10} $$

A three dimensional vector can be rotated by a quaternion using the relationship described in equation (2.11)[86]. ${}^{A}\boldsymbol{v}$ and ${}^{B}\boldsymbol{v}$ are the same vector described in frame A and frame B respectively where each vector contains a 0 (zero) inserted as the first element to make them 4 element row vectors.

$$ {}^{B}\boldsymbol{v} = {}^{A}_{B}\widehat{\boldsymbol{q}} \otimes {}^{A}\boldsymbol{v} \otimes {}^{A}_{B}\widehat{\boldsymbol{q}}^{*} \tag{2.11} $$

The orientation described by ${}^{A}_{B}\widehat{\boldsymbol{q}}$ can be represented by the rotation matrix ${}^{A}_{B}\boldsymbol{R}$:

$$ {}^{A}_{B}\boldsymbol{R} = \begin{bmatrix} 2q_1^2 - 1 + 2q_2^2 & 2(q_2\, q_3 + q_1\, q_4) & 2(q_2\, q_4 - q_1\, q_3) \\ 2(q_2\, q_3 - q_1\, q_4) & 2q_1^2 - 1 + 2q_3^2 & 2(q_3\, q_4 + q_1\, q_2) \\ 2(q_2\, q_4 + q_1\, q_3) & 2(q_3\, q_4 - q_1\, q_2) & 2q_1^2 - 1 + 2q_4^2 \end{bmatrix} \tag{2.12} $$

Regarding orientation of a sensor and this explanation, with equation (2.11), a 3D vector that could be the IMU output, described in sensor frame A, can be rotated by a quaternion, giving the same vector but described in the earth reference frame B.

Therefore, quaternions are used to represent orientation to improve computational efficiency and avoid singularities. Besides, the use of quaternions eliminates the need for computing trigonometric functions [87]. The major disadvantages of using unit quaternions are: that the four quaternion parameters do not have intuitive physical meanings, and that a quaternion must have unity norm to be a pure rotation. The unity norm constraint, which is quadratic in form, is particularly problematic if the attitude parameters are to be included in an optimization, as most standard optimization algorithms cannot encode such constraints [88].

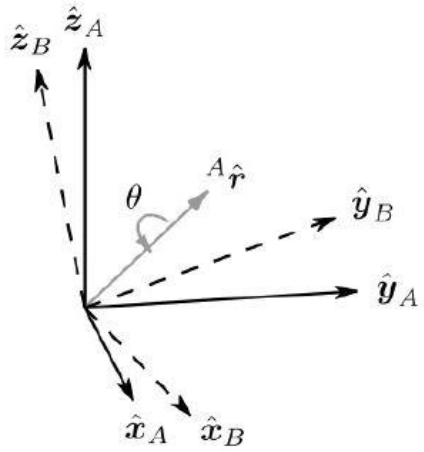

Figure 2.6 - The orientation of frame $\boldsymbol{B}$ is achieved by a rotation, from alignment with frame $\boldsymbol{A}$, of angle $\boldsymbol{\theta}$ around the axis ${}^{\boldsymbol{A}}\boldsymbol{r}$ [85].

Human body tracking using inertial sensors requires an attitude estimation filter capable of tracking in all orientations. Singularities associated with Euler angles make them unsuitable for human body tracking applications. So, quaternions are an alternative method of orientation representation and its rotation is more efficient than the use of rotation matrices and does not involve the use of trigonometric functions [89].

Let $\boldsymbol{q}(t)$ be a unit quaternion function and $\omega(t)$ be the angular velocity determined by $\boldsymbol{q}(t)$. The time derivative of $\boldsymbol{q}(t)$ is given by:

$$\dot{\boldsymbol{q}} = \tfrac{1}{2}\omega\boldsymbol{q}. \tag{2.13}$$

At $t + \Delta t$, the rotation is described as $\boldsymbol{q}(t + \Delta t)$, after some extra rotation during Δt is done on the frame that as already undergo a rotation described by $\boldsymbol{q}(t)$. The supposed extra rotation is about the instantaneous axis $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}/\|\boldsymbol{\omega}\|$ through the angle $\Delta\theta = \|\boldsymbol{\omega}\|\Delta t$. It can be described by a quaternion:

$$\Delta\boldsymbol{q} = \cos\frac{\Delta\theta}{2} + \hat{\boldsymbol{\omega}}\sin\frac{\Delta\theta}{2} = \cos\frac{\|\boldsymbol{\omega}\|\Delta t}{2} + \hat{\boldsymbol{\omega}}\sin\frac{\|\boldsymbol{\omega}\|\Delta t}{2}. \tag{2.14}$$

The rotation at $t + \Delta t$ is, therefore, described by the quaternion sequence $\boldsymbol{q}(t)$, $\Delta\boldsymbol{q}$, implying:

$$\boldsymbol{q}(t + \Delta t) = \Delta\boldsymbol{q}\,\boldsymbol{q}(t). \tag{2.15}$$

In order to obtain the derive $\dot{\boldsymbol{q}}(t)$, first, it is necessary to obtain the difference:

$$\begin{aligned}\boldsymbol{q}(t + \Delta t) - \boldsymbol{q}(t) &= \left(\cos\frac{\|\boldsymbol{\omega}\|\Delta t}{2} + \hat{\boldsymbol{\omega}}\sin\frac{\|\boldsymbol{\omega}\|\Delta t}{2}\right)\boldsymbol{q}(t) - \boldsymbol{q}(t)\\ &= -2\sin^2\frac{\|\boldsymbol{\omega}\|\Delta t}{4}\boldsymbol{q}(t) + \hat{\boldsymbol{\omega}}\sin\frac{\|\boldsymbol{\omega}\|\Delta t}{2}\boldsymbol{q}(t).\end{aligned} \tag{2.16}$$

Since the first term in the above equation is of higher order than Δt, thus its ratio to Δt goes to zero as the latter does. Consequently,

$$\dot{\boldsymbol{q}}(t) = \lim_{\Delta t \to 0} \frac{\boldsymbol{q}(t+\Delta t) - \boldsymbol{q}(t)}{\Delta t} = \hat{\boldsymbol{\omega}} \lim_{\Delta t \to 0} \frac{\sin \frac{\|\boldsymbol{\omega}\| \Delta t}{2}}{\Delta t} \boldsymbol{q}(t) = \hat{\boldsymbol{\omega}} \frac{d}{dt} \sin \frac{\|\boldsymbol{\omega}\| t}{2} \Big|_{t=0} \boldsymbol{q}(t)$$
$$= \hat{\boldsymbol{\omega}} \frac{\|\boldsymbol{\omega}\|}{2} \boldsymbol{q}(t) = \frac{1}{2} \boldsymbol{\omega}(t)\, \boldsymbol{q}(t). \tag{2.17}$$

Often, the angular velocity is in terms of the rotated frame, denoted by ω', thus, $\omega' = \boldsymbol{q}^* \omega \boldsymbol{q}$. Then, the following expression is obtained:

$$\dot{\boldsymbol{q}} = \frac{1}{2} \boldsymbol{q} \omega'. \tag{2.18}$$

If $\dot{\boldsymbol{q}}$ is known, the angular velocity can be achieved from equation 2.13 by right multiplying its both sides with $\boldsymbol{q}^*$:

$$\omega = 2 \dot{\boldsymbol{q}} \boldsymbol{q}^*. \tag{2.19}$$

2.4-4. Joint Reference

In order to specify the position of the body, segment, or object, a reference system is necessary for describing motion when motion has occurred. The reference frame or system is arbitrary and may be within or outside of the body. The reference frame consists of imaginary lines (axes) that orthogonally intersect each other at a common point (origin). The origin of the reference frame is placed at a designated location, such as a joint center. The axes are generally given letter representations to differentiate the direction in which they are pointing. Any position can be described by identifying the distance of the object to each of the axes. In a three-dimensional movement, there are three axes, two horizontal axes that form a plane and a vertical axis. It is always necessary to identify the frame of reference used in the description of motion.

For describing human movement, is used a method based on a system of planes and axes. A plane is a flat, two-dimensional surface. Movement is supposed to occur in a specific plane if it is along that plane or parallel to it. Movement in a plane always occurs about an axis of rotation perpendicular to the plane (see figure 2.7). These planes allow full description of a motion.

The movement in a plane can also be described as a single degree of freedom (DOF). This terminology is commonly used to describe the type and amount of motion structurally allowed by the anatomical joints. Joints with one DOF indicate that the joint allows the segment to move through one plane of motion. Moreover, a joint with one DOF is also termed uniaxial because one axis is perpendicular to the plane of motion about which movement occurs.

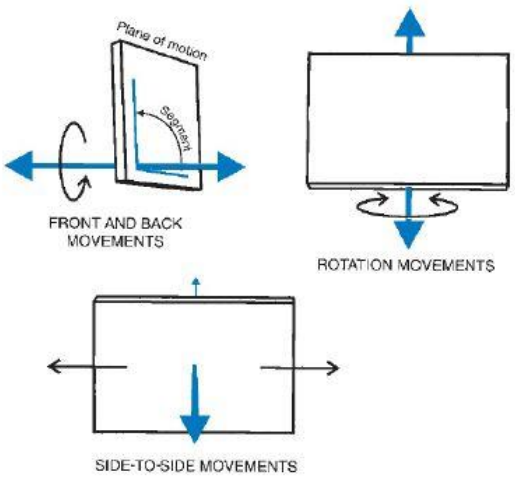

Figure 2.7 - The plane and axis. Movement takes place in a plane about an axis perpendicular to the plane.[90]

Therefore, anatomical movement descriptors should be used to describe segmental movements. This requires the knowledge of the starting position (anatomical), standardized use of segment names (arm, forearm, hand, thigh, leg, and foot), and the correct use of movement descriptors (flexion, extension, abduction, adduction, and rotation)[90].

2.5- Tracking Algorithms

A review of the literature was performed, in order to understand and find already developed algorithms for human rehabilitation using inertial sensors. Some of the algorithms described in the literature are present below.

2.5-1. Daniel Roetenberg, Henk Luinge and Per Slycke [91]

The MVN motion capture system consists only of body worn sensors, it estimates body segment orientation and position changes by integration of gyroscope and accelerometer signals, which are continuously updated by using a biomechanical model of the human body. The initial pose between the sensors and body segments is unknown, and the assessment of distances between body segments is difficult to obtain by numerical integration of acceleration because of the unknown initial position. Therefore, to express segment kinematics in the global frame, the kinematics of the sensors must be subjected to a step of calibration, in which the orientation of the sensor module with respect to the segment and the relative distances between joints are determined (see figure 2.8).

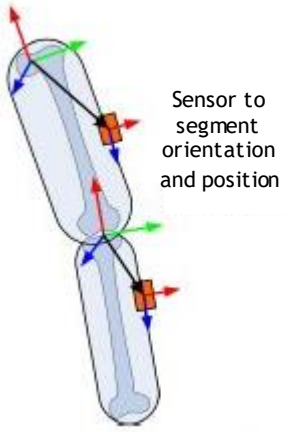

Figure 2.8 - Sensor to segment alignment. Adapted from[91].

In order to find the sensor alignment, this system combines several steps. The first one, the subject is asked to stand in a T-pose (upright with arms horizontally and thumbs forward) or N-pose (arms neutral besides body). The rotation from sensor to body segment ${}^{BS}\boldsymbol{q}$ is obtained by matching the orientation of the sensor in a global frame ${}^{GS}\boldsymbol{q}$ with the know orientation of each segment ${}^{GB}\boldsymbol{q}$ in this pose.

$$ {}^{GB}\boldsymbol{q} = {}^{GS}\boldsymbol{q} \otimes {}^{BS}\boldsymbol{q}^{*}, \tag{2.20} $$

where $\otimes$ denotes a quaternion multiplication and * the complex conjugate of the quaternion.

The second optional step consists in performing a certain movement that is assumed to correspond to a certain axis. The measured orientation and angular velocity are used to find the sensor orientation with respect to the segment's functional axes.

The final step in the calibration procedure, comprises the sensor to segment alignment and segments lengths can be re-estimated by using a priori knowledge about the distance between two points in a kinematic chain (see figure 2.9). This closed kinematic chain can be solved, which will improve the calibration values.

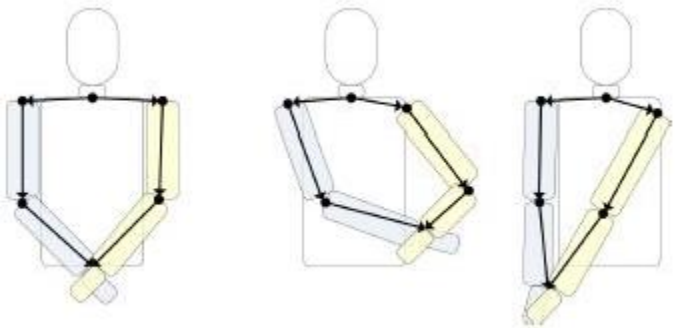

Figure 2.9 - Hand touch calibration.

Gyroscopes measure angular velocity ω, and when integrated over time provide the change in angle with respect to a known angle:

$$^{GS}\dot{\boldsymbol{q}}_t = {}^{\frac{1}{2}GS}\boldsymbol{q}_t \otimes \Omega_t, \tag{2.21}$$

where $^{GS}\boldsymbol{q}_t$ is the quaternion that describes the rotation from the sensor, S, to global frame, G, at time t. The quaternion representation on angular velocity $\boldsymbol{\omega}_t$ is given by $\Omega_t = (0, \omega_x, \omega_y, \omega_z)^T$.

Accelerometers measure the vector of acceleration $\boldsymbol{a}$ and gravitational acceleration $\boldsymbol{g}$ in sensor coordinates, and the sensor signal can be expressed in the global frame if the orientation $^{GS}\boldsymbol{q}_t$ is known:

$$^{G}\boldsymbol{a}_t - {}^{G}\boldsymbol{g} = {}^{GS}\boldsymbol{q}_t \otimes ({}^{S}\boldsymbol{a}_t - {}^{S}\boldsymbol{g}) \otimes {}^{GS}\boldsymbol{q}_t^* \tag{2.22}$$

When the gravitational component is removed, the acceleration $\boldsymbol{a}_t$ can be integrated, in the global frame, once to velocity $\boldsymbol{v}_t$ and twice to position $\boldsymbol{p}_t$:

$$^{G}\ddot{\boldsymbol{p}}_t = {}^{G}\boldsymbol{a}_t \tag{2.23}$$

Joint origins are determined by anatomical frame and are defined in the center of the functional axes with directions X, Y and Z being related to functional movements (see figure 2.10).

When it is known the position of joint origin $\boldsymbol{p}_{U0}$, the orientation $^{GB}\boldsymbol{q}_U$ and the length s_U of the segment U, then, the position $\boldsymbol{p}_{U1}$ in the global frame is given by:

$$^{G}\boldsymbol{p}_{U1} = {}^{G}\boldsymbol{p}_{U0} + {}^{GB}\boldsymbol{q}_U \otimes {}^{B}\boldsymbol{s}_U \otimes {}^{GB}\boldsymbol{q}_U^* \tag{2.24}$$

For $t = 0$, the origin of segment L with point $\boldsymbol{p}_{L0}$, is connected to point $\boldsymbol{p}_{L1}$ of segment U, as illustrated in figure 2.11.

In cases where there is no sensor attached, the kinematics are estimated based on the biomechanical model incorporating stiffness parameters between connecting segments.

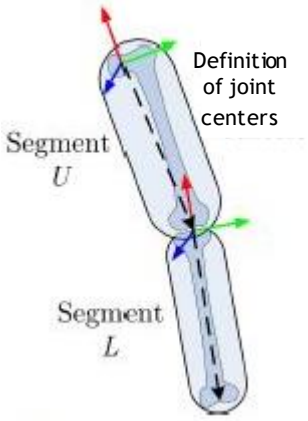

Figure 2.10 - Definition of segment axes and determination of its length. Adapted from [91].

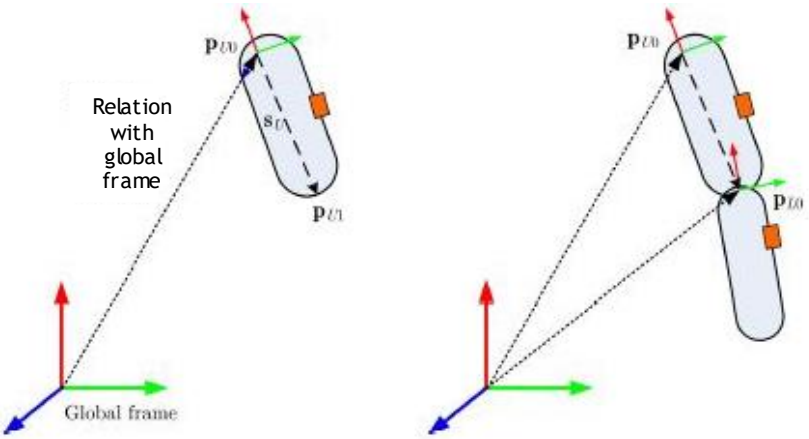

Figure 2.11 - Left: relation of segment with global frame; right: relation of two connecting segments at t=0. Adapted from [91].

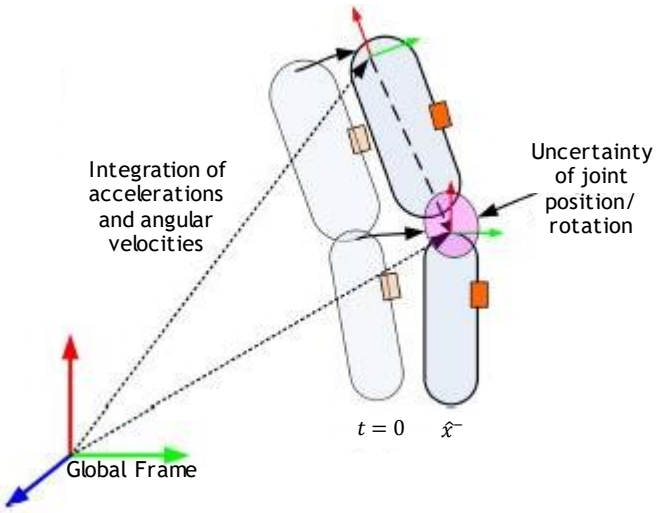

Figure 2.12 - Increasing of uncertainty about the joint position due to integration of acceleration. Adapted from [91].

After each inertial and segment kinematic prediction step, the uncertainty of the joint position and rotation will grow due to sensor noise and movement related errors (see figure 2.12), which will be corrected using the joint measurement updates.

Furthermore, for each joint, the position relation can be obtained as a linearized function:

$$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{w}_t, \tag{2.25}$$

where $\boldsymbol{x}_t$ denotes the state vector at time t, containing the position of segments U and L. $\boldsymbol{C}$ is the measurement matrix relating $\boldsymbol{x}_t$ to $\boldsymbol{y}_t$ and $\boldsymbol{w}_t$ is measurement noise. Then, when two segments are connected, the matrix $\boldsymbol{C}$ is given by:

$$\boldsymbol{C} = [\boldsymbol{I}_3 \quad -\boldsymbol{I}_3], \tag{2.26}$$

in this expression, $\boldsymbol{I}_3$ represents the 3 by 3 identity matrix.

This system relies on a Kalman filter to estimate the state using the joint relation and the state prediction by the segment kinematic integration step:

$$\hat{\boldsymbol{x}}_t^+ = \hat{\boldsymbol{x}}_t^- + K(\boldsymbol{y}_t - C\hat{\boldsymbol{x}}_t^-), \tag{2.27}$$

where $\hat{\boldsymbol{x}}_t^-$ and $\hat{\boldsymbol{x}}_t^+$ are the states before and after the Kalman filter, respectively, and K is the Kalman gain. This gain is computed based on stochastic parameters about positional and rotational characteristics for each joint and propagation of errors by the integration step based on the sensor noise. The Kalman filter will correct the kinematics for drift and the uncertainty of the joint position is reduced (see figure 2.13).

The integration drift of each segment in relation to another segment is eliminated by the assumptions about joints in an articulated body, while the detection of external points on the segment with the global frame (world) is used to limit the boundless integration error of the assembled body model in the global frame. Therefore, under most circumstances, one can assume that the body must be in contact with an external physical world and is subject to gravity.

Joint rotation can be defined as the orientation of a distal segment ${}^{GB}\boldsymbol{q}_L$ with respect to a proximal segment ${}^{GB}\boldsymbol{q}_U$, then:

$${}^{B}\boldsymbol{q}_{UL} = {}^{GB}\boldsymbol{q}_U^* \otimes {}^{GB}\boldsymbol{q}_L \tag{2.28}$$

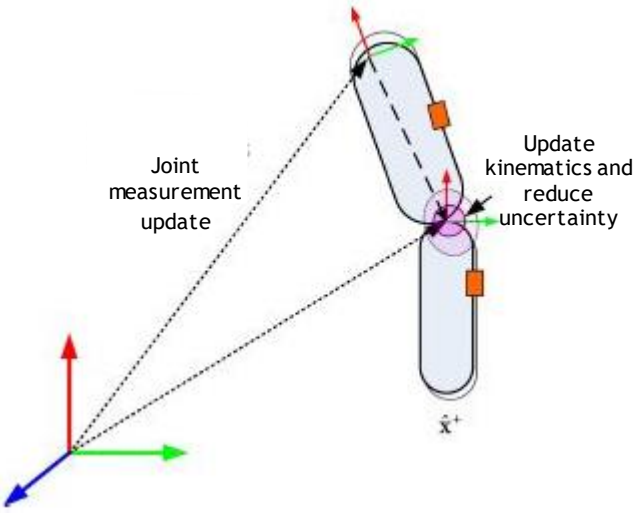

Figure 2.13 - Correction of kinematics and reduction of uncertainty, after joint update. Adapted from [91].

In order to describe joint angles, there are a few parameterizations, such as the Cardan/Euler representation, joint coordinate system and helical angle. All of these representations are based on the same quaternion or rotation matrix, they only differ in how the angles are extracted from this rotation. However, in many cases, joint angles are directly measured with sensors.

2.5-2. Eduardo Palermo, Stefano Rossi, Francesca Marini, Fabrizio Patanè and Paolo Cappa [92]

The aim of this paper was developing a novel two phase functional calibration procedure for lower-limb kinematics evaluation, designed to obtain the body-to-sensor alignment independently for each sensor and without requiring a skilled experimenter.

The estimation of joint angles consists in the evaluation of joint rotations between two body segments and, therefore, in the calculation of joint rotation matrices. The rotation matrix ${}^{b_i}\boldsymbol{R}_{b_j}$ between two coordinate systems (CS_{b_i} and CS_{b_j}) relative to the body frames b_i and b_j can be computed as:

$$ {}^{b_i}\boldsymbol{R}_{b_j} = ({}^{\mathrm{g}}\boldsymbol{R}_{b_i})^T \, {}^{\mathrm{g}}\boldsymbol{R}_{b_j}, \tag{2.29} $$

where the CS_{g} is the ground fixed coordinate system. Introducing the coordinate systems CS_{S_i}, associated to the $i\text{-}th$ sensor:

$$ {}^{\mathrm{g}}\boldsymbol{R}_{b_i} = {}^{\mathrm{g}}\boldsymbol{R}_{S_i} \, {}^{S_i}\boldsymbol{R}_{b_i}, \tag{2.30} $$

where ${}^{\mathrm{g}}\boldsymbol{R}_{S_i}$ represents the output of the $i\text{-}th$ inertial sensor placed on the $i\text{-}th$ body segment and ${}^{S_i}\boldsymbol{R}_{b_i}$ is the related body-to-sensor rotation matrix. The rotation matrix of the joint between $i\text{-}th$ and $j\text{-}th$ body segment is, therefore, equal to:

$$ {}^{b_i}\boldsymbol{R}_{b_j} = ({}^{\mathrm{g}}\boldsymbol{R}_{S_i} \, {}^{S_i}\boldsymbol{R}_{b_i})^T \, {}^{g}\boldsymbol{R}_{S_j} \, {}^{S_j}\boldsymbol{R}_{b_j} \tag{2.31} $$

The ${}^{S_i}\boldsymbol{R}_{b_i}$ matrix depends on the biomechanical convention chosen to define the CS_{b_i}. In the present work, CS_{b_i} is defined, while the subject is in standing position, as following: $\boldsymbol{z}_{b_i}$ axis coincident with the vertical one $\boldsymbol{z}_g$ and the plane yz_{b_i} parallel to the sagittal plane with $\boldsymbol{y}_{b_i}$ pointing forward. This coordinate system is not an anatomical frame, but a technical frame also referred to in the following as TF. The paper refers that TF can be evaluated by means of both IMU with magnetometer, or MIMUs, (TF_MIMU) and an optoelectronic system (TF_OS), which represents the reference technical frame. For the purpose of the present work, only TF_MIMU will be presented.

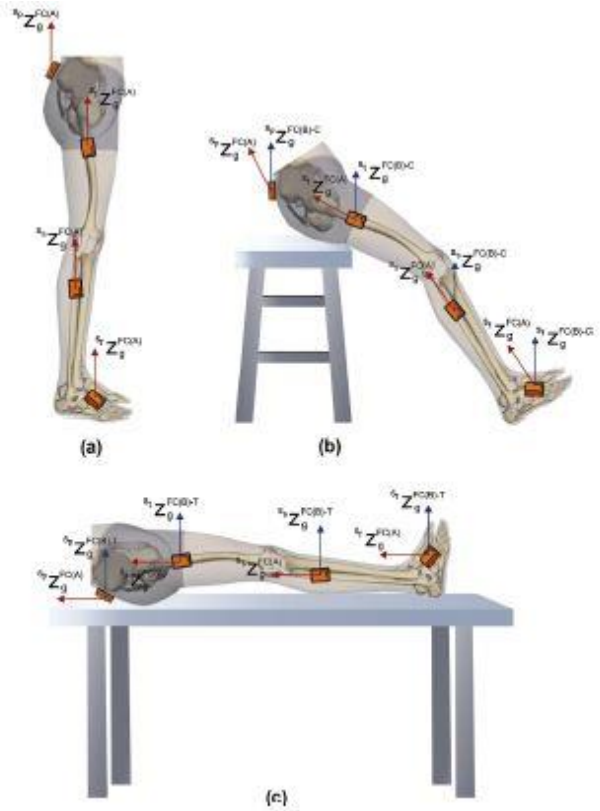

Figure 2.14 - Functional calibration procedure, FC. Vertical axis gathered in: (a) standing upright posture, FC(A); (b) sitting position while the trunk backwards inclined and the legs stretched, FC(B)-C; (c) lying on the table, FC(B)-T.

The TF_MIMU can be evaluated by means of a functional calibration procedure (FC) that consists in the gathering of the sensor outputs for five seconds with the subject keeping still during two consecutive phases (see figure 2.14). Phase A (FC(A)), is conducted while the subject is in a standing upright posture. Phase B (FC(B)), can be carried out while the subject is in a sitting position with the trunk backwards inclined and the legs stretched, addressed as FC(B)-C, or alternatively, while the subject is lying on a table, addressed as FC(B)-T. Thus, two alternatives for phase B to permit the FC conduction for patients with different pathology severity are proposed. Also it was hypothesize that irrelevant differences in the body-to-sensor rotation estimation are obtained by the combination of FC(A) & FC(B)-C or FC(A) & FC(B)-T.

The subject was asked only to maintain the sagittal planes of each body segment parallel between FC(A) and FC(B), avoiding rotations of body segments in frontal and transverse planes. In order to make the procedure easy to be performed by the subject autonomously, additional tools and/or procedures devoted to limit rotations of body segments in frontal and transverse planes were avoided, despite it represents a possible source of uncertainty in the evaluation of body-to-sensor rotation matrices. Thus, one of the aims of this study was the evaluation of accuracy and repeatability of the procedure, including also the above mentioned source of uncertainty.

In FC(A) the z-axis ${}^{S_i}\boldsymbol{z}_{b_i}$ of $i\text{-}th$ body frame is defined parallel to the vertical axis $\boldsymbol{z}_g$, which is coincident to $\boldsymbol{z}_{g_i}$ measured by the MIMU sensor in the local frame:

$$^{S_i}\boldsymbol{z}_{b_i} = {}^{S_i}\boldsymbol{z}_{g_i} \tag{2.32}$$

In FC(B), the sagittal plane (yz-plane) is defined parallel to $^{S_i}\boldsymbol{z}_{b_i}$ and $\boldsymbol{z}_{g_i}$, with $\boldsymbol{y}_{g_i}$ pointing forward, which is again measured by the MIMU sensor locally:

$$^{S_i}\boldsymbol{x}_{b_i} = \frac{^{S_i}\boldsymbol{z}_{b_i} \times {}^{S_i}\boldsymbol{z}_{g_i}}{|^{S_i}\boldsymbol{z}_{b_i} \times {}^{S_i}\boldsymbol{z}_{g_i}|}, \quad ^{S_i}\boldsymbol{y}_{b_i} = {}^{S_i}\boldsymbol{z}_{b_i} \times {}^{S_i}\boldsymbol{x}_{b_i} \tag{2.33}$$

Finally, the rotation matrix $^{S_i}\boldsymbol{R}_{b_i}$ is obtained grouping the three unit vectors:

$$^{S_i}\boldsymbol{R}_{b_i} = \left[{}^{S_i}\boldsymbol{x}_{b_i} \quad {}^{S_i}\boldsymbol{y}_{b_i} \quad {}^{S_i}\boldsymbol{z}_{b_i}\right] \tag{2.34}$$

2.5-3. H.J. Luinge, P.H. Veltink and C.T.M. Baten [93]

This algorithm aims to determinate a method of the orientation of the 2 arm segments using inertial sensors and anatomical elbow constrains.

The orientation of the IMU coordinate frame S to the segment coordinate frame F is given by a rotation matrix containing the 3 unit vectors of the forearm:

$$^{SF}\boldsymbol{R} = \left[\ {}^{S}\boldsymbol{x}^{F} \quad {}^{S}\boldsymbol{y}^{F} \quad {}^{S}\boldsymbol{z}^{F}\right] \tag{2.35}$$

In this case, the coordinate system, in which the vector is expressed, is indicated by the left superscript and the segment under analysis is given by the right superscript. U represents upperarm segment and F is used for forearm.

The angular velocity during pronation ($\boldsymbol{\omega}_{Pron}$) determinates the direction of the forearm y-axis, or the opposite direction of the angular velocity during supination ($\boldsymbol{\omega}_{Sup}$):

$$^{S}\boldsymbol{y}^{F} = \frac{\boldsymbol{\omega}_{Pron}}{|\boldsymbol{\omega}_{Pron}|} = -\frac{\boldsymbol{\omega}_{Pron}}{\boldsymbol{\omega}_{Sup}} \tag{2.36}$$

Similarly, the z-axis can be obtained by measuring the direction of gravity at the start and the end of trial:

$$^{S}\boldsymbol{z}^{F-} = \frac{-\boldsymbol{g}_{start}}{|\boldsymbol{g}_{start}|} \tag{2.37}$$

The minus sign is used to represent a first guess, as the z-axis is later recomputed. Moreover, the x-axis can be found by making an orthogonal coordinated system. Since $\boldsymbol{y}$ and $\boldsymbol{z}$ of the segment are defined by measurements, orthogonality may not be obtained due to errors

in measurements. The direction of z-axis is difficult to measure because it is hard to keep the forearm horizontal, therefore, this axis is recomputed using the y and x axes:

$$^{SF}\boldsymbol{R} = \left[{}^{S}\boldsymbol{y}^{F} \times {}^{S}\boldsymbol{z}^{F-} \quad {}^{S}\boldsymbol{y}^{F} \quad ({}^{S}\boldsymbol{y}^{F} \times {}^{S}\boldsymbol{z}^{F-}) \times {}^{S}\boldsymbol{y}^{F-} \right] \tag{2.38}$$

The orientation of the sensor with respect to the upper-arm was found with 2 movements. First, placing the elbow on the top of a table and performing endorotation/exorotation movement, assuming that the y-axis is the rotation axis. Then, flexing the elbow 90° and abduct the upper-arm while kipping the elbow fixed. The direction of rotation gives the z-axis.

The procedure used to compute the orientation of the upper-arm with respect to the IMU is the same as for the forearm, except for the determination of the z-axis of the segment. The direction of the z-axis can be found using the gravity at the start and end of the abduction movement:

$$^{S}\boldsymbol{z}^{U-} = \frac{{}^{S}\boldsymbol{g}_{Start} \times {}^{S}\boldsymbol{g}_{End}}{|{}^{S}\boldsymbol{g}_{Start} \times {}^{S}\boldsymbol{g}_{End}|} \tag{2.39}$$

The gravity vector was measured using the 3D accelerometer.

It is assumed that the coordinate systems that are identified by the segment calibration can be used to describe the constraint axis inherent to joint DOF. Here it was assumed that the y-axis of the forearm will always be in the zy-plane of the upper arm. The adduction angle γ is here defined as the angle between the x-axis of the upper-arm and the y-axis of the forearm $-90°$. In radians this can be approximated using the dot product:

$$\gamma = \boldsymbol{x}^{U} \bullet \boldsymbol{y}^{F} \tag{2.40}$$

An estimate of $\boldsymbol{x}^{U}$ can be obtained by taking the first column of the upper arm rotation matrix. Likewise $\boldsymbol{y}^{F}$ is the second column of the forearm rotation matrix.

A least-squares filter was designed to use a constraint forcing the adduction angle to be zero. This enforcement allows to improve the orientation estimation generated using only gyroscopes and accelerometers. Each time step, the orientation of the upper-arm and the forearm is estimated using gyroscopes, accelerometers and the previous orientations according to [94] yielding 2 orientation estimates $^{GU}\widehat{\boldsymbol{R}}_{t}^{-}$ and $^{GF}\widehat{\boldsymbol{R}}_{t}^{-}$ with their variances given by error covariance matrices $\boldsymbol{Q}_{\theta,t}^{U}$ and $\boldsymbol{Q}_{\theta,t}^{F}$, respectively. The least-squares filter estimates the orientation errors $^{G}\theta^{F}$ and and $^{G}\theta^{U}$ in a way that sets the adduction angle to zero. Finally, the estimated orientation error is used to correct the orientations $^{GU}\widehat{\boldsymbol{R}}_{t}^{-}$ and $^{GF}\widehat{\boldsymbol{R}}_{t}^{-}$ to obtain $^{GU}\widehat{\boldsymbol{R}}_{t}^{+}$ and $^{GF}\widehat{\boldsymbol{R}}_{t}^{+}$, the input of the next step.

In order for the least-squares filter to correct the orientation in a way that sets the adduction angle to zero, a function will be derived that relates the orientation error to the

adduction angle. An orientation is described by a rotation matrix. The orientation error is expressed using θ, which has the direction and smallest magnitude that the real orientation of a segment has to rotate in order to coincide with the estimated orientation. For small angles an error of the unit x-axis of the upper-arm can be described using the cross product [77]:

$$\boldsymbol{x} = \hat{\boldsymbol{x}} - \hat{\boldsymbol{x}} \times \theta \tag{2.41}$$

Using the relation:

$$\boldsymbol{x} \times \theta \bullet \boldsymbol{y} = \boldsymbol{y} \bullet \boldsymbol{x} \times \theta = \boldsymbol{y} \times \boldsymbol{x} \bullet \theta = (\boldsymbol{y} \times \boldsymbol{x})^T . \theta, \tag{2.42}$$

and neglecting products of errors, the relation was found describing the estimated adduction $\hat{\gamma}$ as a function of the real adduction γ and orientation errors:

$$\begin{aligned} \hat{\gamma} = {}^G\hat{\boldsymbol{x}}^U \bullet {}^G\hat{\boldsymbol{y}}^F &= \left({}^G\boldsymbol{x}^U + {}^G\hat{\boldsymbol{x}}^U \times {}^G\theta^U\right) \bullet \left({}^G\boldsymbol{y}^F + {}^G\hat{\boldsymbol{y}}^F \times {}^G\theta^F\right) \\ &= \gamma + {}^G\hat{\boldsymbol{x}}^U \bullet {}^G\hat{\boldsymbol{y}}^F \times {}^G\theta^F + {}^G\hat{\boldsymbol{x}}^U \times {}^G\theta^U \bullet {}^G\hat{\boldsymbol{y}}^F \\ &= \gamma + \left({}^G\hat{\boldsymbol{x}}^U \times {}^G\hat{\boldsymbol{y}}^F\right)^T . {}^G\theta^F + \left({}^G\hat{\boldsymbol{y}}^F \times {}^G\hat{\boldsymbol{x}}^U\right)^T . {}^G\theta^U \end{aligned} \tag{2.43}$$

A small dot is used to describe a matrix multiplication and a larger dot to indicate the dot product. To obtain the orientation errors using a linear least-squares technique, the above equation was written as a matrix multiplication and the real g was set to zero:

$$\hat{\gamma} = \left[({}^G\hat{\boldsymbol{x}}^U \times {}^G\hat{\boldsymbol{y}}^F)^T ({}^G\hat{\boldsymbol{x}}^U \times {}^G\hat{\boldsymbol{y}}^F)^T\right] \begin{Bmatrix} {}^G\theta^F \\ {}^G\theta^U \end{Bmatrix} = H . \begin{Bmatrix} {}^G\theta^F \\ {}^G\theta^U \end{Bmatrix} \tag{2.44}$$

According to [95], the optimal estimate of such an equation can be obtained by:

$$\begin{Bmatrix} {}^G\theta^F \\ {}^G\theta^U \end{Bmatrix} = K . \hat{\gamma}, \tag{2.45}$$

where K is defined as:

$$K = \boldsymbol{Q} . \boldsymbol{H}^T . [\boldsymbol{H} . \boldsymbol{Q} . \boldsymbol{H}^T + \boldsymbol{R}]^{-1} \tag{2.46}$$

$\boldsymbol{H}$ and $\hat{\gamma}$ can be entirely calculated using ${}^G\hat{\boldsymbol{x}}^U$ and ${}^G\hat{\boldsymbol{y}}^F$, obtained from from the a priory orientation estimates. $\boldsymbol{R}$ is the variance of the adduction angle, a measure of the error that is made by the assumption that the adduction angle is zero. $\boldsymbol{Q}$ is the covariance matrix describing the covariances of the a priori estimated orientation errors:

$$\boldsymbol{Q} = \begin{bmatrix} Q_{\theta,t}^{U} & 0 \\ 0 & Q_{\theta,t}^{F} \end{bmatrix} \tag{2.47}$$

The estimated orientation errors were expressed as a rotation matrices, found in [77], and used to correct the orientation.

2.5-4. Manon Kok, Jeroen D. Hol and Thomas B. Schön [96]

This paper introduces an optimization-based approach for inertial motion capture. The problem of estimating the relative position and orientation of each body segment is formulated as a constrained estimation problem. Given N measurements $\boldsymbol{y}_{1:N} = \{y_1, \dots, y_N\}$, a point estimate of the variables z can be obtained as a constrained maximum a posteriori (MAP) estimate, maximizing the posterior density function:

$$\begin{aligned} &max_z \;\; P(z|y_{1:N}) \\ &s.t. \;\; c_e(z) = 0, \end{aligned} \tag{2.48}$$

where $c_e(z)$ represents the equality constraints. In this paper, z consists of both static parameters θ and time-varying variables $x_{1:N}$. Using this together with the Markov property of the time-varying variables and the fact that the logarithm is a monotonic function, we can rewrite the above equation as:

$$\begin{aligned} &min_{z=\{x_{1:N},\theta\}} \underbrace{-logp(x_1|y_1) - logp(\theta)}_{initialization} \\ &\underbrace{-\sum_{t=2}^{N} logp(x_t|x_{t-1},\theta)}_{dynamic\ model} \underbrace{-\sum_{t=1}^{N} logp(y_t|x_t,\theta)}_{biomechanical/sensor\ model} \end{aligned} \tag{2.49}$$

$$s.t. \; c_{bio}(z) = 0 \tag{2.50}$$

Relevant coordinate frames are the: local coordinate frame L, which is aligned with the local gravity vector, with the z-axis pointing up; body segment coordinate frame B fixed to the bone in the body segment B_j and its origin can be anywhere, usually it's the center of rotation of a joint; sensor coordinate frame S_i of the moving IMU S_i and its origin is located in the center of the accelerometer triad.

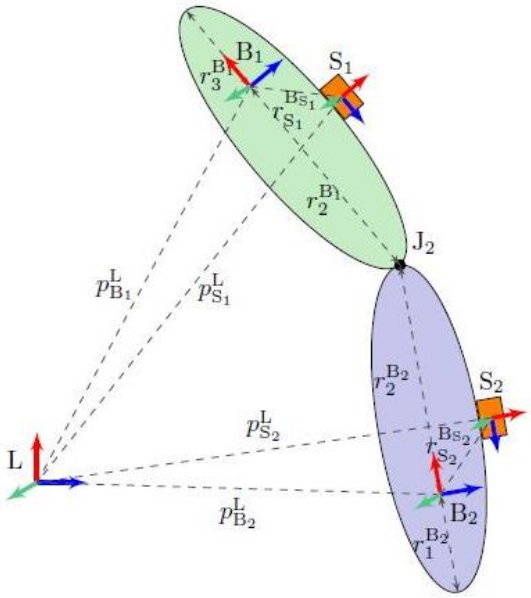

Figure 2.15 - Definition of variables and coordinated frames. Adapted from[96].

In setting up the optimization problem, the first step was to define the set of sensors $\mathcal{S}$, the set of body segments $\mathcal{B}$ and the set of joints $\mathcal{J}$ in the problem. Each inertial sensor needs to be mounted on the body, and sensor S_i is assumed to be placed on body segment B_{S_i}. The distance vector $\boldsymbol{r}_{S_i}^{B_i}$ and orientation $\boldsymbol{q}^{B_{S_i}S_i}$ of sensor S_i with respect to body segment B_{S_i} are without loss of generality assumed to be known from calibration.

Previous knowledge of the human body can be used to identify which body segments are connected by which joints. To express the location of the joint in the body frames of the connected body segments, the distance vectors $\boldsymbol{r}_k^{B_j}$ from the body frame B_k to joint k, need to be defined for all joints $J_k \in \mathcal{J}$ and all $B_j \in \mathcal{B}_{J,k}$. It was assumed that they are known from calibration. Here, all joints are assumed to be ball-and-socket joints, but it was incorporated additional knowledge about a subset of the joints, denoted by $\mathcal{H}$, which we assume to be hinge joints.

Furthermore, it was defined the set of time steps in the optimization as $\mathcal{T}$ rather than explicitly summing over all time steps $t = 1 \dots N$. The variables in the optimization problem are then given by, with respect to local frame L for $\forall t \in \mathcal{T}$:

- The position $\boldsymbol{p}_{S_{i,t}}^{L}$ and velocity $\boldsymbol{v}_{S_{i,t}}^{L}$ of sensor S_i, $\forall S_i \in \mathcal{S}$
- The orientation $\boldsymbol{q}_t^{LS_i}$ of sensor S_i, $\forall S_i \in \mathcal{S}$
- The position $\boldsymbol{p}_{B_{j,t}}^{L}$ of the body segment B_j, $\forall B_j \in \mathcal{B}$
- The orientation $\boldsymbol{q}_t^{LB_j}$ of body segment B_j, $\forall B_j \in \mathcal{B}$

Defining the number of sensors as N_S and the number of body segments as N_B, the number of variables in the optimization problem is $z \in \mathbb{R}^{(9N_S+6N_B+3)N+3N_S}$. When the optimization problem was solved, it was encoded the rotation states using a three-dimensional state vector. Throughout the paper, typically, interchangeably make use of the unit quaternion $\boldsymbol{q}^{LS}$ and the rotation matrix $\boldsymbol{R}^{LS}$ as representations of the orientation. The quaternion conjugate,

representing the inverse rotation will be represented by $(\boldsymbol{q}^{LS})^c = \boldsymbol{q}^{SL}$. Similarly for the rotation matrix, $(\boldsymbol{R}^{LS})^T = \boldsymbol{R}^{SL}$.

Based on the biomechanical model it is possible to derive relations between the different variables, which can be categorized in three classes:

i. Joints between the body segments

The constrains $\boldsymbol{c}_{bio}(z)$, originate from a biomechanical model, enforce the body segments to be connected at the joint locations at all times:

$$\boldsymbol{c}_{bio}(z) = \boldsymbol{p}^{L}_{B_m,t} + \boldsymbol{R}_t^{LB_m}\boldsymbol{r}_k^{B_m} - \boldsymbol{p}^{L}_{B_n,t} - \boldsymbol{R}_t^{LB_n}\boldsymbol{r}_k^{B_n}, \quad \{B_n, B_m\} \in \mathcal{B}_{J_k}, \tag{2.51}$$

for all $J_k \in \mathcal{J}$ and $t \in \mathcal{T}$. Thus, will result in N_J(number of joints) constrains at each time step t.

ii. Placement of the sensors on the body segments

The position and orientation of the sensor can be expressed in terms of its position and orientation on body segment. Since it is impossible to place the sensor directly on the bone, it has to be placed in soft tissue, resulting in slightly movements with respect to the bone. Therefore, the position and orientation of sensor S_i on the body segment B_{S_i} can be obtained by:

$$\boldsymbol{p}^{L}_{S_i,t} = \boldsymbol{p}^{L}_{B_i,t} + \boldsymbol{R}_t^{LB_{S_i}}(\boldsymbol{r}^{B_{S_i}}_{S_i} + \boldsymbol{e}^{B_{S_i}}_{p,t}), \tag{2.52}$$

$$\boldsymbol{q}_t^{LS_i} = \boldsymbol{q}_t^{LB_{S_i}}\boldsymbol{q}^{B_{S_i}S_i}\exp(\tfrac{1}{2}\boldsymbol{e}^{S_i}_{q,t}), \tag{2.53}$$

where $\boldsymbol{e}^{B_{S_i}}_{p,t} \sim \mathcal{N}(0, \Sigma_p)$ and $\boldsymbol{e}^{S_i}_{q,t} \sim \mathcal{N}(0, \Sigma_q)$.

iii. Rotational freedom of the joints

Some joints are limited to one or two axes, for example, the knee. Minimizing:

$$\boldsymbol{e}_{J_{k,t}} = \begin{bmatrix}\boldsymbol{n}_1^T \\ \boldsymbol{n}_2^T\end{bmatrix}(\boldsymbol{R}_t^{LB_m})^T\boldsymbol{R}_t^{LB_n}\boldsymbol{n}_2, \quad \{B_n, B_m\} \in \mathcal{B}_{J_k}, \tag{2.54}$$

where $\boldsymbol{n}_1, \boldsymbol{n}_2$ and $\boldsymbol{n}_3$ represent the different axis directions and $\boldsymbol{e}_{J_{k,t}} \sim \mathcal{N}(0, \Sigma_k)$, will minimize the rotation around any but $\boldsymbol{n}_2$-axis.

The sensor's position, velocity and orientation at each time instance can be related by a dynamic model in which the accelerometer and gyroscope measurements are used as inputs. In this work a slightly different approach was implemented to reduce the number of variables in the optimization problem. To achieve high update rates using a relatively small number of variables, an approach similar to the one discussed by [97] was used. Hence, strapdown inertial integration, in which the accelerometer and gyroscope signals are integrated, is run at high update rates. This leads to accelerometer measurements $\Delta\boldsymbol{p}$ and $\Delta\boldsymbol{v}$ representing a difference in position and velocity and gyroscope measurements $\Delta\boldsymbol{q}$ representing a difference in

orientation. These are integrated for $\frac{T_S}{T}$ times, where T_S is the sampling time of the inertial sensors and T is the sampling time used in the optimization problem.

The position, velocity and orientation of each sensor S_i are related from time t to time $t + T$ using the accelerometer measurements $\Delta \boldsymbol{p}_t^{S_i}$, $\Delta \boldsymbol{v}_t^{S_i}$ and the gyroscope measurements $\Delta \boldsymbol{q}_t^{S_i}$ The position and velocity states at each time step are modeled according to:

$$\boldsymbol{p}_{S_i,t+T}^{L} = \boldsymbol{p}_{S_i,t}^{L} + T\boldsymbol{v}_{S_i,t}^{L} + \boldsymbol{R}_t^{LS_i}\left(\Delta \boldsymbol{p}_t^{S_i} + \Delta \boldsymbol{w}_{p,t}^{S_i}\right) + \frac{T^2}{2}\boldsymbol{g}^{L} \tag{2.55}$$

$$\boldsymbol{v}_{S_i,t+T}^{L} = \boldsymbol{v}_{S_i,t}^{L} + \boldsymbol{R}_t^{LS_i}\left(\Delta \boldsymbol{v}_t^{S_i} + \Delta \boldsymbol{w}_{v,t}^{S_i}\right) + T\boldsymbol{g}^{L} \tag{2.56}$$

where $\Delta \boldsymbol{p}_t^{S_i}$ and $\Delta \boldsymbol{v}_t^{S_i}$ denote the inputs based on the accelerometer measurements. The noise terms are modeled as $\boldsymbol{w}_{p,t} \sim \mathcal{N}(0, Q_V)$. The earth gravity is denoted by $\boldsymbol{g}^{L}$. The orientation states are modeled as:

$$\boldsymbol{q}_{t+T}^{LS_i} = \boldsymbol{q}_t^{LS_i} \Delta \boldsymbol{q}_t^{S_i} \exp\left(\frac{1}{2}\boldsymbol{w}_{q,t}^{S_i}\right), \tag{2.57}$$

where $\Delta \boldsymbol{q}_t^{S_i}$ denotes the gyroscope measurements, corrected for the estimated gyroscope bias, and $\boldsymbol{w}_{q,t}^{S_i} \sim \mathcal{N}(0, Q_q)$.

Since the dynamic model above presented states in terms of their value at the previous time step, the state at the first time instance needs to be treated separately. The orientation $\boldsymbol{q}_1^{LS_i}$ of each sensor S_i is estimated using the first accelerometer and magnetometer sample of that sensor. Note that this is the only place in the algorithm where magnetometer measurements are used. The variables $\boldsymbol{q}_1^{LS_i}$ are then initialized around this estimated orientation with additive noise $\boldsymbol{e}_{1_1}^{S_i} \sim \mathcal{N}(0, \Sigma_{q_1})$. The position $\boldsymbol{p}_{S_{i,1}}^{L}$ of one of the sensors is without loss of generality initialized around zero with additive noise $\boldsymbol{e}_{p_1} \sim \mathcal{N}(0, \Sigma_{q_1})$.This defines the origin of the local coordinate frame L.

2.5-5. Algorithm Analysis

Regarding the algorithm developed by Palermo et al. [92], they proposed a calibration procedure specifically for gait analysis, which may have some restrictions when applied to upper limb segments. Nevertheless, this algorithm was design to be easy for the subject to perform autonomously. In fact, the calibration protocol needs neither a precise sensor positioning, nor the performance of specific and accurate movements. This is an advantage when applied to motor rehabilitation performed, for example at home, by individuals with low mobility.

Both H.J. Luinge et al. [93] and Daniel Roetenberg et al. [91] described a method that required a set of movements by the individuals, in order to calibrate the sensors. This can present a drawback, because healthy can easily perform the supposed movements, in contrast,

individuals with motor disabilities may present difficulties performing those movements, resulting in a poor calibration.

Finally, Schön et al. [98] described an optimization approach to inertial human body motion capture. This method is capable of estimating the relative position and orientation of the body segments. Their experimental results show that the algorithm works well, quickly converging to a feasible solution and resulting in drift-free joint angle estimates which match the joint angles from an optical reference system.

Chapter 3

Joint Axis Estimation

A major problem in IMU-based human motion analysis is that the local coordinate axes of the IMUs are not aligned with any physiologically axes (figure 3.1). In some publications this issue is not addressed, it is assumed that the IMUs can be mounted precisely in a predefined orientation toward the joint [99]. In the more realistic case of arbitrary mounting orientation, it is essential to identify the sensor-to-segment orientation placement of the sensors attached to both ends of the joint. As illustrated in Figure 3.11, these mounting orientations are characterized by the local coordinates of the joint axis. It can be measured manually, but in three-dimensional space, this is a troublesome task that yields low accuracy results [93].

Calibration postures and/or calibration movements are a common method to estimate these vectors. Besides static postures, predefined calibration motions can be used to identify the coordinates of physically axes sensor coordinate system [99]. The protocol used by Roetenberg et al.[91] solves a closed kinematic chain to refine joint axis and position coordinates that have been acquired from a combination of calibration postures, predefined motion and manual measurements of body dimensions. However, both in calibration postures and calibration motions, the accuracy is limited by the precision with which the subject can perform the postures or motions.

Therefore, there is high demand for methods that enable accurate estimation of joint axes with respect to the local sensor frames. In his contribution, Seel et al. [100] demonstrate how this information can be extracted from the measured data of almost arbitrary movements by exploiting the kinematic constraints of the respective joints.

This chapter describes the implemented algorithm for joint axis estimation, based on the work developed by Seel, for knee joint. Thus, the algorithm is explained in the following sections, as well as all changes, adjustments and enhancements included in the scope of this thesis. The complete methodology described in this chapter was implemented in Python.

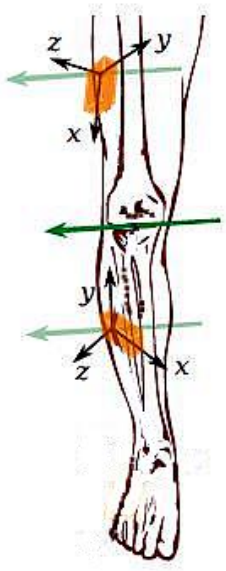

Figure 3.1 - Arbitrary placement of the inertial sensors on the human segments. The coordinates of the joint axis direction (green arrows) in the local coordinate system of the sensors (labels $x - y - z$) characterize the sensor-to-segment mounting.

3.1- Constrains induced by hinge joints

First, let's consider two rigid segments that are connected by a hinge joint and are free to rotate and move in space (figure 3.2). These segments are called first and second segment, and each of them shall be equipped with a three-axial gyroscope attached to the segment in some unknown arbitrary orientation. The unit joint axis vector with respect to the local coordinate system of the gyroscope attached to the first and second segment shall be referred to as $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$, respectively. Furthermore, let the angular velocities measured by the gyroscopes in the coordinates of their local frames be $\boldsymbol{g}_1(t)$ and $\boldsymbol{g}_2(t)$ for the first and second segment, respectively. Then, it is a geometrical fact that $\boldsymbol{g}_1(t)$ and $\boldsymbol{g}_2(t)$ differ only by the joint angle velocity and a (time-variant) rotation matrix [100]. Consequently, their projections into the joint plane, the one which the joint axis is the normal vector, have the same lengths for each instant in time:

$$||\boldsymbol{g}_1(t) \times \boldsymbol{j}_1||_2 - ||\boldsymbol{g}_2(t) \times \boldsymbol{j}_2||_2 = 0 \ \forall t, \tag{3.1}$$

where $||.||_2$ denotes the Euclidean norm. This constrain holds for every moment in time regardless of where and in which orientation the sensors are mounted on the segments [100].

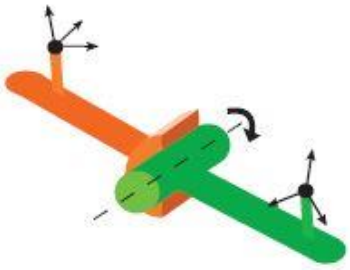

Figure 3.2 - Representation of two rigid segments that are connected by a hinge joint, each one equipped with a three-axial gyroscope (represented by its local coordinate system). The orientations of the sensors toward their segments are assumed to be unknown [100].

This fact turns out to be very useful when identifying the hinge joint axis in case the orientation of the sensors toward the segments is unknown. A large set of measured gyroscopic data is acquired from both sensors and search for the joint axis coordinates that satisfy equation (3.1) for all time instants. More precisely, consider the joint axis candidates $\hat{\boldsymbol{j}}_1$ and $\hat{\boldsymbol{j}}_2$ in the spherical coordinates $\phi_1, \phi_2 \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, $\theta_1, \theta_2 \in [0, 2\pi]$ with:

$$\hat{\boldsymbol{j}}_1 = (\cos(\phi_1)\cos(\theta_1), \cos(\phi_1)\sin(\theta_1), \sin(\phi_1))^T, \tag{3.2}$$

$$\hat{\boldsymbol{j}}_2 = (\cos(\phi_2)\cos(\theta_2), \cos(\phi_2)\sin(\theta_2), \sin(\phi_2))^T, \tag{3.3}$$

where ϕ_i and θ_i are the inclination and azimuth (figure 3.3), respectively, of $\hat{j}_i$ in the i^{th} sensor coordinate system (i = 1, 2).

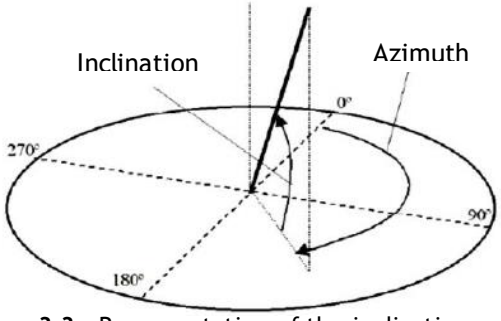

Figure 3.3 - Representation of the inclination and azimuth.

Furthermore, in the context of rehabilitation, the x-axis of the sensor can be assumed to always be aligned with the bone, that is, θ_1, θ_2 are always $\frac{\pi}{2}$, since the sensors are attached with elastic straps and is assume that they do not move along the exercise. Thus, one can re-write equations (3.2) and (3.3):

$$\hat{\boldsymbol{j}}_1 = (0, \cos(\phi_1), \sin(\phi_1))^T, \tag{3.4}$$

$$\hat{\boldsymbol{j}}_2 = (0, \cos(\phi_2), \sin(\phi_2))^T. \tag{3.5}$$

In these reduced coordinates, the error can be defined by:

$$e = ||\boldsymbol{g}_1(t_0 + kt_s) \times \hat{\boldsymbol{j}}_1||_2 - \left|\left|\boldsymbol{g}_2(t_0 + kt_s) \times \hat{\boldsymbol{j}}_2\right|\right|_2, \forall t. \tag{3.6}$$

The proposed approach only yields the true joint axis coordinates when a motion is performed while data is recorded. For instance, if the joint angle remained constant, i.e. the two segments were rigidly connected, then $\boldsymbol{g}_1(t) = \boldsymbol{R}\boldsymbol{g}_2(t), \forall t$, where $\boldsymbol{R}$ is the constant rotation matrix from the second to the first sensor frame. Therefore, (3.1) would hold for any combination $(\boldsymbol{j}_1, \boldsymbol{j}_2)$: $\boldsymbol{j}_1 = \boldsymbol{R}\boldsymbol{j}_2$, no matter what movements the connected segments perform. Thus, motions during which the joint angle remains constant are not suitable for joint axis estimation. Instead, the motion of the segments should be such that the kinematic constrain of the joint axis becomes evident in the measured angular rates.

3.2- Algorithm Implementation

Assuming that N data sets, precisely $\{\boldsymbol{g}_1(t_k), \boldsymbol{g}_2(t_k)\}_{k=1}^{N}, N \gg 4$ were measured. Restricting the joint axis estimates to unit length leads to a two-dimensional problem. Hence, the estimation of the joint axis vector are denote by $\hat{\boldsymbol{j}}_1$ and $\hat{\boldsymbol{j}}_2$ using the parameterization by spherical coordinates, as mentioned above:

$$\xi = [\phi_1, \phi_2], \tag{3.7}$$

$$\begin{aligned} \hat{\boldsymbol{j}}_1 &= [0, \cos(\phi_1), \sin(\phi_1)]^T, \\ \hat{\boldsymbol{j}}_2 &= [0, \cos(\phi_2), \sin(\phi_2)]^T. \end{aligned} \tag{3.8}$$

Therefore, a Gauss-Newton algorithm is implemented and to this end, the following gradients of the left-hand side of equation (3.1) with respect to $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ are derived:

$$\frac{d(\|\boldsymbol{g}_i(t) \times \boldsymbol{j}_i\|_2)}{d\boldsymbol{j}_i} = \frac{(\boldsymbol{g}_i(t) \times \boldsymbol{j}_i \times \boldsymbol{g}_i(t))^T}{\|\boldsymbol{g}_i(t) \times \boldsymbol{j}_i\|_2} \in \mathbb{R}^{1\times 3}, i = 1,2. \tag{3.9}$$

From equation (3.8), the derivatives $\frac{dj_i}{d\xi_i}, i = 1,2$, can be easily obtained, allowing the calculation of the Jacobian matrix

$$\boldsymbol{J}_{axis} := \frac{d\boldsymbol{e}_{axis}}{d\xi_{axis}} \in \mathbb{R}^{N\times 4}, \tag{3.10}$$

of the error vector $\boldsymbol{e}_{axis} \in \mathbb{R}^{N\times 1}$. Its k^{th} entry, which is denoted $\boldsymbol{e}_{axis,k}$, is defined in the equation (3.6). Thus, the k^{th} row of the Jacobian is defined by:

$$\frac{d\boldsymbol{e}_{axis,k}}{d\xi_{axis}} = \frac{(\boldsymbol{g}_1(t) \times \boldsymbol{j}_1 \times \boldsymbol{g}_1(t))^T}{\|\boldsymbol{g}_1(t) \times \boldsymbol{j}_1\|_2} \frac{d\boldsymbol{j}_1}{d\xi_{axis}} - \frac{\left(\boldsymbol{g}_2(t) \times \boldsymbol{j}_2 \times \boldsymbol{g}_2(t)\right)^T}{\|\boldsymbol{g}_2(t) \times \boldsymbol{j}_2\|_2} \frac{d\boldsymbol{j}_2}{d\xi_{axis}} \in \mathbb{R}^{1\times 4}. \quad (3.11)$$

Thus, the Gauss-Newton algorithm is implemented as below. First, random initial values for ξ_{axis} are generated and then, the following update loop is performed repeatedly:

1. Calculation of $\hat{\boldsymbol{j}}_1$ and $\hat{\boldsymbol{j}}_2$ from ξ_{axis} , as in equation 3.8.
2. Calculation of the error vector $\boldsymbol{e}_{axis} \in \mathbb{R}^{N\times 1}$ and the Jacobian $\boldsymbol{J}_{axis}$.
3. Update ξ_{axis} by $\xi_{axis} - (\boldsymbol{J}_{axis}^T \boldsymbol{J}_{axis})^{-1} \boldsymbol{J}_{axis}^T \boldsymbol{e}_{axis}$.
4. Calculation of the tolerance by $\xi_{axis}^{new} - \xi_{axis}^{old} = \vec{\boldsymbol{V}}_{diff}$ and repeat from 1.

Notice that the loop only stops when $\|\vec{\boldsymbol{V}}_{diff}\|$ is lower than a fixed value. In the next chapter, results are obtained and assessed for distinct $\|\vec{\boldsymbol{V}}_{diff}\|$ values.

This proposed methodology, an adaptation of the one presented in [100], is much simpler to implement than other techniques available in the literature. In addition, as already mentioned in [100], this technique allows faster convergences and more accurate results.

Chapter 4

Simulated Data Validation

Before evaluating the previous method in subjects, a kinematic simulation model of the leg was developed. This model consisted of two-segments (thigh and shank) connected by a hinge joint, which represents the knee. Each simulated inertial measurement unit is rigidly attached to the body segment.

This simulation was divided in two major steps:

i. Random placement of the sensors in both segments, within the range $\phi_1, \phi_2 \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$;

ii. Specific orientation of the sensors in both segments.

4.1- Data Simulation Method

In order to validate the algorithm, an exact expression of a movement was required. Thus, the exact expression reproducing a squat movement was deducted, allowing to define for each time instant the exact global position of the joint and the exact vector defining the joint axis. This acquired data is the theoretical exact data. Then, it was possible to assess the robustness and accuracy of the implement algorithm by adding noise to the theoretical exact data and challenging the algorithm to find the correct joint axis.

Thus, assuming that an individual does the squat movement with constant velocity, the supposed movement can be described as a parabola opening to the top (see figure 4.1 left), that has as equation: $s(t) = at^2 + bt + c$. If it is considered that the sensor can read 200 samples per second, then, its rate is 0.005. Therefore, the above expression was used to reproduce the exercise with 1 second of duration. The discrete values of s for each time instant considered were listed in an Excel file.

For a male measuring $1.80\,m$, its thigh, L_1, measures about $0.4428\ m$ and its shank, L_2, $0.4410\ m$, which means both segments have approximately the same length. Thus, an arithmetic mean can be calculated in order to obtain the total length of both segments, yielding

$L = 0.4419\ m$. The amplitude A of the parabola is equal to half of the length of the leg. In order to find the parameters a, b and c, the following equations can be defined:

$$s(0) = 0 = c, \tag{4.1}$$

$$s\left(\frac{t_f}{2}\right) = a\left(\frac{t_f}{2}\right)^2 + b\left(\frac{t_f}{2}\right) + c = A, \tag{4.2}$$

$$s'\left(\frac{t_f}{2}\right) = 2a\left(\frac{t_f}{2}\right) + b = 0, \tag{4.3}$$

where $s(t)$ represents the squat function and $s'(t)$ the corresponding derivative. Now that the parameters are known, the graph for the squat displacement and velocity can be achieved (figure 4.1 right).

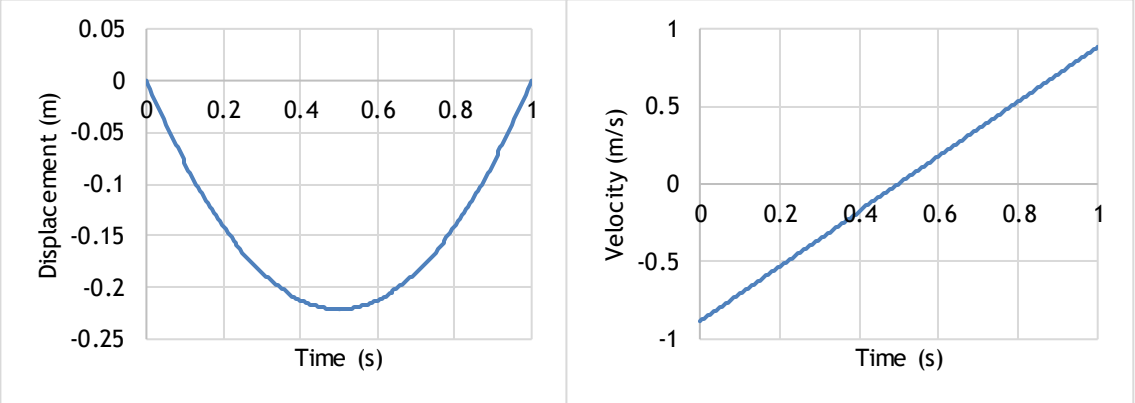

Figure 4.1 - Representation of a squat. On the left is the displacement over time and on the right is the velocity over time.

In order to find the angular velocity of a squat exercise, with the above specifications, the following scheme was drawn.

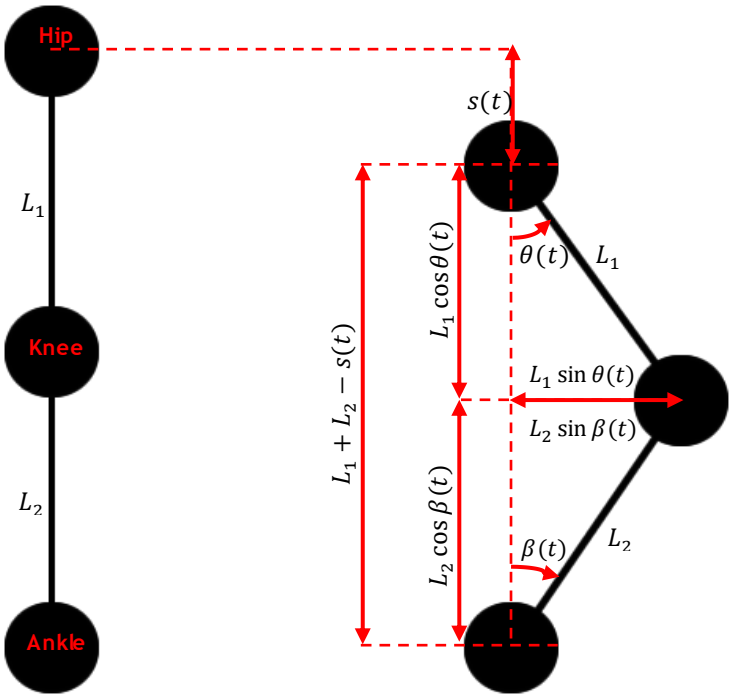

Figure 4.2 - Representation of a squat.

By observing the figure, one can easily perceive that:

$$L_1 \sin\theta(t) = L_2 \sin\beta(t). \tag{4.4}$$

As already mentioned, if it is considered that both segments have approximately the same length ($L_1 = L_2$), then a simplification can be applied:

$$\sin\theta(t) = \sin\beta(t), \tag{4.6}$$

$$\theta(t) = -\beta(t). \tag{4.7}$$

Considering another variable, $\gamma(t) = \theta(t) = -\beta(t)$, to simplify calculations:

$$2L = s(t) + L\cos\gamma(t) + L\cos\gamma(t), \tag{4.8}$$

$$\gamma(t) = \arccos\left(\frac{L - s(t)}{L}\right), \tag{4.9}$$

$$\dot{\gamma}(t) = -\frac{\frac{\dot{s}(t)}{L}}{\sqrt{1 - \left(\frac{L - s(t)}{L}\right)^2}}. \tag{4.10}$$

Thus, one can find the angular velocity associated to the simulated squat exercise, and it is represented in figure 4.3. Finally, in order to obtain the angular velocity in the three directions, the three components of j_1 and j_2 are multiplied (equation 3.4 and 3.5) by $\theta(t)$ and $\beta(t)$, respectively.

In order to make the simulation as close to reality as possible, a random number was added to the acceleration data, between -1 and 1, multiplied by a noise error. Further, to evaluate how the increment of noise influence the results, two different values of noise were considered (10% and 20%).

Moreover, for each of the above steps, three different values for the algorithm tolerance, $\|\vec{V}_{diff}\|$, were assessed: 0.05, 0.01 and 0.001. Furthermore, the orientation of the sensor was fictitiously varied, in order to understand how the algorithm behaves in situation in which both sensor are not perfectly aligned. Thus, two cases were assumed. One in which the initial orientation of each sensor is completely random, and another in which the orientation of each sensor is defined by figure 4.3. In this second case, the orientation of sensor 1 is fixed and the orientation of sensor 2 varies between -90º and 90º, allowing 25 combinations.

A scheme of the considered combinations is represented in figure 4.4 for a better understanding. In figure 4.5 through 4.10, a representation of some of the signals that will feed the algorithm are illustrated, one can find in Annex A the rest of the signals. In these figures, it is possible to observe how the clean exact acceleration of figure 4.3 becomes erratic with the introduced error (10% or 20%), reproducing a real case scenario. Note that the term *sensor 1* and *sensor 2* are used to describe the sensor placed in the thigh and shank, respectively.

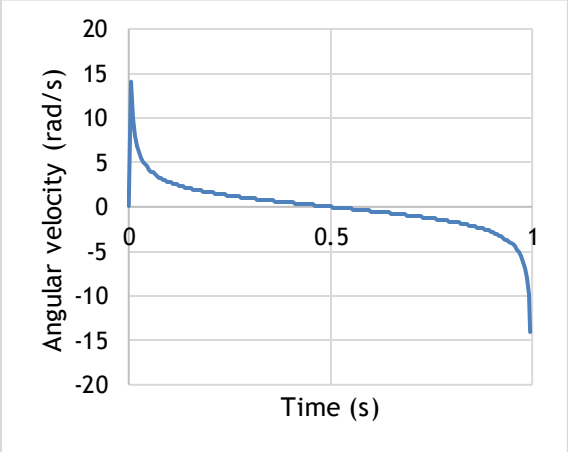

Figure 4.3 - Angular velocity of the simulated exercise.

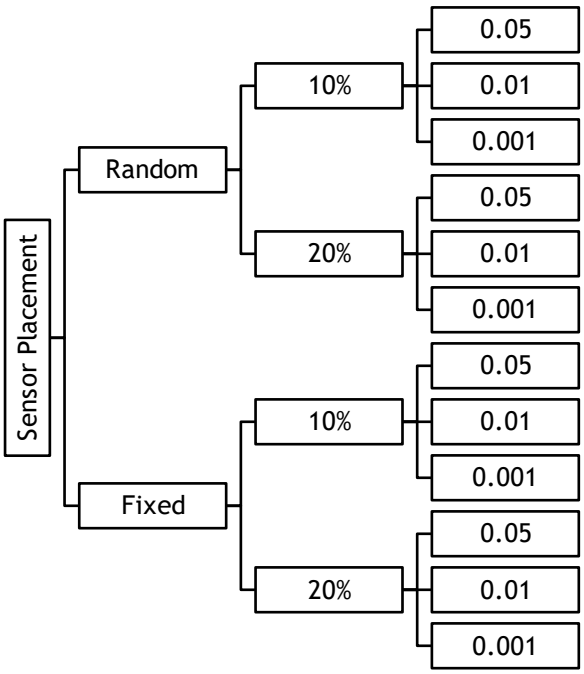

Figure 4.4 - Illustration of the different evaluations performed with the simulated data. It should be noted that 10% and 20% correspond to the added signal noise to data and the next column corresponds to the algorithm tolerance.

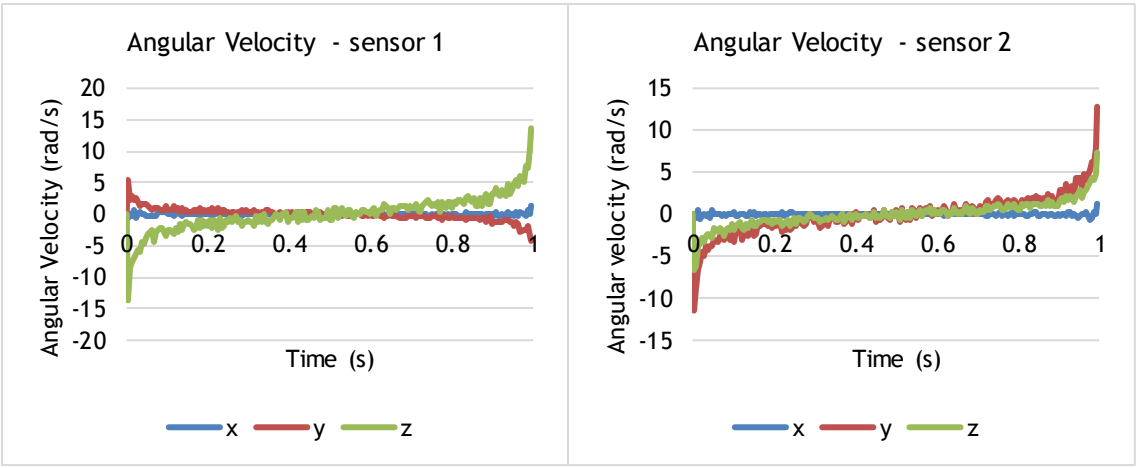

Figure 4.5 - Input signal for random placement of the sensors, with 10% noise.

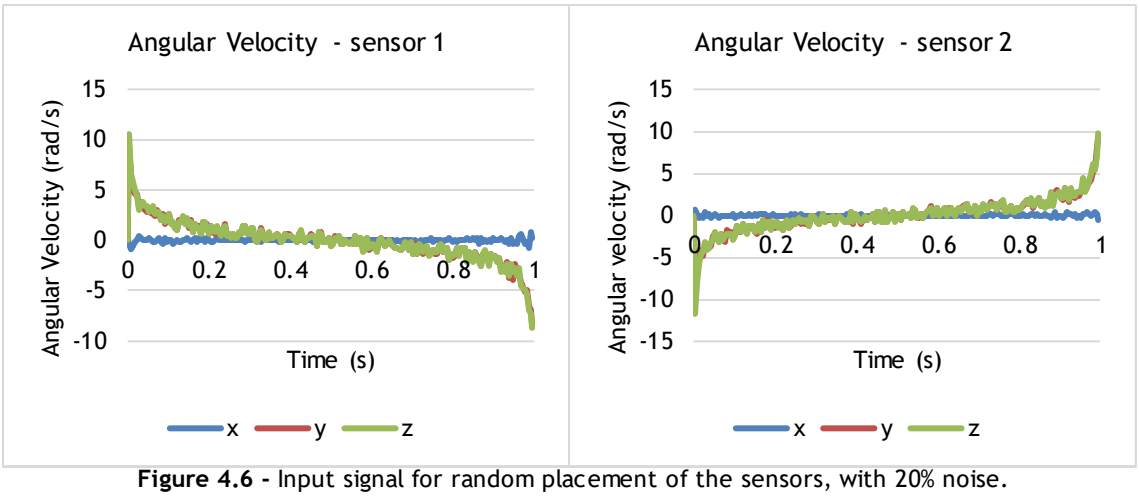

Figure 4.6 - Input signal for random placement of the sensors, with 20% noise.

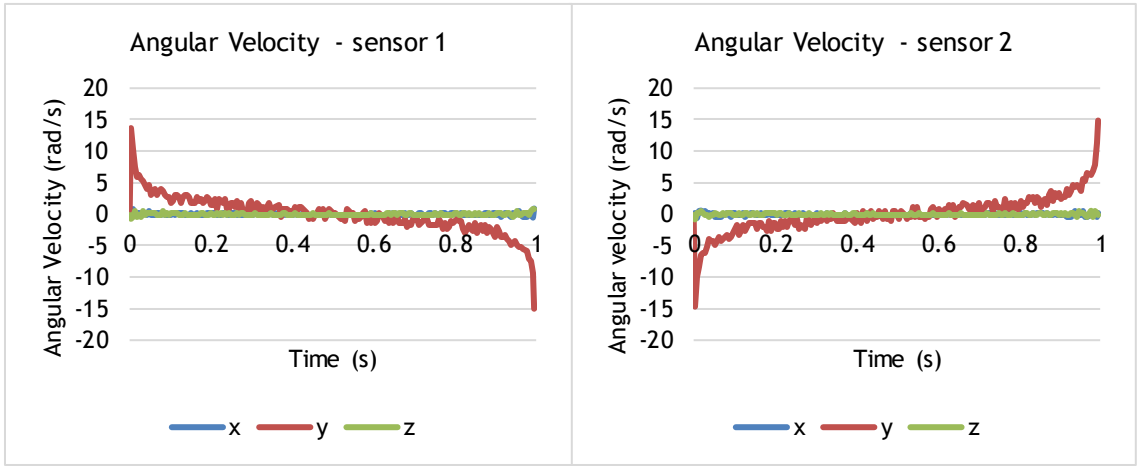

Figure 4.7- Input signal for sensor1 and sensor2 placed at 0°, with 10% noise.

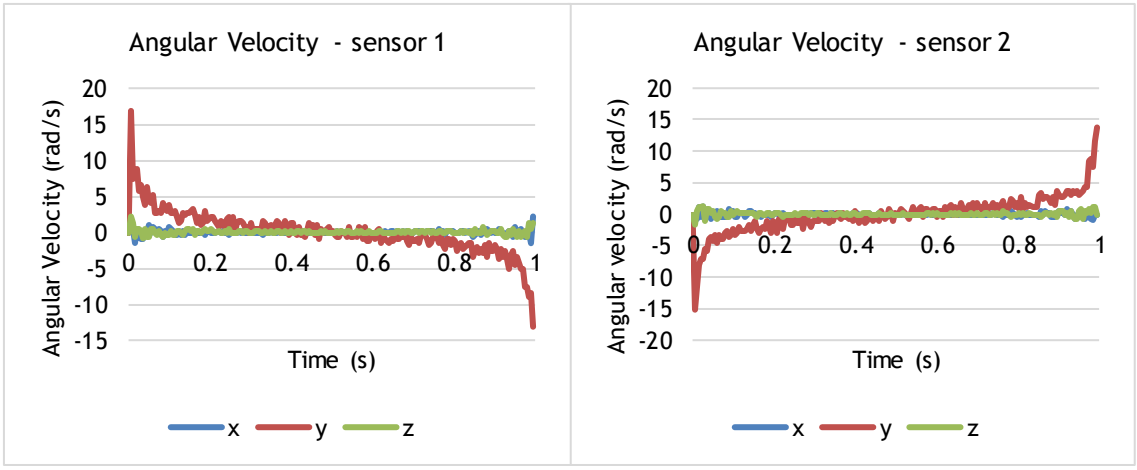

Figure 4.8 - Input signal for sensor1 and sensor2 placed at 0°, with 20% noise.

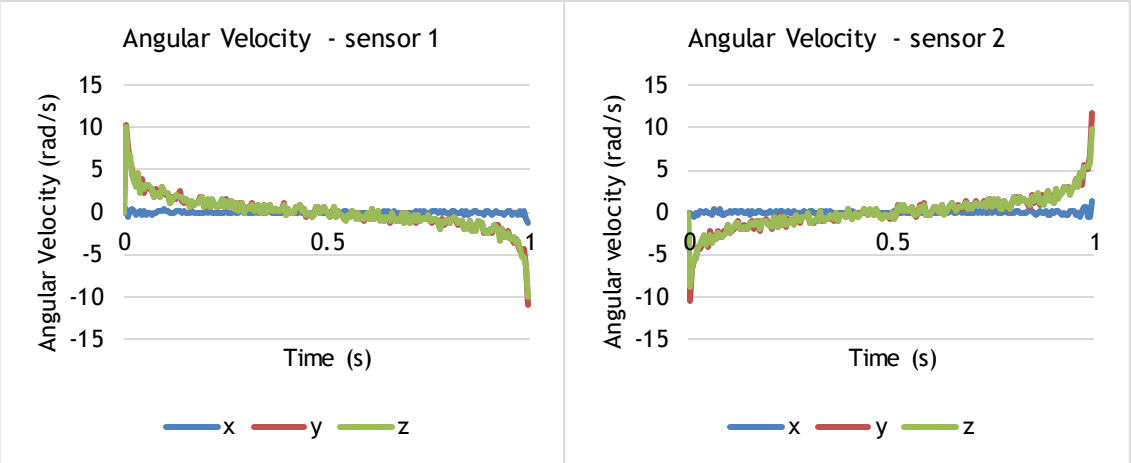

Figure 4.9 - Input signal for sensor1 and sensor2 placed at 45°, with 10% noise.

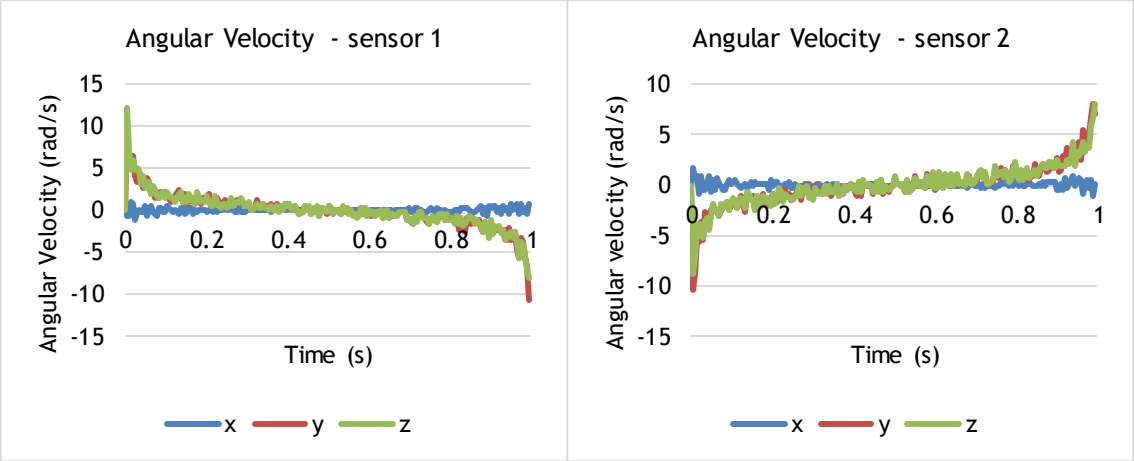

Figure 4.10 - Input signal for sensor1 and sensor2 placed at 45°, with 20% noise.

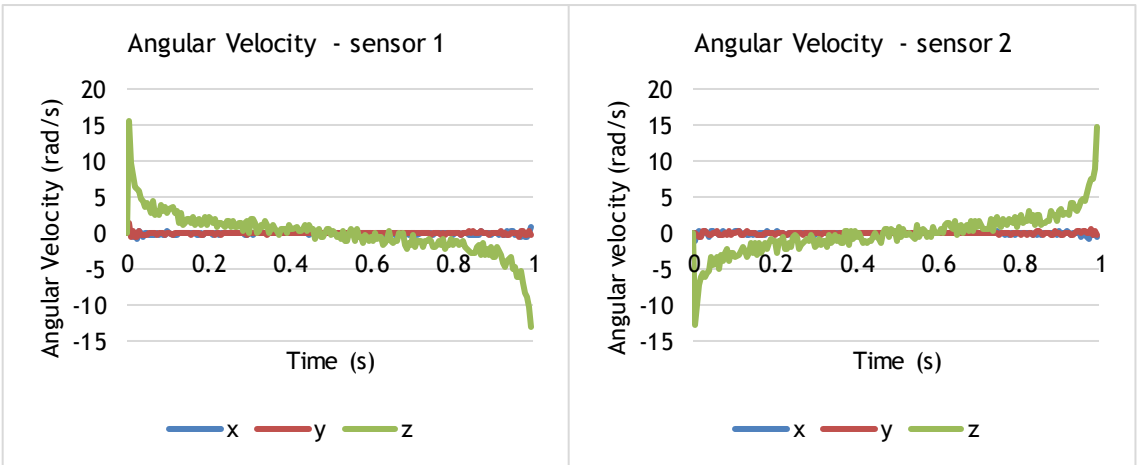

Figure 4.11 - Input signal for sensor1 and sensor2 placed at 90°, with 10% noise.

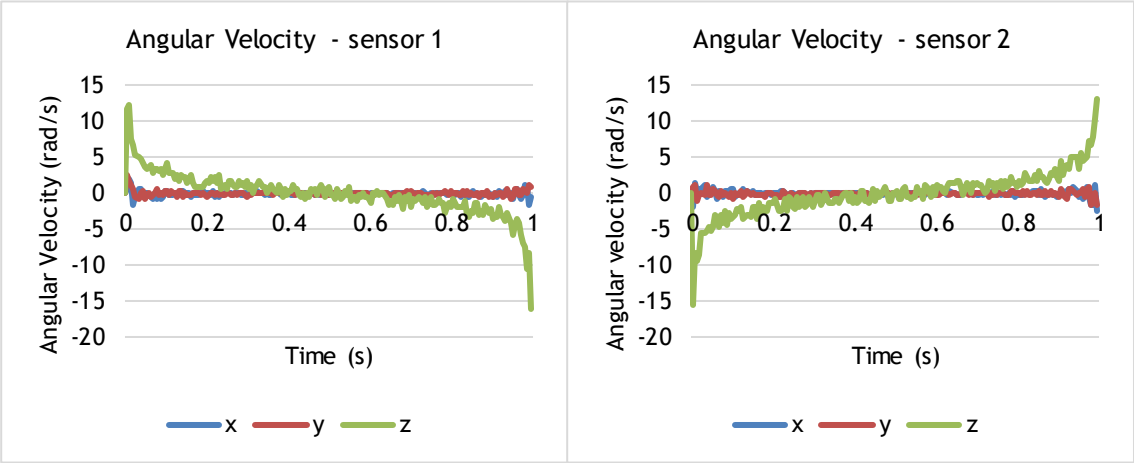

Figure 4.12 - Input signal for sensor1 and sensor2 placed at 90°, with 10% noise.

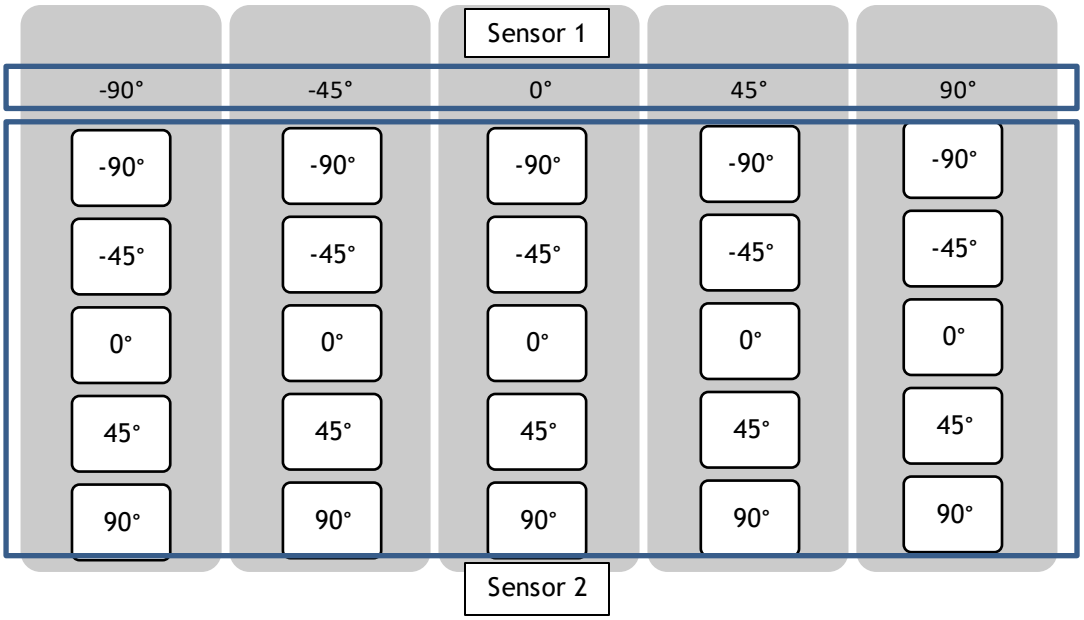

Figure 4.13 - All possible combinations for both sensors.

Notice that here are not presented all of the 25 possible combinations, see Annex A for further consult and in the figure below is demonstrated such combinations.

4.2- Simulation Results

For the random placement of the sensors, 50 runs were performed for each value of tolerance in each noise value. On the other hand, for fixed values, 10 runs were performed for each of the 25 combinations of sensors, also in each noise value. In the end, it was performed 300 runs for random placement and 1500 runs for fixed placement.

Let ${\phi_1}^{true}, {\phi_2}^{true}$ be the true values of ϕ_1 and ϕ_2, respectively, and ${\phi_1}^{sim}, {\phi_2}^{sim}$ be the final values which the algorithm converged to. Then, it will the assessed the difference between ${\phi_i}^{sim}$ and ${\phi_i}^{true}$ for each run, in the end an arithmetic mean is applied.

Concerning the random placement of sensors, the results are exhibited below.

Table 4.1 - Summary of the results obtained for the random placement of sensors.

Noise	10%			20%		
Tolerance	0.05	0.01	0.001	0.05	0.01	0.001
Mean difference ϕ_1 (°)	0.91173	0.86295	0.74518	1.46716	1.58024	1.45685
Std difference ϕ_1	1.06479	0.75288	0.67196	1.35706	1.26367	2.33721
Mean difference ϕ_2 (°)	0.89793	0.88199	0.85751	1.70131	1.64038	1.39668
Std difference ϕ_2	0.79312	0.82909	0.56456	1.45919	1.26027	1.28166
Mean iterations	3.0	4.0	6.0	4.0	5.0	7.0
Std iterations	0.77643	0.84610	2.06712	1.18559	1.96685	2.00940

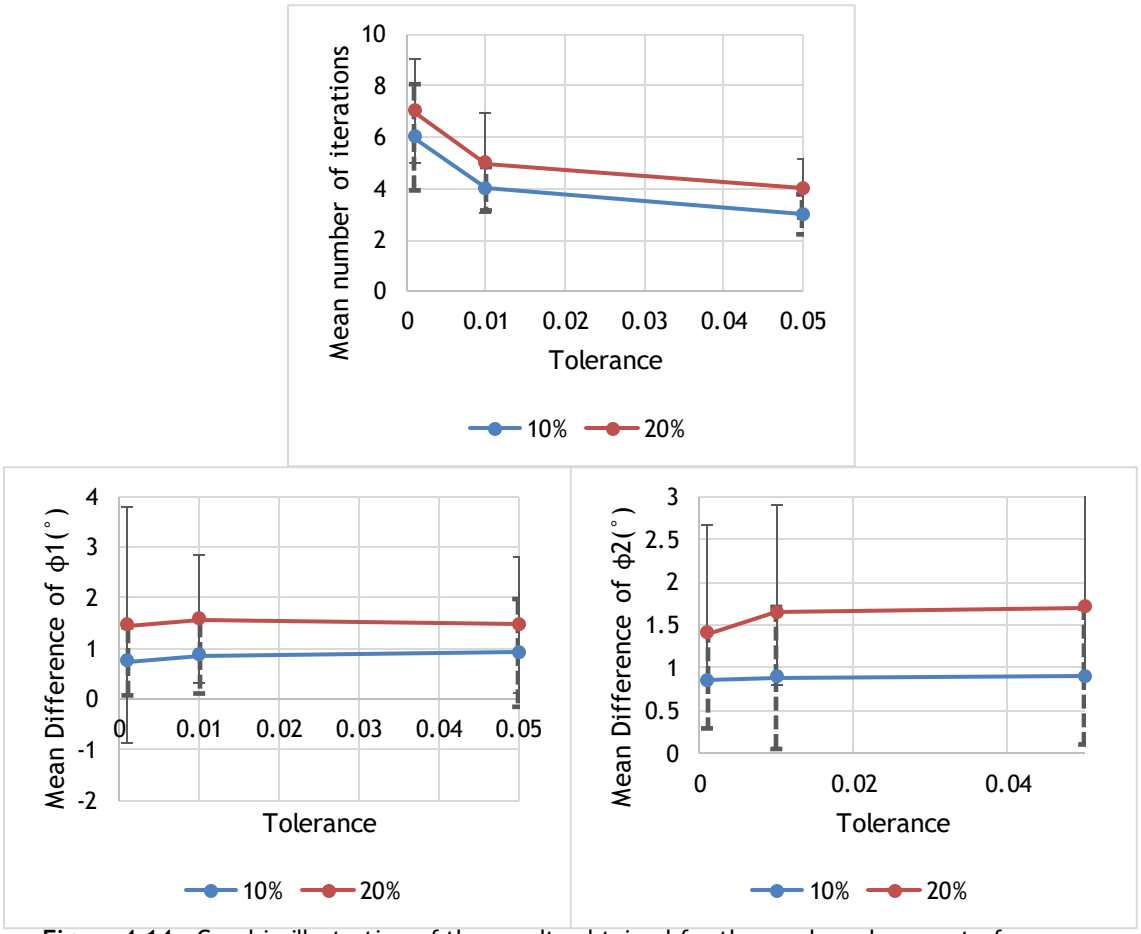

Figure 4.14 - Graphic illustration of the results obtained for the random placement of sensors.

The table and figure above allow to take relevant conclusions. First, as tolerance increases, the number of iterations decreases as expected, since a higher value of tolerance means that the algorithm will stop sooner. This fact is valid for both values of noise, being that for a noise of 20%, the number of iterations is higher because the signal was more contaminated and so the algorithm needed to performed more iterations to converge.

Regarding the difference between ϕ_1^{sim} and ϕ_1^{true}, for 10% noise, the difference was less than 1.0°, and for 20% noise less than 1.6°. For the same reason above mentioned, it was expected that as the noise increases, the difference was higher. Nevertheless, both values are

quite acceptable. Concerning the difference between ϕ_2^{sim} and ϕ_2^{true}, a similar conclusion can be drawn, however, for a 20% noise the difference is less than 1.8° and for a tolerance of 0.05, the mean is slightly higher than for 0.01 which was not expected. Nonetheless, the difference is not significant and it can be explained, for example, by the initial estimations for 0.01 were worse than the ones for 0.05, and so, the algorithm needed a few more iterations to converge. Regardless, both are very satisfactory and prove that the algorithm is capable of a reliable convergence, independently of the amount of noise added to data analyzed in this project.

The following six tables present the results obtained for the fixed placements of sensors.

Table 4.2 – Mean of the obtained results for tolerance of 0.05 and noise 10% and corresponding standard deviation.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	0.70297	0.43605	0.24508	0.35143	0.59202
		Std difference ϕ_1	0.07655	0.38399	0.14814	0.36262	0.09430
		Mean difference ϕ_2 (°)	0.47078	0.55416	1.18276	1.56429	1.28558
		Std difference ϕ_2	0.45185	0.20245	0.83166	0.38431	0.29579
		Mean iterations	3.0	4.0	3.0	3.0	3.0
		Std iterations	0.40000	0.64031	0.60000	0.74833	0.60000
	-45°	Mean difference ϕ_1 (°)	0.57086	0.25423	1.95845	1.00574	0.72201
		Std difference ϕ_1	0.38425	0.21999	0.48933	0.57260	0.86210
		Mean difference ϕ_2 (°)	0.46765	1.27897	0.45543	1.19398	1.32955
		Std difference ϕ_2	0.94390	1.05669	0.55940	0.65878	0.95514
		Mean iterations	4.0	3.0	3.0	3.5	4.0
		Std iterations	0.64031	0.74833	1.01980	0.78102	0.80000
	0°	Mean difference ϕ_1 (°)	0.33789	1.87022	0.38317	0.93493	0.59021
		Std difference ϕ_1	0.42776	0.55137	0.22196	0.60625	0.58762
		Mean difference ϕ_2 (°)	0.53932	1.60851	1.07745	0.95580	0.46659
		Std difference ϕ_2	0.46702	1.10357	0.65995	0.54120	0.64272
		Mean iterations	3.0	3.0	4.0	3.0	3.0
		Std iterations	0.94340	0.45826	0.89443	0.63246	0.94340
	45°	Mean difference ϕ_1 (°)	0.49997	0.51188	0.40608	1.90704	0.81703
		Std difference ϕ_1	0.60832	0.34505	0.69065	0.46996	0.74050
		Mean difference ϕ_2 (°)	1.50019	1.17420	0.12226	0.78353	1.72238
		Std difference ϕ_2	0.43279	0.71927	0.08838	0.59663	0.38285
		Mean iterations	4.0	3.0	3.0	3.0	3.0
		Std iterations	0.74833	0.91652	0.74833	0.70000	0.63246
	90°	Mean difference ϕ_1 (°)	1.11728	1.68935	0.41323	0.20719	0.61222
		Std difference ϕ_1	0.45405	0.47289	0.37642	0.67451	0.39244
		Mean difference ϕ_2 (°)	0.77364	0.74238	1.23685	0.83157	0.55345
		Std difference ϕ_2	0.48109	0.64283	0.82876	0.40747	0.76007
		Mean iterations	3.0	3.0	3.0	3.0	3.5
		Std iterations	0.77460	0.60000	0.77460	0.48990	0.66332

Table 4.3 - Mean of the obtained results for tolerance of 0.05 and noise 20% and corresponding standard deviation.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	1.88865	0.41113	0.77914	1.55443	1.65129
		Std difference ϕ_1	0.49164	0.22124	1.65016	0.60459	0.85624
		Mean difference ϕ_2 (°)	1.14051	3.50534	1.12546	1.36180	0.78398
		Std difference ϕ_2	1.13473	1.69243	0.90441	0.39944	0.40217
		Mean iterations	3.0	4.0	3.0	4.0	3.0
		Std iterations	0.40000	1.18322	1.00499	0.74833	0.92195
	-45°	Mean difference ϕ_1 (°)	2.94131	1.34479	3.77264	0.62544	1.09890
		Std difference ϕ_1	0.55908	1.54807	1.13338	2.90597	1.29081
		Mean difference ϕ_2 (°)	0.60640	2.57153	1.41319	2.82436	1.45361
		Std difference ϕ_2	0.27091	0.66352	0.65794	1.63894	0.99595
		Mean iterations	3.0	3.0	4.0	4.0	4.5
		Std iterations	0.90000	1.18322	1.00000	1.40000	0.78102
	0°	Mean difference ϕ_1 (°)	2.50084	1.36730	1.90251	1.23925	1.61046
		Std difference ϕ_1	1.42174	1.64542	1.21390	0.68021	0.80511
		Mean difference ϕ_2 (°)	2.49804	1.37504	2.29288	2.11012	2.99269
		Std difference ϕ_2	1.93221	1.80747	2.03669	1.48776	1.62991
		Mean iterations	3.5	3.0	4.0	4.0	3.5
		Std iterations	0.97980	1.96214	0.80623	0.89443	0.66332
	45°	Mean difference ϕ_1 (°)	3.91229	0.89486	0.78974	0.28058	2.41381
		Std difference ϕ_1	0.23636	0.50420	0.67311	0.64838	1.96283
		Mean difference ϕ_2 (°)	0.59656	0.94489	2.34979	3.06485	1.58185
		Std difference ϕ_2	0.43631	2.05006	0.42503	0.49825	1.30100
		Mean iterations	4.0	3.0	3.5	4.0	4.0
		Std iterations	1.07703	0.83066	0.66332	1.00000	0.87178
	90°	Mean difference ϕ_1 (°)	0.60289	0.42934	2.17507	1.21679	1.45300
		Std difference ϕ_1	1.04994	0.43557	0.95701	1.42198	1.09319
		Mean difference ϕ_2 (°)	1.50249	0.86052	2.13171	1.95193	1.57088
		Std difference ϕ_2	0.45759	1.36931	1.11525	1.20089	0.49219
		Mean iterations	4.0	3.5	3.0	3.0	3.0
		Std iterations	0.30000	1.07703	0.30000	1.37477	0.45826

Table 4.4 - Mean of the obtained results for tolerance of 0.01 and noise 10% and corresponding standard deviation.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	0.35109	1.08445	0.97436	0.52392	0.66617
		Std difference ϕ_1	0.38582	0.34656	0.53192	0.40291	0.44936
		Mean difference ϕ_2 (°)	0.20598	0.50642	0.44582	0.90422	0.88009
		Std difference ϕ_2	0.41424	0.17095	0.72343	0.26055	0.74983
		Mean iterations	4.0	5.0	4.0	4.0	4.0
		Std iterations	0.70000	1.11803	1.28452	0.60000	0.66332
	-45°	Mean difference ϕ_1 (°)	0.69696	1.42140	0.45609	1.21940	0.24898
		Std difference ϕ_1	0.24778	0.09988	0.41801	0.34811	0.41476
		Mean difference ϕ_2 (°)	0.35530	0.42429	1.48287	0.33292	0.99856
		Std difference ϕ_2	0.31366	0.07011	0.65606	0.44967	0.41925
		Mean iterations	5.0	4.0	4.5	4.5	4.0
		Std iterations	0.60000	0.92195	0.91652	1.20000	0.80000
	0°	Mean difference ϕ_1 (°)	0.38805	0.95422	0.74295	0.74909	0.18045
		Std difference ϕ_1	0.43991	0.67640	0.44381	0.77056	0.56145
		Mean difference ϕ_2 (°)	0.69556	0.90330	1.15343	1.25055	0.73174
		Std difference ϕ_2	0.63781	0.70373	0.56743	0.66272	0.52286
		Mean iterations	4.5	3.5	5.0	4.0	4.0
		Std iterations	1.20000	1.04403	0.92195	1.55242	0.80000
	45°	Mean difference ϕ_1 (°)	0.75694	0.32176	0.74711	0.92789	1.37019
		Std difference ϕ_1	0.53074	0.29429	0.32948	0.38976	0.48776
		Mean difference ϕ_2 (°)	1.06025	0.52628	0.21223	0.96864	0.64814
		Std difference ϕ_2	0.52077	0.18240	0.50751	0.47583	0.41942
		Mean iterations	4.0	4.0	4.5	4.0	4.5
		Std iterations	0.53852	1.00000	1.02470	0.74833	0.87178
	90°	Mean difference ϕ_1 (°)	0.16033	0.48101	0.78465	0.86625	0.38451
		Std difference ϕ_1	0.38032	0.23698	0.57184	0.72917	0.28753
		Mean difference ϕ_2 (°)	0.06866	0.17958	0.46130	0.98738	0.56344
		Std difference ϕ_2	0.73811	0.14968	0.64476	0.69927	0.75477
		Mean iterations	4.0	5.0	4.0	4.0	4.0
		Std iterations	0.94340	0.70000	0.94340	0.45826	0.94340

Table 4.5 - Mean of the obtained results for tolerance of 0.01 and noise 20% and corresponding standard deviation.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	2.27720	2.68221	1.31863	2.59522	1.69349
		Std difference ϕ_1	0.09878	0.72338	0.10991	0.83420	0.04515
		Mean difference ϕ_2 (°)	0.30900	0.98383	1.03044	2.86667	2.23126
		Std difference ϕ_2	0.65201	0.15574	0.02831	0.87152	1.09159
		Mean iterations	5.50	7.00	4.00	5.50	5.00
		Std iterations	0.90000	1.40000	0.66332	1.68819	0.74833
	-45°	Mean difference ϕ_1 (°)	2.62635	0.94877	0.78558	2.55256	1.21675
		Std difference ϕ_1	0.78530	1.54608	0.08355	2.42569	0.02928
		Mean difference ϕ_2 (°)	0.76265	2.51926	0.65241	2.39378	2.67069
		Std difference ϕ_2	1.04924	1.36549	0.03781	0.70700	0.94941
		Mean iterations	5.0	5.0	4.0	5.0	4.5
		Std iterations	0.87178	0.87178	1.28062	0.83066	0.94340
	0°	Mean difference ϕ_1 (°)	2.45694	0.84428	0.72416	4.08383	0.17000
		Std difference ϕ_1	1.57629	0.28401	0.20681	0.20644	0.01063
		Mean difference ϕ_2 (°)	1.56720	3.04753	2.92875	2.20902	0.65657
		Std difference ϕ_2	1.68866	0.53258	1.11361	0.11786	0.22760
		Mean iterations	4.50	5.50	6.00	4.00	5.00
		Std iterations	1.64012	1.41774	1.67631	1.72047	0.90000
	45°	Mean difference ϕ_1 (°)	1.14021	3.04241	1.12828	0.88703	1.07126
		Std difference ϕ_1	1.68110	0.62078	0.04626	0.69436	0.31771
		Mean difference ϕ_2 (°)	0.25125	0.62774	1.82208	0.55036	0.55328
		Std difference ϕ_2	0.41038	0.41329	0.00720	1.80469	0.01049
		Mean iterations	4.5	6.0	5.0	5.0	5.0
		Std iterations	1.30000	0.70000	1.48324	1.16619	1.24900
	90°	Mean difference ϕ_1 (°)	3.83111	0.89492	0.89273	2.40007	0.93231
		Std difference ϕ_1	0.04066	0.08647	0.80251	0.40604	0.69360
		Mean difference ϕ_2 (°)	0.37185	0.50149	0.57008	0.23826	0.97988
		Std difference ϕ_2	0.19307	0.13616	0.58050	1.14409	1.37934
		Mean iterations	5.0	4.5	5.0	5.0	4.0
		Std iterations	0.97980	0.91652	0.97980	0.64031	1.04403

Table 4.6 – Mean of the obtained results for tolerance of 0.001 and noise 10% and corresponding standard deviation.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	1.41491	0.69313	0.22224	0.56785	0.74301
		Std difference ϕ_1	0.44097	0.94006	0.40515	0.43784	0.30500
		Mean difference ϕ_2 (°)	1.52925	0.78872	0.79570	1.05580	0.64616
		Std difference ϕ_2	0.43187	0.93125	0.26487	0.32300	0.47731
		Mean iterations	6.5	6.0	5.0	6.0	5.0
		Std iterations	1.50000	2.15639	1.28452	2.10713	1.55242
	-45°	Mean difference ϕ_1 (°)	0.79624	0.98427	0.34377	0.93550	0.45308
		Std difference ϕ_1	0.60258	0.72783	0.24636	0.33479	0.07982
		Mean difference ϕ_2 (°)	0.54032	1.20600	1.13490	1.68465	0.66324
		Std difference ϕ_2	0.32042	0.72759	0.39532	0.42716	0.38431
		Mean iterations	6.0	7.0	6.0	6.5	6.0
		Std iterations	2.10713	2.37487	2.30000	2.14709	1.13578
	0°	Mean difference ϕ_1 (°)	0.65953	0.92233	0.70974	0.90300	0.05297
		Std difference ϕ_1	0.33591	0.19884	0.35109	0.19536	0.01078
		Mean difference ϕ_2 (°)	1.19061	0.73377	0.40639	0.21559	0.06550
		Std difference ϕ_2	0.40895	0.65035	0.17633	0.47131	0.02430
		Mean iterations	6.5	8.0	8.0	7.5	6.0
		Std iterations	2.11896	1.40000	1.81384	2.82843	1.28062
	45°	Mean difference ϕ_1 (°)	1.28145	0.58295	0.84620	0.38007	0.50593
		Std difference ϕ_1	0.74334	0.84432	0.52743	0.22251	0.09681
		Mean difference ϕ_2 (°)	0.13743	0.90962	1.28534	0.87270	0.30046
		Std difference ϕ_2	0.28738	0.23931	0.82916	0.32449	0.04430
		Mean iterations	6.0	6.0	7.0	6.5	7.0
		Std iterations	0.45826	1.44568	1.30000	2.50799	1.95959
	90°	Mean difference ϕ_1 (°)	0.57077	0.43807	0.56909	0.38109	0.80413
		Std difference ϕ_1	1.11031	0.43900	0.06775	0.71407	0.28818
		Mean difference ϕ_2 (°)	0.87084	1.07142	0.09177	0.56099	0.93443
		Std difference ϕ_2	0.61836	0.61504	0.95304	0.02417	0.20618
		Mean iterations	7.0	6.0	6.0	6.0	5.0
		Std iterations	1.42829	2.07123	1.42829	0.83066	0.97980

Table 4.7 - Mean of the obtained results for tolerance of 0.001 and noise 20% and corresponding standard deviation.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	0.76407	1.92040	1.92136	1.96659	1.06854
		Std difference ϕ_1	1.17928	1.40852	0.69724	0.58595	0.38014
		Mean difference ϕ_2 (°)	0.37125	1.54709	0.82594	2.64599	0.51332
		Std difference ϕ_2	0.25144	1.20025	0.86890	1.28278	0.28346
		Mean iterations	6.0	7.0	6.5	6.5	6.0
		Std iterations	4.06940	2.13542	1.16619	1.16619	1.00000
	-45°	Mean difference ϕ_1 (°)	1.86008	1.81108	2.65092	0.67651	2.32524
		Std difference ϕ_1	0.83308	1.80178	0.60832	0.77818	0.94671
		Mean difference ϕ_2 (°)	1.37756	1.75043	1.86217	0.57948	0.61364
		Std difference ϕ_2	0.75421	1.51211	0.74075	0.48977	2.13511
		Mean iterations	9.0	7.0	7.0	7.0	7.5
		Std iterations	2.60960	2.34307	2.08806	1.75784	2.64764
	0°	Mean difference ϕ_1 (°)	0.34867	1.61462	0.36656	1.45707	0.80166
		Std difference ϕ_1	0.54067	0.46656	1.01137	1.66723	1.63335
		Mean difference ϕ_2 (°)	0.94192	0.96011	0.31882	0.68285	0.91466
		Std difference ϕ_2	0.36739	0.95462	1.14183	1.72517	0.24840
		Mean iterations	6.5	6.5	9.0	7.5	7.0
		Std iterations	0.66332	2.62488	1.72047	2.41039	4.72123
	45°	Mean difference ϕ_1 (°)	1.53858	1.77272	0.08953	0.98264	2.05052
		Std difference ϕ_1	2.01859	0.78034	1.56550	0.35092	1.51445
		Mean difference ϕ_2 (°)	0.77130	0.88273	0.99012	1.17358	1.45141
		Std difference ϕ_2	0.69499	0.56455	0.56001	1.27904	1.22453
		Mean iterations	7.0	9.5	7.5	6.0	7.5
		Std iterations	2.37487	5.04083	1.02470	0.92195	1.62788
	90°	Mean difference ϕ_1 (°)	0.33626	0.17431	1.47653	2.84905	0.95679
		Std difference ϕ_1	0.82932	0.79718	1.05780	1.37363	0.37614
		Mean difference ϕ_2 (°)	1.40386	0.96144	1.75144	1.56886	0.20888
		Std difference ϕ_2	1.42486	0.08065	0.58600	1.62542	0.32919
		Mean iterations	7.0	6.5	6.0	7.0	7.0
		Std iterations	1.16619	1.35647	1.16619	2.06155	0.70000

First, just focusing on results for 10% noise, increasing the tolerance, decreases the number of iterations, similar to the previous results. Moreover, with increasing tolerance, the difference between $\phi_i{}^{sim}$ and $\phi_i{}^{true}$, $i = 1,2$, also increases, which was expected since a higher tolerance means that the algorithm will converge for higher values of difference. It is noteworthy that the larger difference for ϕ_1 is 1.95845°, 1.42140° and 1.41491° for 0.05, 0.01 and 0.001, respectively. On the other hand, for ϕ_2 the higher difference is 1.72238°, 1.48287° and 1.68465° for 0.05, 0.01 and 0.001, in that order.

Regarding the results for 20% noise, once can come to similar conclusions concerning the decreasing number of iterations and increasing values of difference for both ϕ_1 and ϕ_2, with increasing tolerance. Comparing the results for both values of noise, its increment implies a higher number of iterations for each value of tolerance and higher values of difference for both ϕ_1 and ϕ_2. As already mentioned, this was expected because for 20% of noise, the signal is more contaminated and so it takes more iterations to converge and ϕ_1^{sim} and ϕ_2^{sim} are further from their true values.

Summarizing, regardless the placement of the sensor, whether it is randomly placed or in fixed orientations, the algorithm can converge to very satisfactory values of ϕ_1 and ϕ_2 with less than 10 iterations, with simulation data.

Chapter 5

Experimental Validation

As a final step of algorithm assessment, an experimental validation was performed. Wireless inertial sensors, provided by SWORD Health (figure 5.1), are attached to the right thigh and shank using elastic straps. Four different exercises were analyzed: gait, squat, sitting knee flexion and sitting knee flexion with hip flexion (see figure 5.2 through 5.4). These exercises were chosen, in order to reproduce real cases of patients with different mobility of the lower limbs and to proof that the algorithm converges independently of the prescribed exercise.

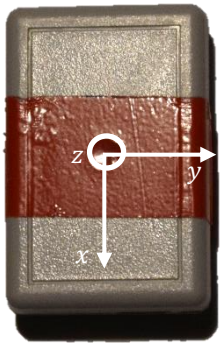

Figure 5.1- Sensor provided by SWORD Health.

Figure 5.2 - Sitting knee flexion with hip flexion exercise.

Figure 5.3 – Squat exercise.

Figure 5.4 – Sitting knee flexion exercise.

Similar to simulated data validation, five positions for each sensor were considered: -90°, -45°, 0°, 45° and 90°, producing in the end 25 combinations and for each combination, and 10 runs were performed, yielding 250 runs for each exercise (figure 5.5 through 9). The true orientation of the local sensor frame are roughly determined using manual measurements: two circular wires, perimeter equal to each member, were built. In each one, with the aid of a protractor, the five positions were marked, and the positions 0° for both sensors was aligned with the knee (see figure 5.10). Let ${\phi_1}^{true}, {\phi_2}^{true}$ be the true values of ϕ_1 and ϕ_2, respectively, and ${\phi_1}^{exp}, {\phi_2}^{exp}$ be the final values to which the algorithm converges to. Then, it will the assessed the difference between ${\phi_i}^{exp}$ and ${\phi_i}^{true}$ for each run, in the end an arithmetic mean is applied.

All of the trials were performed by the same individual and the tolerance applied was 0.01 for all exercises.

Figure 5.5 - Sensor placement: $\phi_1 = -90°$ and $\phi_2 = -90°, -45°, 0°, 45°$ and $90°$ (right to left).

Figure 5.6 - Sensor placement: $\phi_1 = -45°$ and $\phi_2 = -90°, -45°, 0°, 45°$ and $90°$ (right to left).

Figure 5.7 - Sensor placement: $\phi_1 = 0°$ and $\phi_2 = -90°, -45°, 0°, 45°$ and $90°$ (right to left).

Figure 5.8 - Sensor placement: $\phi_1 = 45°$ and $\phi_2 = -90°, -45°, 0°, 45°$ and $90°$ (right to left).

Figure 5.9 - Sensor placement: $\phi_1 = 90°$ and $\phi_2 = -90°, -45°, 0°, 45°$ and $90°$ (right to left).

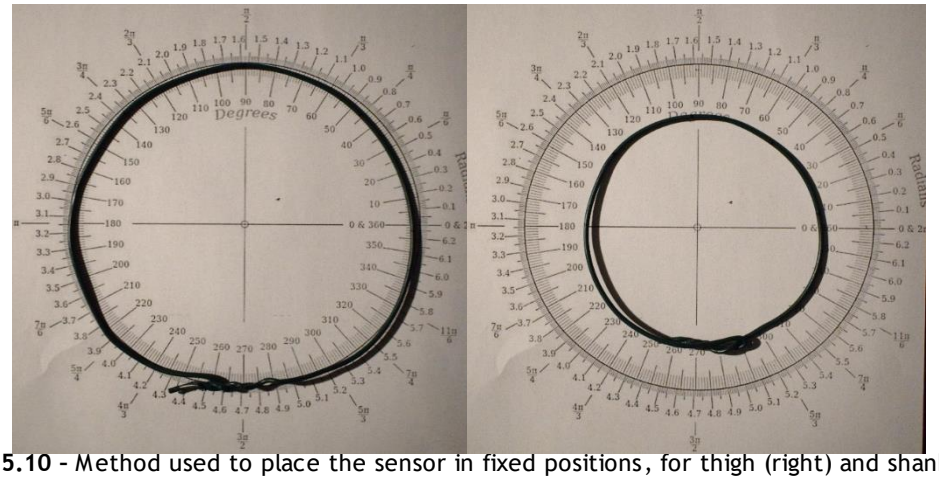

Figure 5.10 - Method used to place the sensor in fixed positions, for thigh (right) and shank (left).

5.1- Experimental Results

The following four tables exhibit the results obtained for the four different exercises performed.

Table 5.1 – Mean Results obtained for gait.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	6.16569	6.06058	3.70908	9.30295	5.95940
		Std difference ϕ_1	4.67283	6.05288	2.74804	8.03744	3.27336
		Mean difference ϕ_2 (°)	10.79066	7.95375	9.11622	6.64533	13.73589
		Std difference ϕ_2	6.33422	4.21472	3.92824	7.03387	3.62221
		Mean iterations	5.5	5.0	8.5	4.5	5.5
		Std iterations	1.00499	3.82230	2.99333	1.70000	1.80278
	-45°	Mean difference ϕ_1 (°)	7.83136	6.58090	2.55293	16.14445	10.89599
		Std difference ϕ_1	9.27505	4.38601	4.81504	8.07604	5.47310
		Mean difference ϕ_2 (°)	17.95196	15.99401	17.25602	17.76944	10.89599
		Std difference ϕ_2	2.25055	4.97366	4.56861	2.14580	5.47310
		Mean iterations	5.0	5.5	6.0	4.5	5.0
		Std iterations	0.70000	4.16653	1.77764	1.20000	2.10000
	0°	Mean difference ϕ_1 (°)	11.62555	3.23654	5.97828	10.78824	5.63522
		Std difference ϕ_1	5.08945	4.62417	4.47389	5.97258	4.13423
		Mean difference ϕ_2 (°)	13.66826	8.55514	8.46943	6.14044	8.00034
		Std difference ϕ_2	4.24464	5.27896	4.46503	5.67096	7.43498
		Mean iterations	5.0	5.5	7.0	5.0	5.0
		Std iterations	1.51327	2.40832	3.80132	1.43178	1.28062
	45°	Mean difference ϕ_1 (°)	4.86398	6.63870	5.62814	7.32671	5.32402
		Std difference ϕ_1	8.78925	4.68410	3.09983	7.23685	2.03903
		Mean difference ϕ_2 (°)	14.02631	13.69110	12.23662	2.48324	10.86846
		Std difference ϕ_2	4.14848	2.68884	2.93250	7.96482	5.21284
		Mean iterations	6.0	4.0	8.0	5.0	4.5
		Std iterations	2.08806	3.25730	3.98121	2.36854	1.98997
	90°	Mean difference ϕ_1 (°)	9.36322	6.98764	8.79794	5.89077	8.92996
		Std difference ϕ_1	9.31346	5.36820	7.70716	7.27236	5.15905
		Mean difference ϕ_2 (°)	12.31198	11.85872	14.70445	9.74145	3.93813
		Std difference ϕ_2	5.29945	2.91273	3.99397	4.93039	4.96851
		Mean iterations	5.0	4.5	4.0	5.0	5.5
		Std iterations	4.07431	2.19317	3.00167	1.84662	1.74642

Table 5.2 – Mean results obtained for sitting hip flexion with knee flexion.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	8.38324	10.44092	6.18288	9.98703	8.69625
		Std difference ϕ_1	4.12013	4.39262	3.42841	4.46005	4.18479
		Mean difference ϕ_2 (°)	11.79090	2.77592	7.97632	6.39412	2.05543
		Std difference ϕ_2	7.09446	4.72511	5.50429	1.75946	6.55688
		Mean iterations	5.5	5.0	4.5	4.0	4.0
		Std iterations	2.77308	2.02237	1.48324	1.56205	1.09545
	-45°	Mean difference ϕ_1 (°)	0.69437	6.73541	7.08633	1.65537	4.84456
		Std difference ϕ_1	1.66135	2.85807	4.20860	1.27650	3.88971
		Mean difference ϕ_2 (°)	9.42488	3.42142	2.90537	4.31645	4.84456
		Std difference ϕ_2	3.27664	3.88785	3.46540	2.13544	3.88971
		Mean iterations	4.0	3.5	5.5	4.5	4.5
		Std iterations	0.92195	2.58650	2.08806	0.91652	1.43178
	0°	Mean difference ϕ_1 (°)	5.31096	8.46125	7.08196	7.32132	14.28125
		Std difference ϕ_1	6.55130	5.14395	4.95861	5.51449	4.82682
		Mean difference ϕ_2 (°)	13.77842	14.12213	9.98318	11.58833	6.79843
		Std difference ϕ_2	3.71426	4.75505	6.45279	6.70249	8.39489
		Mean iterations	4.5	6.5	5.0	5.0	4.0
		Std iterations	1.93907	2.56905	1.61555	3.07409	2.85657
	45°	Mean difference ϕ_1 (°)	4.64517	9.51959	9.10177	5.55792	9.25601
		Std difference ϕ_1	4.38351	3.06735	3.32475	4.26681	3.72975
		Mean difference ϕ_2 (°)	10.57865	8.75709	2.87092	4.02123	7.70777
		Std difference ϕ_2	3.74938	2.77325	1.55683	2.80889	2.45435
		Mean iterations	6.0	6.5	4.0	5.0	5.0
		Std iterations	4.43734	2.22711	3.38378	1.54919	2.30000
	90°	Mean difference ϕ_1 (°)	9.83841	7.08674	12.68303	1.47351	4.65061
		Std difference ϕ_1	5.60425	4.53903	4.50548	5.56333	7.42683
		Mean difference ϕ_2 (°)	10.68933	8.57865	14.58718	3.69626	4.58508
		Std difference ϕ_2	3.29663	3.55470	6.56177	2.40894	8.51027
		Mean iterations	6.0	6.0	3.5	6.0	7.0
		Std iterations	2.14709	2.13542	2.31517	2.05183	2.60960

Table 5.3 – Mean results obtained for squat exercise.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	1.95745	5.83705	4.84338	3.22343	1.98584
		Std difference ϕ_1	3.55990	2.88241	3.42777	5.18630	2.62796
		Mean difference ϕ_2 (°)	56.41426	31.94214	54.01912	33.18733	40.90003
		Std difference ϕ_2	19.62979	19.94060	23.92985	13.92367	11.09975
		Mean iterations	9.5	8.0	14.5	6.0	9.5
		Std iterations	87.86899	57.72876	75.05198	77.85506	57.01973
	-45°	Mean difference ϕ_1 (°)	6.68097	9.70133	8.42181	4.86821	9.45701
		Std difference ϕ_1	4.53515	1.84851	3.95204	3.45688	4.25315
		Mean difference ϕ_2 (°)	67.04785	24.33137	39.86177	37.80232	9.45701
		Std difference ϕ_2	30.36165	32.59309	20.52791	12.59382	4.25315
		Mean iterations	10.0	6.5	5.5	11.0	8.5
		Std iterations	5.26878	3.84187	1.10000	57.07364	76.85337
	0°	Mean difference ϕ_1 (°)	4.79337	5.41542	7.28117	0.88034	4.75547
		Std difference ϕ_1	2.68933	2.10007	6.04304	3.65186	2.26218
		Mean difference ϕ_2 (°)	47.34474	44.16369	38.76352	37.91226	18.27480
		Std difference ϕ_2	30.32630	19.23812	13.50166	14.00222	22.43078
		Mean iterations	48.5	9.5	5.0	9.0	8.5
		Std iterations	90.05337	75.28373	1.28062	76.52189	50.11746
	45°	Mean difference ϕ_1 (°)	7.55144	7.71769	9.00244	7.40066	3.81630
		Std difference ϕ_1	2.69868	3.25548	11.31671	2.65399	2.09741
		Mean difference ϕ_2 (°)	54.16290	34.11635	40.11031	37.27955	38.90441
		Std difference ϕ_2	16.85694	26.17256	23.39936	7.30645	16.04115
		Mean iterations	8.0	6.5	12.0	7.5	4.0
		Std iterations	1.41774	2.70000	90.06294	57.47521	1.55242
	90°	Mean difference ϕ_1 (°)	3.35846	1.60621	13.50639	3.90306	6.27919
		Std difference ϕ_1	1.88946	3.06496	6.96862	2.45414	3.60070
		Mean difference ϕ_2 (°)	30.18564	33.23157	46.97396	35.71157	52.55203
		Std difference ϕ_2	18.13555	16.27617	13.43274	6.29632	19.65417
		Mean iterations	6.0	5.5	6.0	5.0	5.5
		Std iterations	2.00998	1.73494	2.63818	1.95959	2.05913

Table 5.4 - Mean results obtained for sitting knee flexion.

			ϕ_2				
			-90°	-45°	0°	45°	90°
ϕ_1	-90°	Mean difference ϕ_1 (°)	64.63712	71.95399	61.67620	65.44963	64.49053
		Std difference ϕ_1	20.75168	19.68838	25.95599	14.37611	23.61421
		Mean difference ϕ_2 (°)	5.41402	2.25078	1.87604	1.55148	5.17709
		Std difference ϕ_2	2.59700	1.07413	1.04243	1.21812	1.10170
		Mean iterations	11.0	13.0	13.0	200.0	200.0
		Std iterations	87.16473	93.77932	75.45303	86.23943	88.16241
	-45°	Mean difference ϕ_1 (°)	22.23790	28.19276	40.17578	25.88095	36.48371
		Std difference ϕ_1	24.39160	24.12437	19.89755	11.83217	22.61197
		Mean difference ϕ_2 (°)	5.57278	5.01032	4.51748	2.14293	5.99182
		Std difference ϕ_2	3.70970	1.76435	1.90909	1.23813	2.83790
		Mean iterations	16.0	14.5	7.0	11.0	124.5
		Std iterations	93.06283	75.13162	76.29443	86.09669	92.63201
	0°	Mean difference ϕ_1 (°)	16.07005	27.10381	25.66656	25.88861	24.81329
		Std difference ϕ_1	17.28476	23.48812	14.69294	18.59049	25.21637
		Mean difference ϕ_2 (°)	5.10223	6.12580	5.34754	1.07304	11.93294
		Std difference ϕ_2	3.09614	3.09206	2.33550	0.87751	4.15674
		Mean iterations	5.5	7.5	8.0	4.5	5.0
		Std iterations	2.10000	57.61389	76.67177	58.64034	58.14465
	45°	Mean difference ϕ_1 (°)	41.20323	24.18660	19.70350	40.48047	23.69114
		Std difference ϕ_1	9.47505	7.86676	6.17134	15.29094	8.28160
		Mean difference ϕ_2 (°)	5.14782	2.20597	5.94599	0.92570	4.50052
		Std difference ϕ_2	1.77470	3.34143	2.66074	2.24094	4.40109
		Mean iterations	6.0	5.5	4.0	5.0	7.0
		Std iterations	3.84708	0.94340	2.45153	0.40000	58.05170
	90°	Mean difference ϕ_1 (°)	70.17170	73.92445	67.76123	67.16629	55.06015
		Std difference ϕ_1	24.85173	19.38396	23.02276	33.29202	25.50127
		Mean difference ϕ_2 (°)	6.55654	5.81089	9.07704	8.01476	6.59298
		Std difference ϕ_2	1.41621	2.40280	2.22787	1.69255	2.85838
		Mean iterations	7.0	5.0	6.5	9.0	5.0
		Std iterations	4.10000	2.00000	58.29717	57.69133	14.71734

First, for the walking exercise, the higher difference for ϕ_1 and ϕ_2 found was 16.1445°, for a placement of ϕ_1 in -45° and ϕ_2 in 45° and 17.95196° for ϕ_1 in -45° and ϕ_2 in -90°, respectively. On the other hand, the smallest difference found was 2.55293° for ϕ_1 in -45° and ϕ_2 in 0° and 2.48324° for ϕ_1 and ϕ_1 in 45°. Besides, the algorithm converged with less than 9 iterations.

Regarding the hip flexion with knee flexion exercise, the larger difference for ϕ_1 was 14.28125° for ϕ_1 in 0° and ϕ_2 in 90° and for ϕ_2 was 14.58718° for a 90° and 0° combination of ϕ_1 and ϕ_2. While the smallest difference of ϕ_1 was 0.69437° for ϕ_1 in -45° and ϕ_2 in -90°, and for ϕ_2 was 2.05543° for a combination of -90° and 90°, ϕ_1 and ϕ_2. In this exercise, the algorithm converges with less than 7 iterations.

As it can be perceived, for both exercises, the algorithm achieves very good results in a few iterations. Apparently, there are no relation between the sensors placement and the smallest difference, which can lead to the assumption that the placement is not relevant to improve the performance of the algorithm.

Now focusing, on the squat exercise and sitting knee flexion. For the first, the major difference for ϕ_1 and ϕ_1 was 13.50639° for a combination of 90° and 0°, for ϕ_1 and ϕ_2, and for ϕ_2 was 67.04785° for a combination of -45° and 90°, in the same order. Whereas, the smallest difference was 0.88034° for ϕ_1 in 0° and ϕ_2 in 45° regarding ϕ_1, and for ϕ_2 the smallest difference was 9.945701° for a combination of -45° and 90°. For the latter, the higher difference for ϕ_1 and ϕ_2 found was 73.92445° in 90° and -45°, and for ϕ_2 was 11.93294° for ϕ_1 in 0° and ϕ_2 in 90°. Finally, the smallest difference found for ϕ_1 was 16.07005° for ϕ_1 in 0° and ϕ_2 in -90°, and for ϕ_2 was 0.92570 for ϕ_1 in 45° and ϕ_2 in 45°. Once more, it does not seem to exist a relation between the placement of the sensors and the smallest difference. However, in both exercises, the ϕ values are always very different from the true values (the cases where this happens are highlighted as dark grey in the tables), ϕ_2 in the squat exercise and ϕ_1 in the sitting knee flexion. In order to clarify what is occurring in these cases, the magnitude of angular velocities of each body segment were analyzed. Thus, five cases in which this happened were selected, for each exercise, and let τ be the ratio between the magnitude of the angular velocity of the shank and thigh:

$$\tau = \frac{||angular\ velocity\ shank||_2}{||angular\ velocity\ thigh||_2} \tag{5.1}$$

For a squat, τ is 0.24 and for sitting knee flexion is 6.29 (table 5.5).

Table 5.5 - Results for τ obtained in both exercises.

	τ_{Squat}	$\tau_{Sitting\ knee\ flexion}$
Trial 1	0.26473	7.21067
Trial 2	0.25215	5.09854
Trial 3	0.24090	6.51717
Trial 4	0.27509	5.09906
Trial 5	0.20860	7.37476
Mean	0.24829	6.26004

For the second exercise this is very perceptible since in this exercise the subject is sitting and only moving the shank back and forward. This leads to a very small angular velocity of the thigh and as this is the quantity that is introduced into the algorithm, if it is not rich enough the algorithm will not converge. Regarding the squat, the sensor placed in the shank does not converge correctly because the perfect squat is not a natural human movement as the one simulated in the previous chapter. For example, when squatting, the hip is not perfectly aligned with the ankle: on one hand, the angular velocity of the thigh will be mostly around the yy axes, in sensor coordinates, while the angular velocity of the shank will not have a preferred axis, leading to a poor convergence.

Therefore, in order to predict such phenomena and increasing the algorithm robustness and accuracy, an extra condition was added to the algorithm: if $\tau < 0.25$, the j of the knee is the j of the thigh, if $\tau > 6$, the j of the knee is the j of the shank, otherwise both j can be considered.

Moreover, highlighted as light grey in tables 5.3 and 5.4, are marked the standard deviation of the number of iterations higher than 50, in squat and sitting knee flexion. As above mentioned, in some cases the convergence is weak for only one sensor, and in those cases the algorithm enters an infinite loop since the angular velocity of such body segment was not rich enough to converge properly. To overcome this situation, an additional condition was implemented in the algorithm: the algorithm will stop if $\|\vec{V}_{diff}\|$ is smaller than 0.01 or if the number of iterations is greater than 200, to prevent the algorithm entering an infinite loop. A specific case where this occurred for a squat is illustrated in figure 5.1. Observing such figure, the algorithm entered an infinite loop since it was undecided between two pairs of ϕ_1 and ϕ_2. For ϕ_1, the two values are very similar and both are good convergence, whereas for ϕ_2 both are very different from each other and both are poor convergences, because it is known for a fact that a movement with angular velocity around the yy axis, ϕ_1 and ϕ_2 must be close to 0, according to equation 3.4 and 3.5.

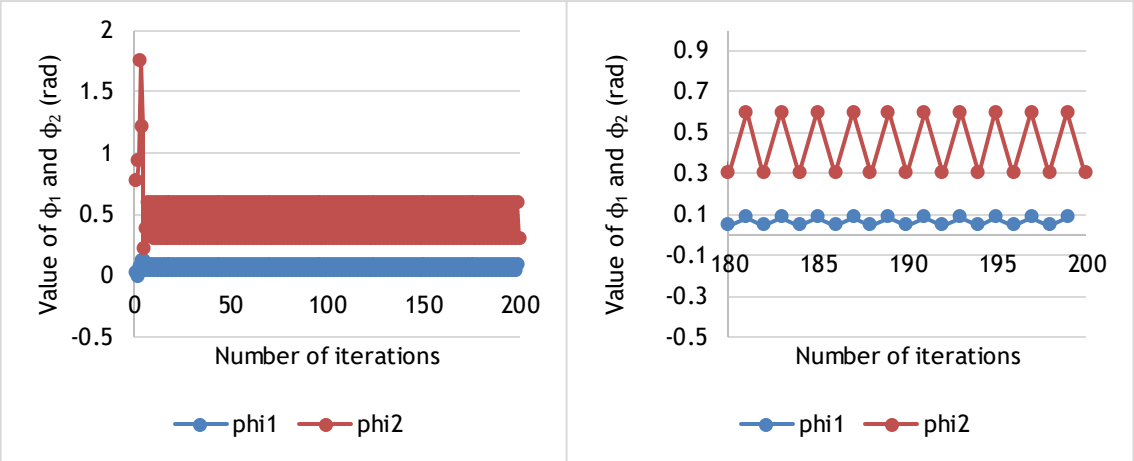

Figure 5.11 – Results obtained in an infinite loop case. Right: all values for the 200 possible iterations; left: zoom of right.

Thus, with the above adjustments, it was observed that the algorithm is now capable to converge with very satisfactory results to the true values of $\boldsymbol{\phi_1}$ and $\boldsymbol{\phi_2}$, leading to a correct $\boldsymbol{j}$ of the knee joint.

It is necessary to reinforce the idea that without the introduced modifications (the two conditions above mentioned) the data from the human movement have to be rich enough in order to achieve a successful convergence, which is not always possible in individuals with physical limitations. With the included modification, the algorithm becomes more robust and allows to obtain always a very accurate estimation of the knee joint axis.

Chapter 6

Joint Position Estimation

Joint orientation and position are two quantities that need to be address when it comes to IMU-based human motion analysis. As already mentioned, one major problem is that the local coordinates axes of the IMUs are not aligned with the body segment axes. Both quantities can be measured manually, but in a three-dimensional space, it can lead to low accuracy results [99]. Besides the need to know joint axis, some joint angle algorithms require additional information of the joint position in local sensor coordinates [91][99]. Previous methods rely on the accuracy of the joint orientation to improve the position of the joint if the kinematic constrains are exploited, and vice versa [91][93].

In this chapter, it is presented a method to improve the joint position estimation based on the joint orientation, obtained by the previous algorithm. It must be mentioned that this part of the thesis goes beyond the proposed work by SWORD Health. Although this is a real problem and adds an extra feature to the previous developed algorithm, it will also improve the company's software for visualization of the avatar and it can also be the first step for a future work. The algorithm was also implemented in Python, using PyCharm. As a final step, it was added a graphic visualization of results in real-time using VPython tool.

7.1- Algorithm Implementation

After the orientation of the joint axis is obtained, $\boldsymbol{j}_i, i = 1, 2$, in the local coordinates, one can find the joint position of the knee. First, let $\vec{\boldsymbol{o}}_i$ be the vector that determines the positions of the knee in coordinates of sensor 1 (thigh) and sensor 2 (shank), respectively (figure 6.1). If one assume that the sensors are placed in the middle of the body segment, then, the position of the joint, $\vec{\boldsymbol{o}}_i, i = 1, 2$, is positioned halfway (figure 6.2 and 6.3), locally in the coordinates of sensor 1 and 2, L_1 and L_1, respectively.

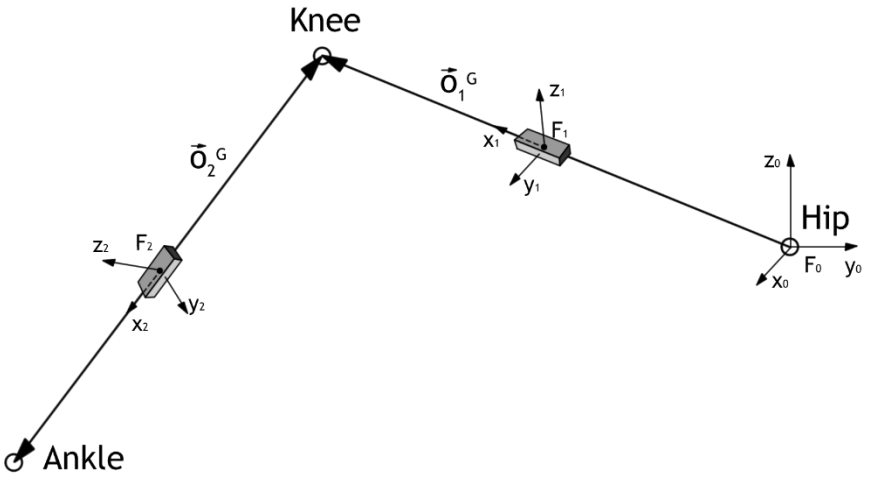

Figure 6.1 - Representation of the leg. In each body segment is placed a sensor with its own coordinated system, $\boldsymbol{F_i, i = 1,2}$.

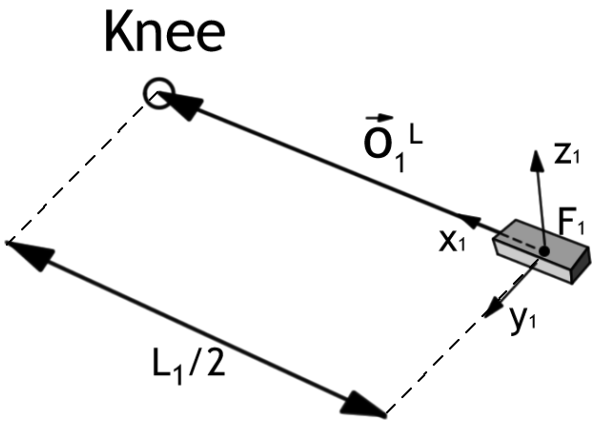

Figure 6.2 - Representation of half the thigh.

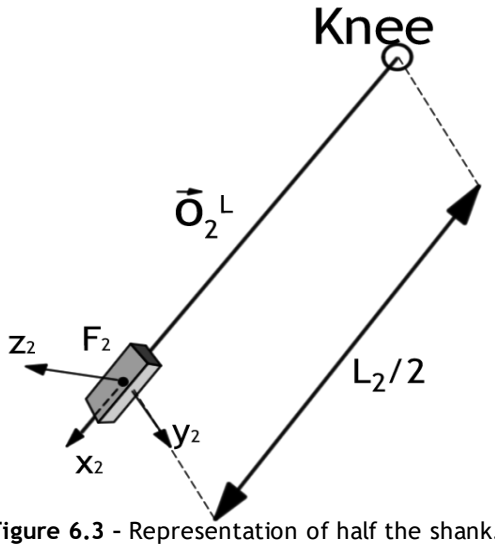

Figure 6.3 - Representation of half the shank.

Thereby, the positions can be defined as:

$$\vec{\boldsymbol{o}}_1^L = \begin{bmatrix} \frac{L_1}{2} \\ 0 \\ 0 \end{bmatrix}, \tag{6.1}$$

$$\vec{\boldsymbol{o}}_2^L = \begin{bmatrix} -\frac{L_2}{2} \\ 0 \\ 0 \end{bmatrix}. \tag{6.2}$$

Since each sensor has its own coordinates, it is not possible to perform any operations with both (local) positions. This is only possible if they are both expressed in a global frame. Therefore, the matrix described in 2.12, will be applied to transform $\{\boldsymbol{o}_i, \boldsymbol{j}_i, \}, i = 1,2$, in global coordinates:

$$\vec{\boldsymbol{o}}_i^G = {}_G^L[\boldsymbol{R}]\vec{\boldsymbol{o}}_i^L, i = 1,2, \tag{6.3}$$

$$\vec{\boldsymbol{j}}_i^G = {}_G^L[\boldsymbol{R}]\vec{\boldsymbol{j}}_i^L, i = 1,2, \tag{6.4}$$

where ${}_G^L[\boldsymbol{R}]$ represents the rotation matrix from local coordinates to global ones. Now, the estimated joint axis can be used to correct the joint position coordinates along the corresponding sensor segment:

$$\hat{\boldsymbol{o}}_{1,proj} = \hat{\boldsymbol{o}}_1 - \hat{\boldsymbol{j}}_1 \frac{\hat{\boldsymbol{o}}_1 . \hat{\boldsymbol{j}}_1 + \hat{\boldsymbol{o}}_2 . \hat{\boldsymbol{j}}_2}{2}, \tag{6.5}$$

$$\hat{\boldsymbol{o}}_{2,proj} = \hat{\boldsymbol{o}}_2 - \hat{\boldsymbol{j}}_2 \frac{\hat{\boldsymbol{o}}_1 . \hat{\boldsymbol{j}}_1 + \hat{\boldsymbol{o}}_2 . \hat{\boldsymbol{j}}_2}{2}, \tag{6.6}$$

In other words, the point that is described by $\hat{\boldsymbol{o}}_i, i = 1,2$, is projected into the plane that is perpendicular to the joint axis $\hat{\boldsymbol{j}}_i, i = 1,2$ by the expression: $\hat{\boldsymbol{j}}_i \frac{\hat{\boldsymbol{o}}_1 . \hat{\boldsymbol{j}}_1 + \hat{\boldsymbol{o}}_2 . \hat{\boldsymbol{j}}_2}{2}$. Then, the position of $\hat{\boldsymbol{o}}_i$ is corrected in order to meet the plane defined by $\hat{\boldsymbol{j}}_i$. In figures 6.4 and 6.5, a graphical representation is provided to clarify the explanation.

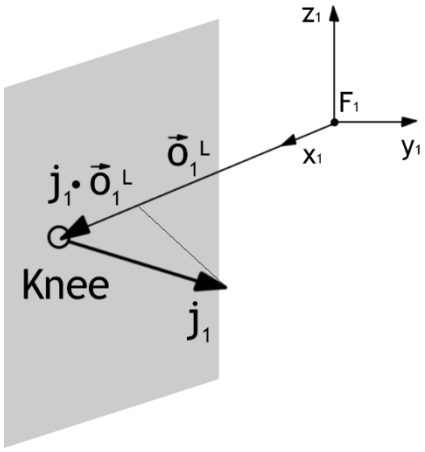

Figure 6.4 – Representation of the projection for the thigh.

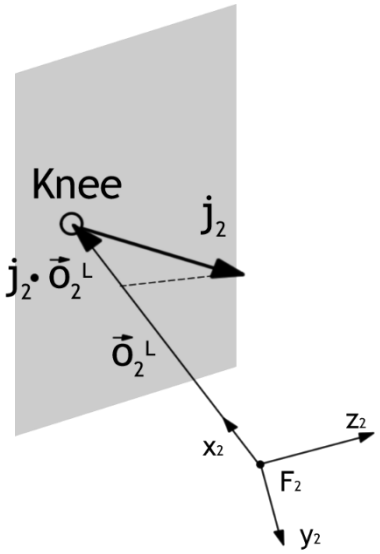

Figure 6.5 – Representation of the projection for the shank.

The algorithm is then implemented as follows:

1. Calculation of the rotation matrix provided in 2.12;
2. Calculation of $\vec{o}_1$ and $\vec{o}_2$ in local coordinates;
3. Transformation of $\vec{o}_1$, $\vec{o}_2$, $\vec{j}_1$ and $\vec{j}_2$ in global coordinates using the results of step 1;
4. Achievement of $\vec{o}_{1,proj}$ and $\vec{o}_{2,proj}$ using 6.5 and 6.6;

Similar to the algorithm for joint axis estimation, the above algorithm is easy to implement, when compared with other algorithms described in the literature for position estimation.

7.2- Experimental Results

Concerning the experimental validation, results were obtained from an individual walking at normal pace for approximately 4 seconds, in a straight line. An avatar was assembled, representing the leg, where green corresponds to the thigh and the blue to the shank. Further, these positions were represented graphically for a better assessment of the variation of position

over time. Moreover, it is also presented an avatar for hip flexion with knee flexion and knee flexion. The results achieved were not compared with the real position of the knee over time.

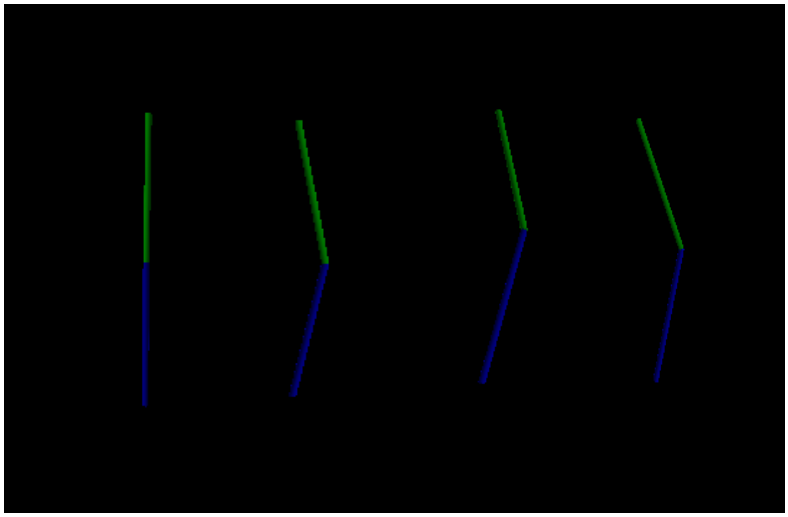

Figure 6.6 – Sequence of avatars obtained for gait. Left to right: standing phase, toe-off, midstance and terminal stance.

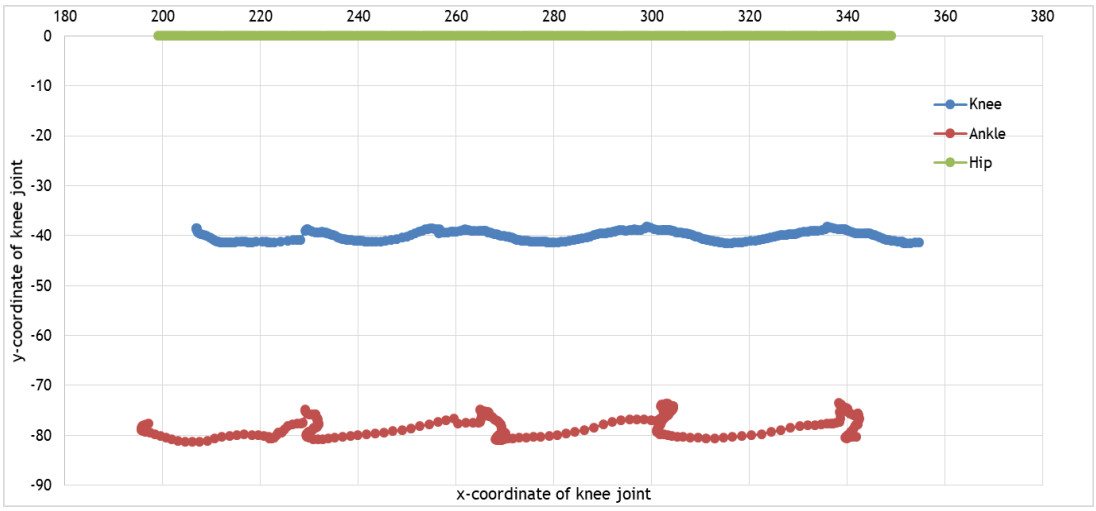

Figure 6.7 - Position of the hip, knee and ankle for walking for distinct time instants.

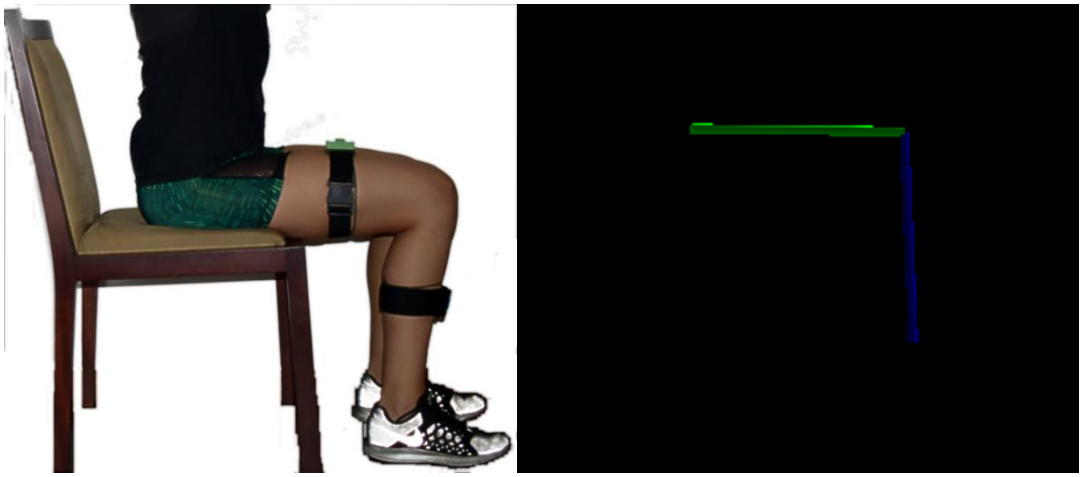

Figure 6.8 - Avatar obtained for sitting knee flexion.

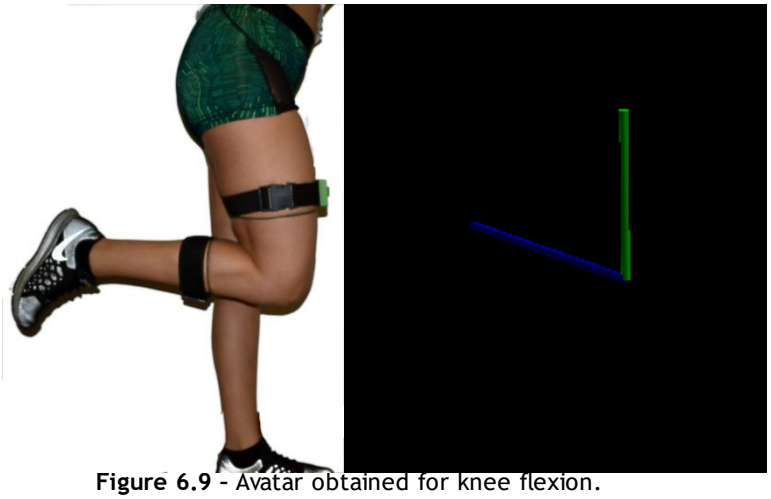

Figure 6.9 - Avatar obtained for knee flexion.

As it can be perceived by the graph of figure 6.7, considering that the hip is always in a fixed position, then over time the position of the knee and ankle will vary over time but in a proportional manner. In other words, when the y coordinate of the knee rises, the same coordinate of the ankle rises the same way, making the length of the leg constant. It is noteworthy that in both ends of the graph the three joints are not aligned, which is consistent with the human pace.

With these preliminary results, it can be concluded that the developed algorithm was successfully implemented and it is capable to provide sufficiently accurate knee joint positions.

Chapter 7

Conclusions and Future Work

The hereby thesis consisted of developing an algorithm capable of finding accurately the knee joint axis, independently of the sensor placement on the body segment. This project was purposed by the SWORD Health, a startup company focused on physical rehabilitation through the combination of science-driven therapeutic methods with effective technologies. A major problem in IMU-based human motion analysis is that the local coordinate axes of the IMUs are not aligned with any physiologically axes. Calibration postures and/or calibration movements are a common method to estimate these vectors, but in the context of rehabilitation, this is not appropriate for the patients because they may not be physically able to perform such movements. Therefore, there is high demand for methods that enable accurate estimation of joint axes with respect to the local sensor frames. Seel et al. [100] demonstrates in their work, how this information can be extracted from the measurement data of almost every arbitrary movements by exploiting the kinematic constraints of the respective joints. The algorithm explained in the present thesis is an adaptation of the first, with some adjustments to improve its robustness and accuracy and, at the same time, to be more suitable for the proposed problem.

In order to evaluate the algorithm, two types of validation were performed: one with simulated known data (an exact equation of the movement is considered) and another with real data acquired from the sensors. The parameter taken into consideration was the difference between ϕ_i^{True} and ϕ_i^{Sim} (for simulation known data) or ϕ_i^{Exp} (experimental data). Furthermore, for the simulated known data, it was evaluated the influence of noise (10% and 20%), the influence of the relative positions of the sensor and, also, the influence of tolerance (0.05, 0.01 and 0.001). In the random placement of the sensors, 300 runs were performed: 50 runs for 10% noise and 50 for 20% in each value of tolerance. On the other hand, for fixed placement of sensors, 25 combinations of sensors were evaluated and for each 10 runs were executed, yielding 1500 runs. Both validations demonstrated that the algorithm is capable to find the knee

joint axis for random and fixed placement of sensors. Further, it appears that there is not any relation between the smallest difference of ϕ and a preferred sensor placement.

Regarding the simulated known data, for both 10% and 20% noise, increasing tolerance implies a smaller number of iterations. In addition, the difference between ϕ_i^{True} and $\phi_i^{Sim}, i = 1,2$ is less than $2°$ for both cases of noise, which represents quite satisfactory results. Also, when the tolerance of the algorithm increases, the difference is higher, as expected. Concerning the experimental data, four different exercises were performed by a female individual: walking, sitting knee flexion with hip flexion, squat and sitting knee flexion. In this validation, only fixed combinations of sensors were evaluated. In the walking and sitting knee flexion with hip flexion, results were similar. As in the simulation, a correlation between the smallest difference and a preferred sensor placement was not found. For walking exercise, the higher difference obtained was less than $18°$ and the algorithm converged with less than 9 iterations. Regarding the latter exercise, the higher difference found was less than $15°$, obtained for an average of less than 7 iterations. Concerning the squat and sitting knee flexion some limitations of the algorithm were encountered. Since the algorithm relies of the angular velocities of both body segments to properly converge, when one of the segments does not have this physical quantity well defined, the algorithm will not arrive to a correct value. Nevertheless, for the body segment with a significant angular velocity, the algorithm is capable of a very satisfactory convergence. For the squat exercise, the thigh has angular velocity and so the higher difference obtained was less than $14°$ and for the second exercise the shank was the body segment that converged properly with a difference less than $12°$. In order to overcome this situation, the angular velocities of both body segments were analyzed in both exercises. This search led to the creation of ratio τ, which translates the relation between the angular velocity of the thigh and shank. Therefore, a final condition was implemented to the algorithm: if $\tau < 0.25$, the j of the knee is the j of the thigh, if $\tau > 6$, the j of the knee is the j of the shank, otherwise both j can be considered. Still regarding this last two exercises, due to this poor convergence for one of the body segments, occasionally, the algorithm entered an infinite loop. To prevent such, another condition was added: the algorithm will stop if $\|\vec{V}_{diff}\|$ is smaller than 0.01 or if the number of iterations is greater than 200. Thus, it is concluded that the algorithm is capable of a reliable convergence to the knee joint axis, independently of the sensor placement.

Some limitations were found throughout the study, namely, data must be rich enough in order to provide a good convergence. In exercises that do not have a well-defined movement of both segments, squat for instances, this issue was overcome but in exercises like hip abduction, this is impossible to solve (since it is relative rigid body motion, both segments present exactly the same angular acceleration, thus it is not possible to obtain the joint axis). In these cases, the algorithm will assume that the leg consists in only one segment and, thereby, convergence is unlikely. Another limitation encountered is that the sensors are not able to

provide information of quaternions and angular velocities at the same time. Thus, quaternions were used to obtain the angular velocities and, since the data was subjected to mathematical calculations, the angular velocities may be subjected to round errors.

In the future, it could be interesting to continue the work here started to improve the developed methodology. First, it could be performed a more comprehensive validation of the joint position estimation, in order to understand if the algorithm developed represents the true joint position. Also, implement the algorithm to the upper limbs, creating an algorithm for the whole human body and allowing, likewise, a random placement of sensors for the upper limbs. This is a very useful feature in the context of motor rehabilitation, especially when it is performed at home autonomously, since the patients do not need to concern about placing the sensors in a specific position. Furthermore, it would be important to extend the algorithm to other types of joints, not only hinge joints, then the placement of all sensors would not be an issue. Finally, it could be important to find a connection between the movement performed by the patient and the muscles involved in such movement. Through this information, reaction forces may be studied, allowing to perform computational mechanics analyses combining body motion with finite element analysis. This enhancement would allow to predict the stresses and strains in real-time in bone and muscles, permitting to avoid overload and to prevent lesions.

References

[1] T. Vos *et al.*, “Global, regional, and national incidence, prevalence, and years lived with disability for 301 acute and chronic diseases and injuries in 188 countries, 1990–2013: a systematic analysis for the Global Burden of Disease Study 2013,” *Lancet*, vol. 386, no. 9995, pp. 743–800, Jul. 2017.

[2] A. D. Woolf and B. Pfleger, “Burden of major musculoskeletal conditions,” *Bull. World Health Organ.*, vol. 81, no. 9, pp. 646–656, 2003.

[3] F. Coupar, A. Pollock, P. Rowe, C. Weir, and P. Langhorne, “Predictors of upper limb recovery after stroke: a systematic review and meta-analysis,” *Clin. Rehabil.*, vol. 26, no. 4, pp. 291–313, Oct. 2011.

[4] B. C. Bernhardt, K. J. Worsley, H. Kim, A. C. Evans, A. Bernasconi, and N. Bernasconi, “Longitudinal and cross-sectional analysis of atrophy in pharmacoresistant temporal lobe epilepsy,” *Neurology*, vol. 72, no. 20, pp. 1747–1754, May 2009.

[5] L. Vargas-Valencia, A. Elias, E. Rocon, T. Bastos-Filho, and A. Frizera, “An IMU-to-Body Alignment Method Applied to Human Gait Analysis,” *Sensors*, vol. 16, no. 12, p. 2090, 2016.

[6] T. Seel, “Learning Control and Inertial Realtime Gait Analysis in Biomedical Applications,” Technischen Universität Berlin, 2016.

[7] W. H. Organization, “World report on disability,” *Lancet*, vol. 377, no. 9782, p. 1977, 2011.

[8] K. Hashimoto, K. Higuchi, Y. Nakayama, and M. Abo, “Ability for Basic Movement as an Early Predictor of Functioning Related to Activities of Daily Living in Stroke Patients,” *Neurorehabil. Neural Repair*, vol. 21, no. 4, pp. 353–357, Mar. 2007.

[9] A. Muro-de-la-Herran, B. Garc??a-Zapirain, and A. M??ndez-Zorrilla, “Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications,” *Sensors (Switzerland)*, vol. 14, no. 2, pp. 3362–3394, 2014.

[10] W. Tao, T. Liu, R. Zheng, and H. Feng, “Gait analysis using wearable sensors,” *Sensors*, vol. 12, no. 2, pp. 2255–2283, 2012.

[11] L. Ren, Z. Qian, and L. Ren, “Biomechanics of musculoskeletal system and its biomimetic implications: A review,” *J. Bionic Eng.*, vol. 11, no. 2, pp. 159–175, 2014.

[12] M. G. Pandy, “Computer Modeling and Simulation of Human Movement,” *Kinesiology*, vol. 3, no. 1, pp. 245–273, 2001.

[13] R. D. Sinelnikov, *ATLAS OF HUMAN ANATONY. Volume I - Musculoskeletal System*, vol. I. MIR Publisher.

[14] G. S. Firestein, R. C. Budd, S. E. Gabriel, I. B. McInnes, and J. R. O'Dell, *Kelley's Textbbok of Rheumatology*, Ninth Edit. ELSEVIER Saunders, 2012.

[15] C. Hartmann, "Transcriptional networks controlling skeletal development," *Curr. Opin. Genet. Dev.*, vol. 19, no. 5, pp. 437–443, 2009.

[16] G. G. Karsenty, H. M. Kronenberg, and C. Settembre, "Genetic control of bone formation.," *Annu. Rev. Cell Dev. Biol.*, vol. 25, pp. 629–48, 2009.

[17] I. Takada, A. P. Kouzmenko, and S. Kato, "Wnt and PPAR[gamma] signaling in osteoblastogenesis and adipogenesis," *Nat Rev Rheumatol*, vol. 5, no. 8, pp. 442–447, Aug. 2009.

[18] B. R. Olse, A. M. Reginato, and W. Wang, "Bone Development," *Annu. Rev. Cell Dev.*, pp. 191–220, 2000.

[19] S. L.-Y. Woo, S. D. Abramowitch, R. Kilger, and R. Liang, "Biomechanics of knee ligaments: injury, healing, and repair," *J. Biomech.*, vol. 39, no. 1, pp. 1–20, 2006.

[20] J. O. Holloszy and E. F. Coyle, "Adaptations of skeletal muscle to endurance exercise and their metabolic consequences," *J. Appl. Physiol.*, vol. 56, no. 4, p. 831 LP-838, Apr. 1984.

[21] K. M. Van De Graaff, *Human Anatomy*. McGraw-Hill, 2000.

[22] J. R. Andrews, G. L. Harrelson, and K. E. Wilk, *Physical Rehabilitation of the Injured Athlete*, Fourth Edi. ELSEVIER Saunders, 2012.

[23] M. E. H. Larsson and L. a Nordholm, "Responsibility for managing musculoskeletal disorders--a cross-sectional postal survey of attitudes.," *BMC Musculoskelet. Disord.*, vol. 9, p. 110, 2008.

[24] V. Y. Ma, L. Chan, and K. J. Carruthers, "Incidence, prevalence, costs, and impact on disability of common conditions requiring rehabilitation in the united states: Stroke, spinal cord injury, traumatic brain injury, multiple sclerosis, osteoarthritis, rheumatoid arthritis, limb loss, and back pa," *Arch. Phys. Med. Rehabil.*, vol. 95, no. 5, p. 986–995.e1, 2014.

[25] S. von Campenhausen *et al.*, "Prevalence and incidence of Parkinson's disease in Europe," *Eur. Neuropsychopharmacol.*, vol. 15, no. 4, pp. 473–490, 2005.

[26] A. S. Go *et al.*, "Heart Disease and Stroke Statistics—2014 Update," *Circulation*, vol. 129, no. 3, p. e28 LP-e292, Jan. 2014.

[27] D. Mozaffarian *et al.*, "Heart Disease and Stroke Statistics—2016 Update," *Circulation*, vol. 133, no. 4, p. e38 LP-e360, Jan. 2016.

[28] D. Lloyd-Jones *et al.*, "Executive Summary: Heart Disease and Stroke Statistics—2010 Update," *Circulation*, vol. 121, no. 7, p. 948 LP-954, Feb. 2010.

[29] D. S. Nichols-Larsen, P. C. Clark, A. Zeringue, A. Greenspan, and S. Blanton, "Factors Influencing Stroke Survivors' Quality of Life During Subacute Recovery," *Stroke*, vol. 36, no. 7, p. 1480 LP-1484, Jun. 2005.

[30] S.-M. Lai, S. Studenski, P. W. Duncan, and S. Perera, "Persisting Consequences of Stroke Measured by the Stroke Impact Scale," *Stroke*, vol. 33, no. 7, p. 1840 LP-1844, Jul. 2002.

[31] N. Shishov, I. Melzer, and S. Bar-Haim, "Parameters and Measures in Assessment of Motor Learning in Neurorehabilitation; A Systematic Review of the Literature," *Front. Hum. Neurosci.*, vol. 11, no. February, p. 82, 2017.

[32] J. Luker, E. Lynch, S. Bernhardsson, L. Bennett, and J. Bernhardt, "Stroke Survivors' Experiences of Physical Rehabilitation: A Systematic Review of Qualitative Studies," *Arch. Phys. Med. Rehabil.*, vol. 96, no. 9, p. 1698–1708e10, 2015.

[33] T. Sugavanam, G. Mead, C. Bulley, M. Donaghy, and F. van Wijck, "The effects and experiences of goal setting in stroke rehabilitation – a systematic review," *Disabil. Rehabil.*, vol. 35, no. 3, pp. 177–190, Feb. 2013.

[34] A. Russo and R. Andrews, "STATISTICAL BRIEF # 51 Cerebrovascular Diseases , 2005," *Heal. (San Fr.*, pp. 1–11, 2008.

[35] T. N. Taylor, P. H. Davis, J. C. Torner, J. Holmes, J. W. Meyer, and M. F. Jacobson, "Lifetime Cost of Stroke in the United States," *Stroke*, vol. 27, no. 9, p. 1459 LP-1466, Sep. 1996.

[36] A. Gustavsson *et al.*, "Cost of disorders of the brain in Europe 2010," *Eur. Neuropsychopharmacol.*, vol. 21, no. 10, pp. 718–779, 2011.

[37] S. von Campenhausen *et al.*, "Costs of illness and care in Parkinson's Disease: An evaluation in six countries," *Eur. Neuropsychopharmacol.*, vol. 21, no. 2, pp. 180–191, 2011.

[38] N. M. van der Kolk and L. A. King, "Effects of exercise on mobility in people with Parkinson's disease," *Mov. Disord.*, vol. 28, no. 11, pp. 1587–1596, 2013.

[39] J. E. Ahlskog, "Does vigorous exercise have a neuroprotective effect in Parkinson disease?," *Neurology*, vol. 77, no. 3, pp. 288–294, Jul. 2011.

[40] C. L. Tomlinson *et al.*, "Physiotherapy versus placebo or no intervention in Parkinson's disease.," *Cochrane database Syst. Rev.*, no. 7, p. CD002817, Jul. 2012.

[41] G. Abbruzzese, R. Marchese, L. Avanzino, and E. Pelosin, "Rehabilitation for Parkinson's disease: Current outlook and future challenges," *Park. Relat. Disord.*, vol. 22, pp. S60–S64, 2016.

[42] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, no. 10, pp. 3605–3620, 2010.

[43] A. Yurtman and B. Barshan, "Automated evaluation of physical therapy exercises using multi-template dynamic time warping on wearable sensor signals," *Comput. Methods Programs Biomed.*, vol. 117, no. 2, pp. 189–207, 2014.

[44] A. M. Sabatini, "Inertial Sensing in Biomechanics," in *http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59140-836-9.ch002*, IGI Global,

1AD, pp. 70–100.

[45] P.-C. Lin, H. Komsuoglu, and D. E. Koditschek, "Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits," *IEEE Transactions on Robotics*, vol. 22, no. 5. pp. 932–943, 2006.

[46] G. Secer and B. Barshan, "Improvements in deterministic error modeling and calibration of inertial sensors and magnetometers," *Sensors Actuators, A Phys.*, vol. 247, pp. 522–538, 2016.

[47] D. Titterton, J. L. Weston, and I. of Electrical Engineers, *Strapdown Inertial Navigation Technology*. Institution of Engineering and Technology, 2004.

[48] T. Meydan, "Recent trends in linear and angular accelerometers," *Sensors Actuators A Phys.*, vol. 59, no. 1, pp. 43–50, 1997.

[49] O. Sprdlik, "Detection and Estimation of Human Movement Using Inertial Sensors: Applications in Neurology," Czech Technical University in Prague, 2012.

[50] J. J. Kavanagh and H. B. Menz, "Accelerometry: A technique for quantifying movement patterns during walking," *Gait Posture*, vol. 28, no. 1, pp. 1–15, 2008.

[51] W. H. Baird, "An introduction to inertial navigation," *Am. J. Phys.*, vol. 77, no. 9, pp. 844–847, 2009.

[52] A. Nez, L. Fradet, P. Laguillaumie, T. Monnet, and P. Lacouture, "Comparison of calibration methods for accelerometers used in human motion analysis," *Med. Eng. Phys.*, vol. 38, no. 11, pp. 1289–1299, 2016.

[53] S. Levy, A. E. McPherson, and E. V Hobbs, "Calibration of accelerometers," *J. Res. Nat. Bur. Stand.*, vol. 41, no. November, pp. 359–369, 1948.

[54] G. LANGFELDER and A. TOCCHIO, *MEMS integrating motion and displacement sensors*. Woodhead Publishing Limited, 2014.

[55] T. Tamura, "Wearable Inertial Sensors and Their Applications," *Wearable Sensors*, no. 1, pp. 85–104, 2014.

[56] N. C. Tsai and C. Y. Sue, "Design and analysis of a tri-axis gyroscope micromachined by surface fabrication," *IEEE Sens. J.*, vol. 8, no. 12, pp. 1933–1940, 2008.

[57] M. Iosa, P. Picerno, S. Paolucci, and G. Morone, "Wearable inertial sensors for human movement analysis," *Expert Rev. Med. Devices*, vol. 13, no. 7, pp. 641–659, 2016.

[58] L. Ojeda, H. Chung, and J. Borenstein, "Precision calibration of fiber-optics gyroscopes for mobile robot navigation," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 3. pp. 2064–2069 vol.3, 2000.

[59] M. M. Tehrani, "Ring Laser Gyro Data Analysis With Cluster Sampling Technique," *Proc. SPIE*, vol. 412. pp. 207–220, 1983.

[60] Y. Li, X. Niu, Q. Zhang, H. Zhang, and C. Shi, "An *in situ* hand calibration method using a pseudo-observation scheme for low-end inertial measurement units," *Meas. Sci.*

Technol., vol. 23, no. 10, p. 105104, Oct. 2012.

[61] R. Khosla, *Knowledge-Based Intelligent Information and Engineering Systems: 9th International Conference, KES 2005, Melbourne, Australia, September 14-16, 2005, Proceedings*, no. pt. 1. Springer Berlin Heidelberg, 2005.

[62] B. Li, J. Lu, W. Xiao, and T. Lin, "In-field fast calibration of FOG-based MWD IMU for horizontal drilling," *Meas. Sci. Technol.*, vol. 26, no. 3, p. 35001, Mar. 2015.

[63] H. Emmerich and M. Schofthaler, "Magnetic field measurements with a novel surface micromachined magnetic-field sensor," *IEEE Transactions on Electron Devices*, vol. 47, no. 5. pp. 972–977, 2000.

[64] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, "Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2. pp. 1293–1306, 2011.

[65] C. C. Foster and G. H. Elkaim, "Extension of a two-step calibration methodology to include nonorthogonal sensor axes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3. pp. 1070–1078, 2008.

[66] D. Gebre-Egziabher, G. H. Elkaim, J. David Powell, and B. W. Parkinson, "Calibration of Strapdown Magnetometers in Magnetic Field Domain," *J. Aerosp. Eng.*, vol. 19, no. April, pp. 87–102, 2006.

[67] D. Gebre-Egziabher, G. Elkaim, J. Powell, and B. Parkinson, "A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers.," *8th Int. Conf. Integr. Navig. Syst.*, pp. 290–297, 2001.

[68] J. Metge, R. Mégret, A. Giremus, Y. Berthoumieu, and T. Décamps, "Calibration of an inertial-magnetic measurement unit without external equipment, in the presence of dynamic magnetic disturbances," *Meas. Sci. Technol.*, vol. 25, no. 12, p. 125106, Dec. 2014.

[69] P. Picerno, "25 years of lower limb joint kinematics by using inertial and magnetic sensors: A review of methodological approaches," *Gait Posture*, vol. 51, pp. 239–246, 2017.

[70] A. Cappozzo, U. Della Croce, A. Leardini, and L. Chiari, "Human movement analysis using stereophotogrammetry. Part 1: Theoretical background," *Gait Posture*, vol. 21, no. 2, pp. 186–196, 2005.

[71] V. Cimolin and M. Galli, "Summary measures for clinical gait analysis: A literature review," *Gait Posture*, vol. 39, no. 4, pp. 1005–1010, Apr. 2014.

[72] L. K. Drewes *et al.*, "Altered Ankle Kinematics and Shank-Rear-Foot Coupling in Those with Chronic Ankle Instability," *J. Sport Rehabil.*, vol. 18, no. 3, pp. 375–388, Aug. 2009.

[73] S.-L. Chiu, T.-W. Lu, and L.-S. Chou, "Altered inter-joint coordination during walking in patients with total hip arthroplasty," *Gait Posture*, vol. 32, no. 4, pp. 656–660, Oct. 2010.

[74] A. Winogrodzka, R. C. Wagenaar, J. Booij, and E. C. Wolters, “Rigidity and bradykinesia reduce interlimb coordination in Parkinsonian gait,” *Arch. Phys. Med. Rehabil.*, vol. 86, no. 2, pp. 183–189, Feb. 2005.

[75] G. Kwakkel and R. C. Wagenaar, “Effect of Duration of Upper- and Lower-Extremity Rehabilitation Sessions and Walking Speed on Recovery of Interlimb Coordination in Hemiplegic Gait,” *Phys. Ther.*, vol. 82, no. 5, p. 432, 2002.

[76] J. R. W. Morris, “Accelerometry—A technique for the measurement of human body movements,” *J. Biomech.*, vol. 6, no. 6, pp. 729–736, 1973.

[77] J. Bortz, “A New Mathematical Formulation for Strapdown Inertial Navigation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-7, no. 1, pp. 61–66, 1971.

[78] A. T. M. Willemsen, J. A. van Alsté, and H. B. K. Boom, “Real-time gait assessment utilizing a new way of accelerometry,” *J. Biomech.*, vol. 23, no. 8, pp. 859–863, 1990.

[79] T. Beth, I. Boesnach, M. Haimerl, J. Moldenhauer, K. Bös, and V. Wank, “Characteristics in human motion – From acquisition to analysis,” *Int. Conf. Humanoid Robot. (Humanoids), Karlsruhe/M\"unchen, Ger.*, no. July 2017, 2003.

[80] H. Zhou and H. Hu, “Human motion tracking for rehabilitation-A survey,” *Biomed. Signal Process. Control*, vol. 3, no. 1, pp. 1–18, 2008.

[81] H. I. Krebs, B. T. Volpe, M. L. Aisen, and N. Hogan, “Increasing productivity and quality of care: Robot-aided neuro-rehabilitation,” *J. Rehabil. Res. Dev.*, vol. 37, no. 6, p. 639, 2000.

[82] C. G. Burgar, P. S. Lum, P. C. Shor, and H. F. Machiel Van der Loos, “Development of robots for rehabilitation therapy: the Palo Alto VA/Stanford experience.,” *J. Rehabil. Res. Dev.*, vol. 37, no. 6, pp. 663–73, 2000.

[83] D. J. Reinkensmeyer, L. E. Kahn, M. Averbuch, A. McKenna-Cole, B. D. Schmit, and W. Z. Rymer, “Understanding and treating arm movement impairment after chronic brain injury: progress with the ARM guide.” 2014.

[84] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker, “Survey of Motion Tracking Methods Based on Inertial Sensors: A Focus on Upper Limb Human Motion,” *Sensors*, vol. 17, no. 6, p. 1257, 2017.

[85] S. O. H. Madgwick, “An efficient orientation filter for inertial and inertial/magnetic sensor arrays,” *Rep. x-io Univ. …*, p. 32, 2010.

[86] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, 2002.

[87] X. Yun and E. R. Bachmann, “Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking,” *IEEE Transactions on Robotics*, vol. 22, no. 6. pp. 1216–1227, 2006.

[88] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, pp. 1–35, 2006.

[89] E. R. Bachmann, I. Duman, U. Y. Usta, R. B. McGhee, X. P. Yun, and M. J. Zyda, "Orientation Tracking for Humans and Robots Using Inertial Sensors," *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, pp. 187–194, 1999.

[90] J. Hamill and K. M. Knutzen, *Biomechanical Basis of Human Movement*. Lippincott Williams & Wilkins, 2006.

[91] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens MVN: full 6DOF human motion tracking using miniature inertial sensors," *Xsens Motion Technol.*, pp. 1–9, 2009.

[92] E. Palermo, S. Rossi, F. Marini, F. Patan??, and P. Cappa, "Experimental evaluation of accuracy and repeatability of a novel body-to-sensor calibration procedure for inertial sensor-based gait analysis," *Meas. J. Int. Meas. Confed.*, vol. 52, no. 1, pp. 145–155, 2014.

[93] H. J. Luinge, P. H. Veltink, and C. T. M. Baten, "Ambulatory measurement of arm orientation," *J. Biomech.*, vol. 40, no. 1, pp. 78–85, 2007.

[94] H. J. Luinge and P. H. Veltink, "Inclination measurement of human movement using a 3-D accelerometer with autocalibration.," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 12, no. 1, pp. 112–121, 2004.

[95] A. Gelb, *Applied Optimal Estimation*. MIT Press, 1974.

[96] M. Kok, J. D. Hol, and T. B. Sch, "An optimization-based approach to human body motion capture using inertial sensors," 2014.

[97] P. G. Savage, "Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms," *J. Guid. Control. Dyn.*, vol. 21, no. 1, pp. 19–28, Jan. 1998.

[98] M. Kok, J. D. Hol, and T. B. Schön, "An optimization-based approach to human body motion capture using inertial sensors," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 79–85, 2014.

[99] K. Liu, T. Liu, K. Shibata, Y. Inoue, and R. Zheng, "Novel approach to ambulatory assessment of human segmental orientation on a wearable sensor system," *J. Biomech.*, vol. 42, no. 16, pp. 2747–2752, 2009.

[100] T. Seel, T. Schauer, and J. Raisch, "Joint axis and position estimation from inertial measurement data by exploiting kinematic constraints," *Proc. IEEE Int. Conf. Control Appl.*, pp. 45–49, 2012.

Appendixes

Appendix A: Simulated signal for all different combinations of sensors

The following figures represent the input signals for the algorithm in the simulated data validation chapter, for all possible combinations and for the two values of noise considered.

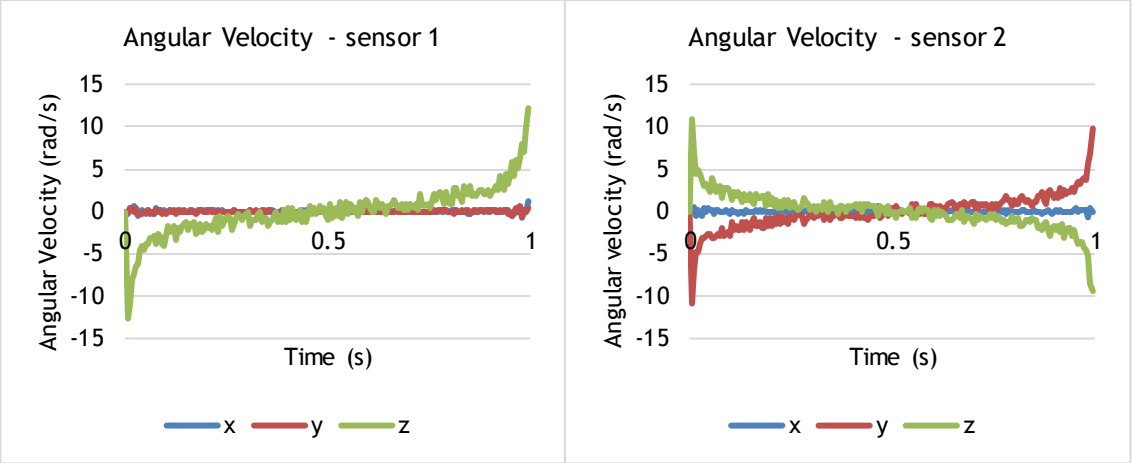

Figure A.1 - Input signal for the simulated data validation. $\phi_1 = -90°, \phi_1 = -45°$, noise 10%.

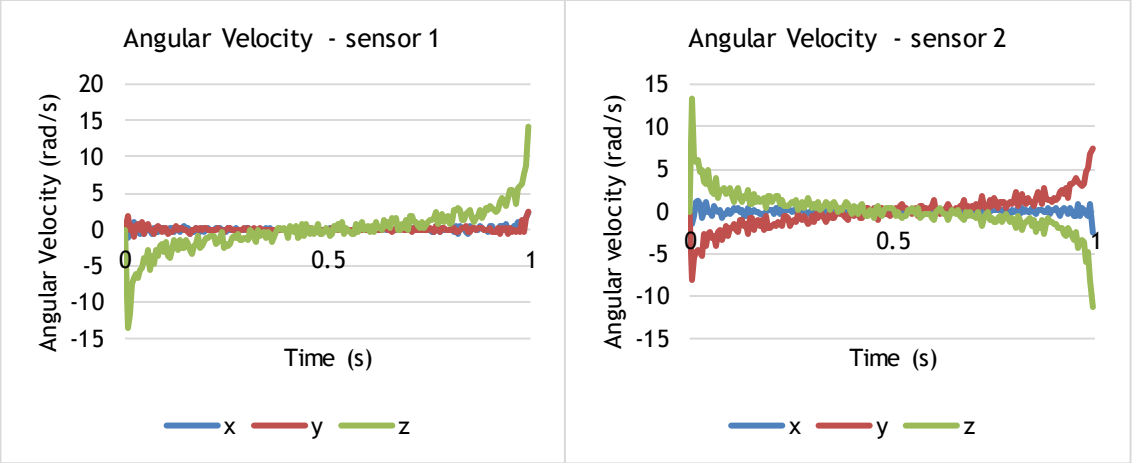

Figure A.2 - Input signal for the simulated data validation. $\phi_1 = -90°, \phi_1 = -45°$, noise 20%.

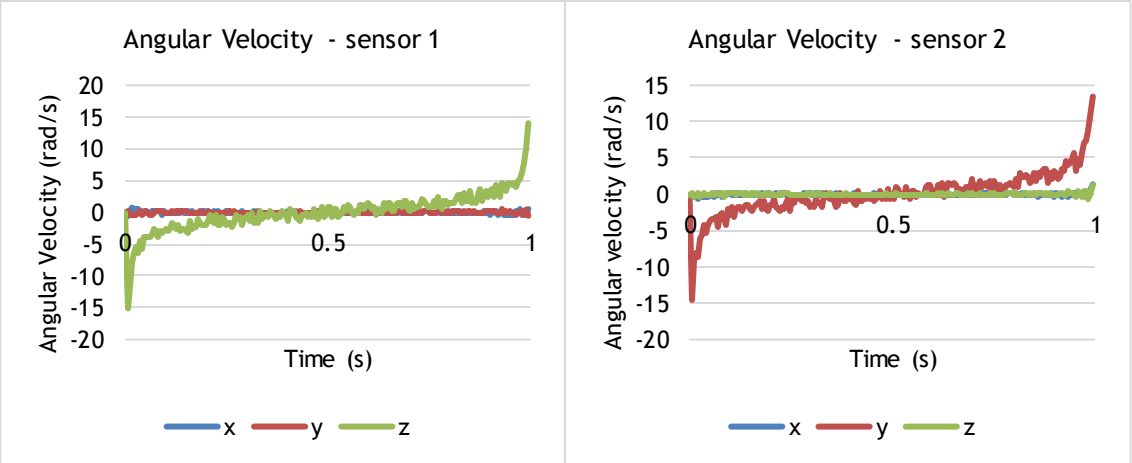

Figure A.3 - Input signal for the simulated data validation. $\phi_1 = -90°, \phi_1 = 0°$, noise 10%.

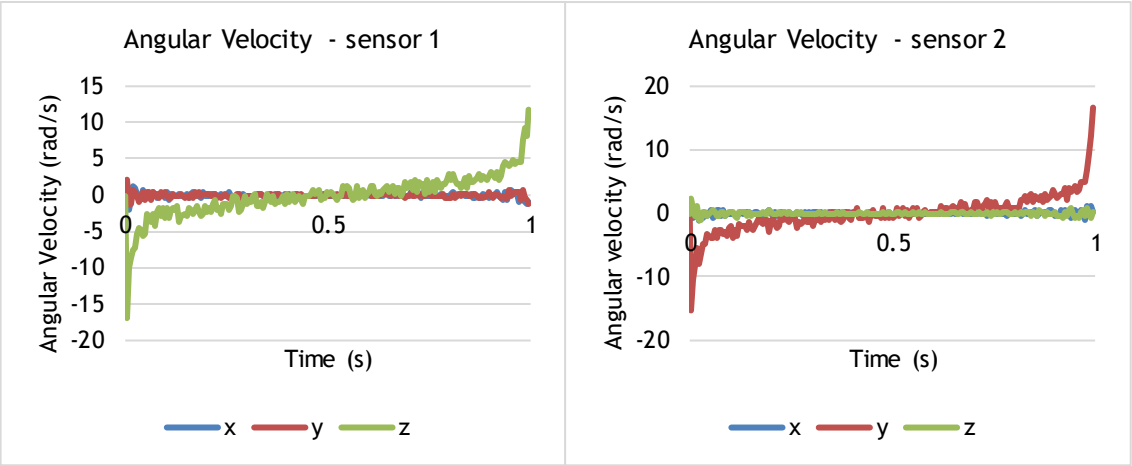

Figure A.4 - Input signal for the simulated data validation. $\phi_1 = -90°, \phi_1 = 0°$, noise 20%.

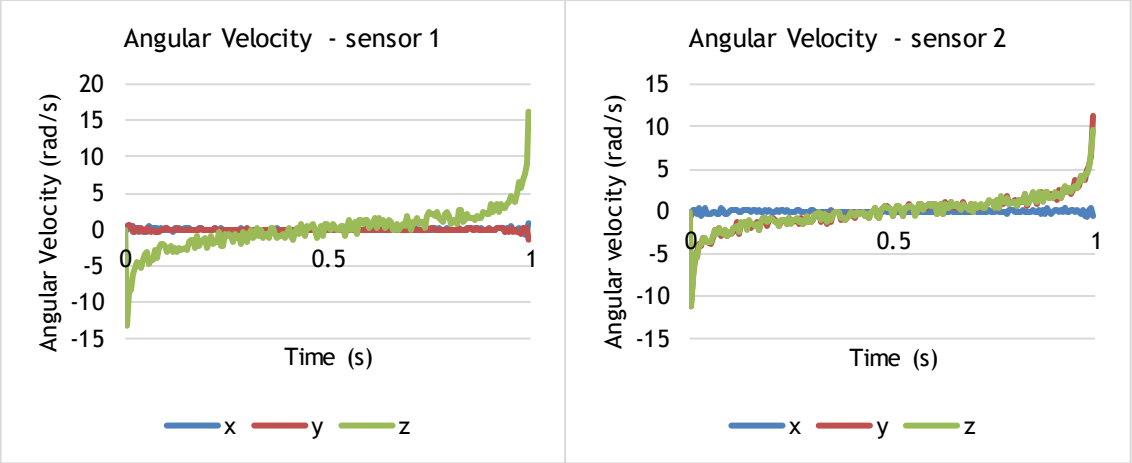

Figure A.5 - Input signal for the simulated data validation. $\phi_1 = -90°, \phi_1 = 45°$, noise 10%.

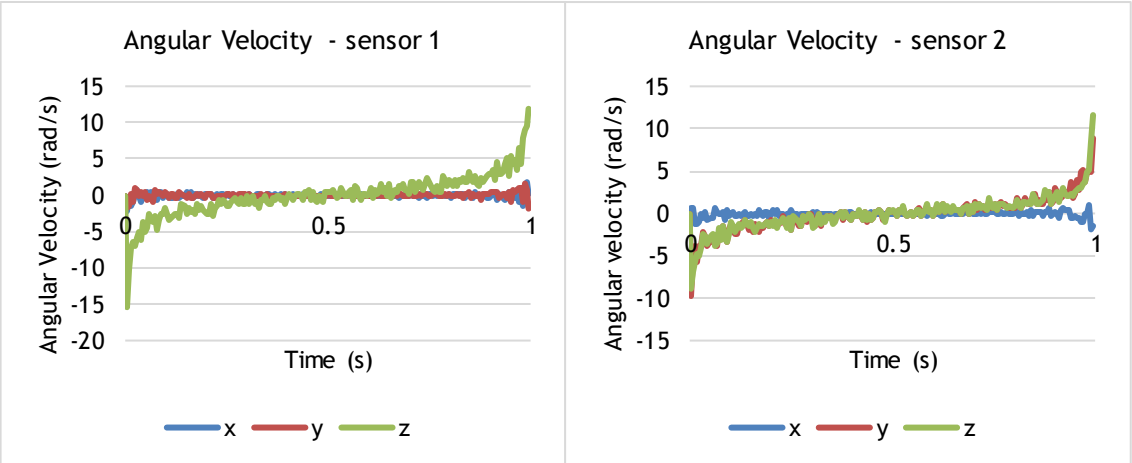

Figure A.6 - Input signal for the simulated data validation. $\phi_1 = -90°, \phi_1 = 45°$, noise 20%.

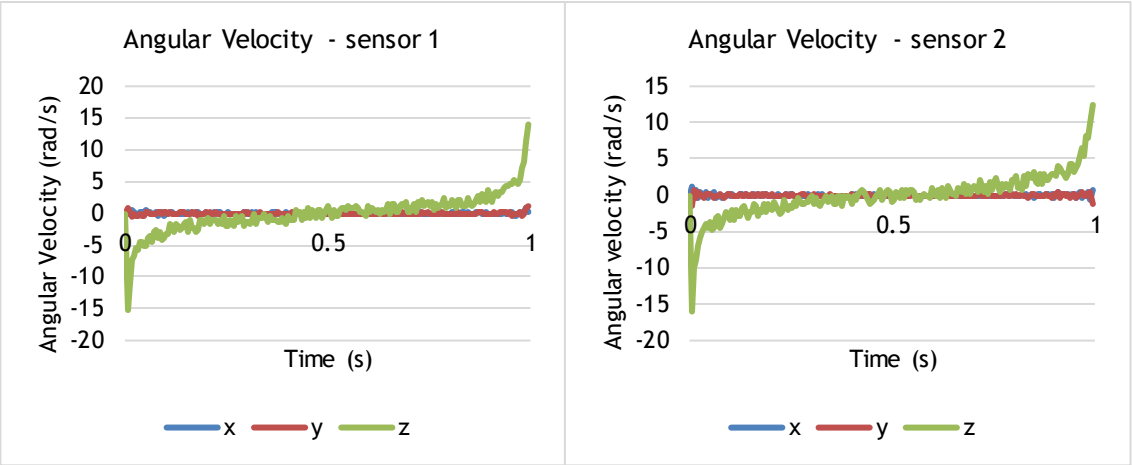

Figure A.7 - Input signal for the simulated data validation. $\phi_1 = -90°, \phi_1 = 90°$, noise 10%.

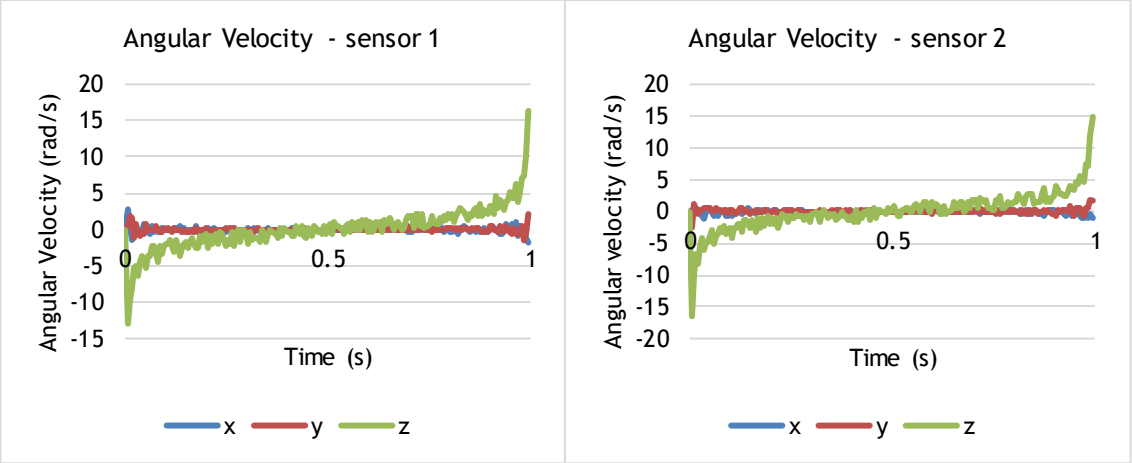

Figure A.8 - Input signal for the simulated data validation. $\phi_1 = -90°, \phi_1 = 90°$, noise 20%.

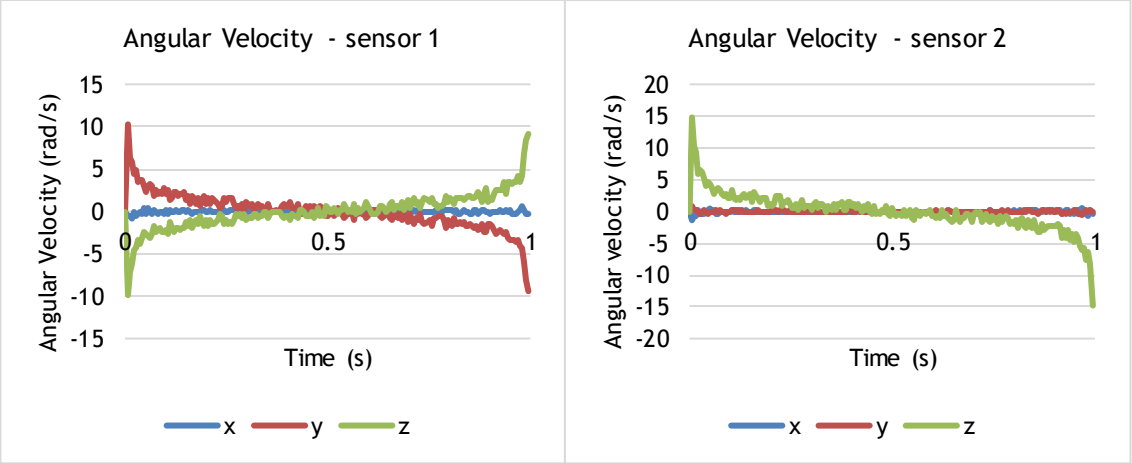

Figure A.9 - Input signal for the simulated data validation. $\phi_1 = -45°, \phi_1 = -90°$, noise 10%

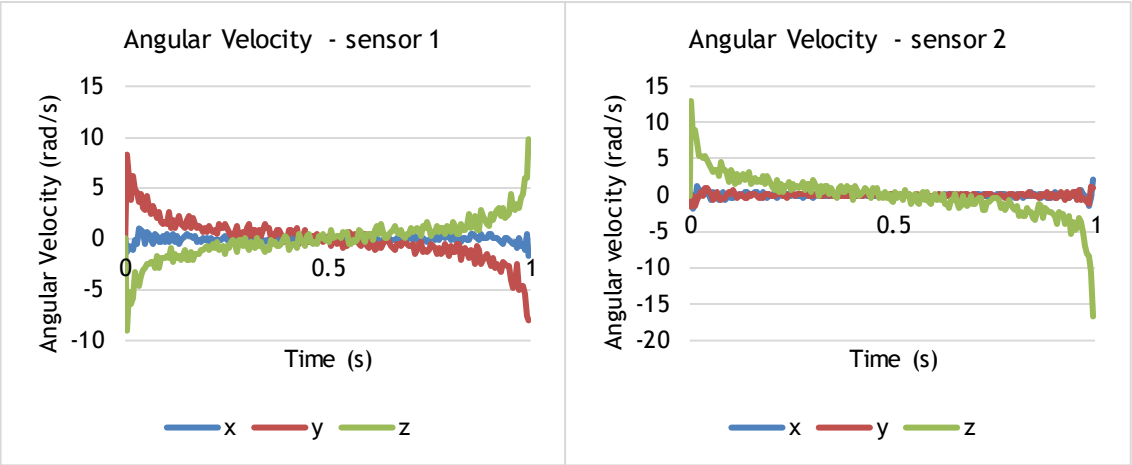

Figure A.10 - Input signal for the simulated data validation. $\phi_1 = -45°, \phi_1 = -90°$, noise 20%

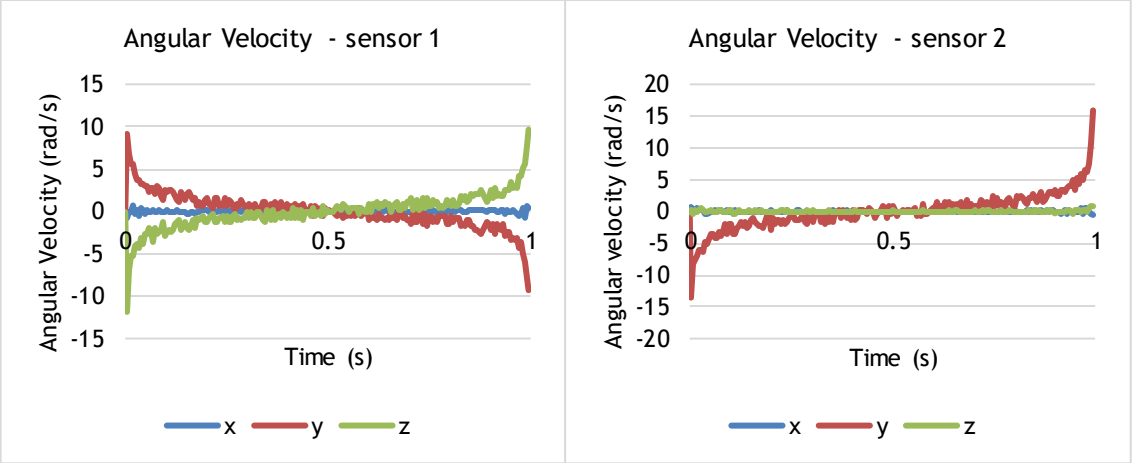

Figure A.11 - Input signal for the simulated data validation. $\phi_1 = -45°, \phi_1 = 0°$, noise 10%

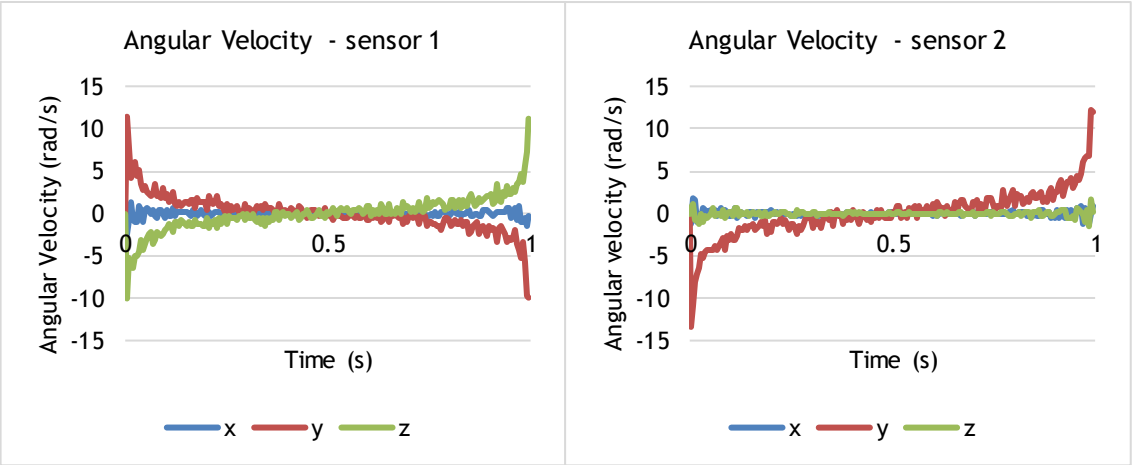

Figure A.12 - Input signal for the simulated data validation. $\phi_1 = -45°, \phi_1 = 0°$, noise 20%

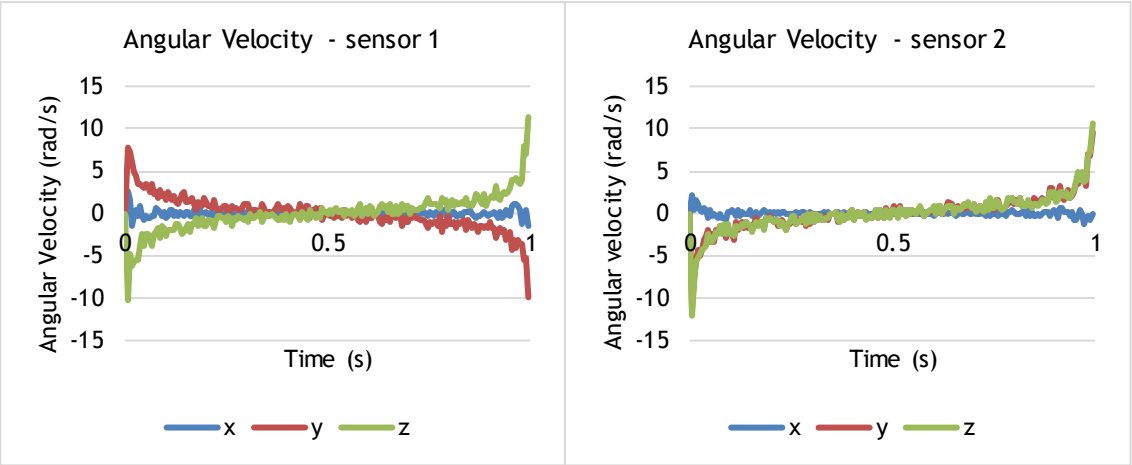

Figure A.13 - Input signal for the simulated data validation. $\phi_1 = -45°, \phi_1 = 45°$, noise 10%

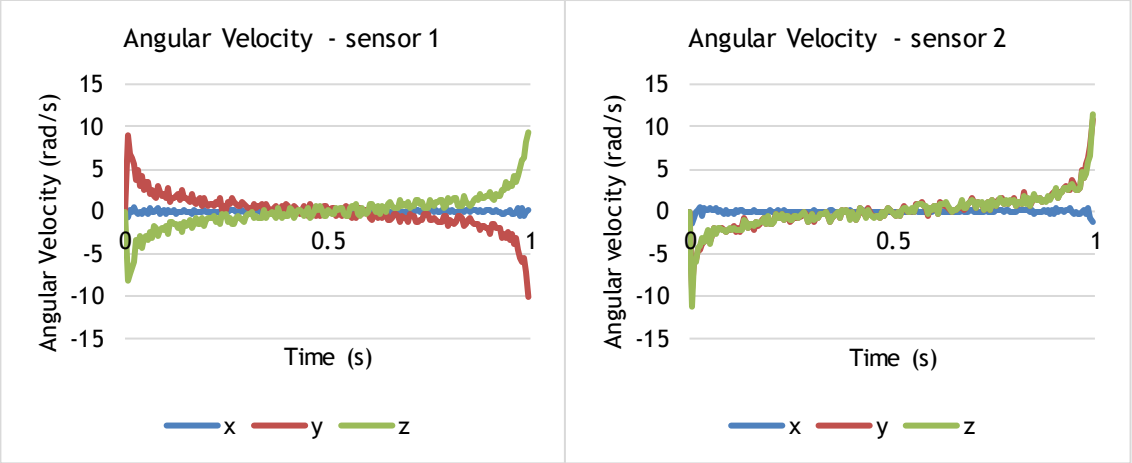

Figure A.14 - Input signal for the simulated data validation. $\phi_1 = -45°, \phi_1 = 45°$, noise 20%

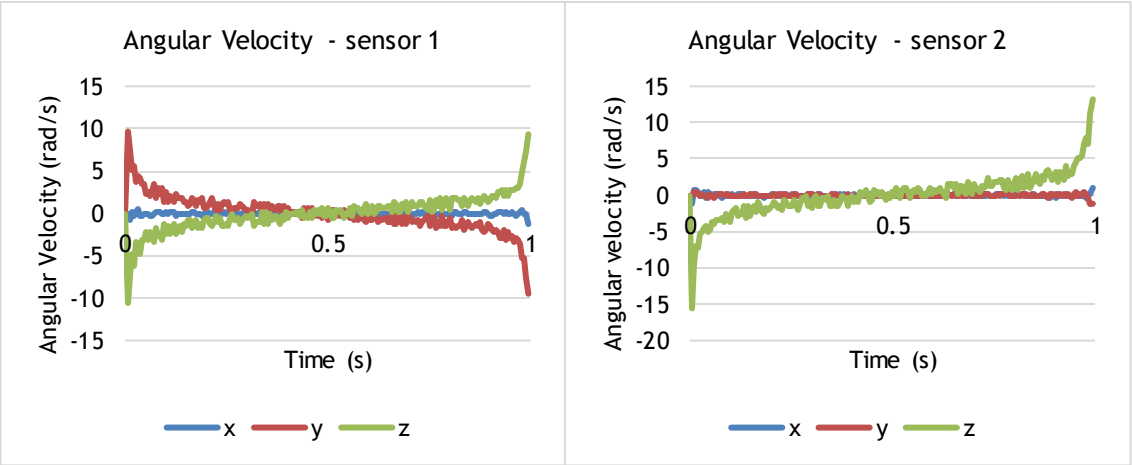

Figure A.15 - Input signal for the simulated data validation. $\phi_1 = -45°, \phi_1 = 90°$, noise 10%

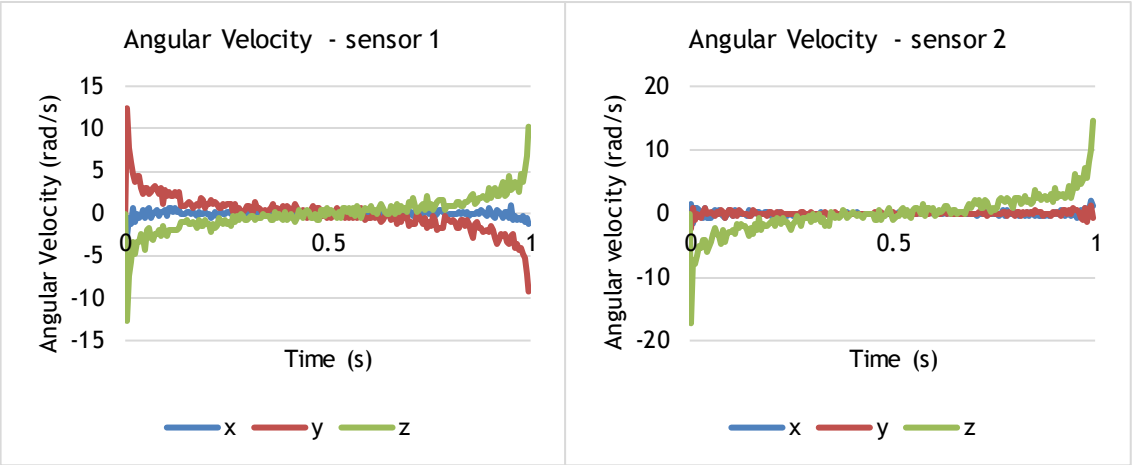

Figure A.16 - Input signal for the simulated data validation. $\phi_1 = -45°, \phi_1 = 90°$, noise 20%

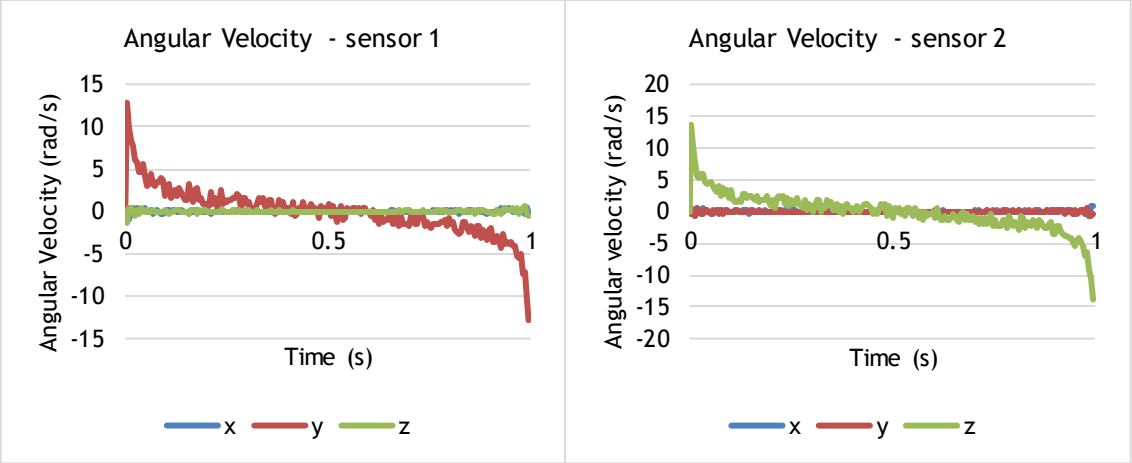

Figure A.17 - Input signal for the simulated data validation. $\phi_1 = 0°, \phi_1 = -90°$, noise 10%

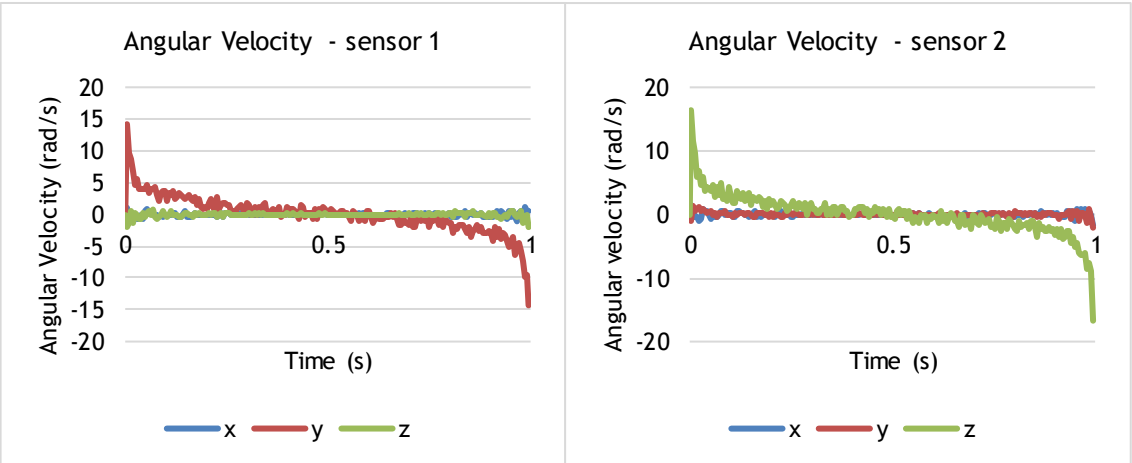

Figure A.18 - Input signal for the simulated data validation. $\phi_1 = 0°, \phi_1 = -90°$, noise 20%

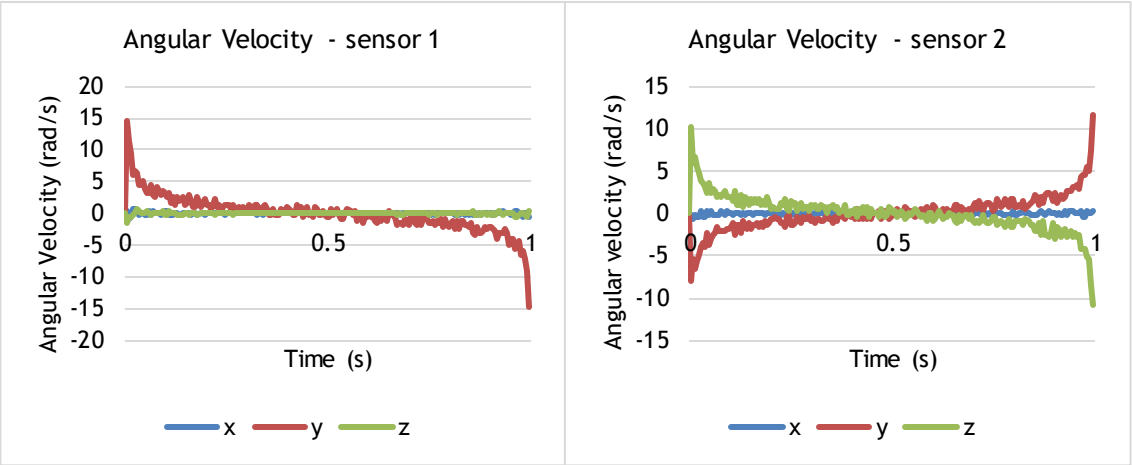

Figure A.19 - Input signal for the simulated data validation. $\phi_1 = 0°, \phi_1 = -45°$, noise 10%

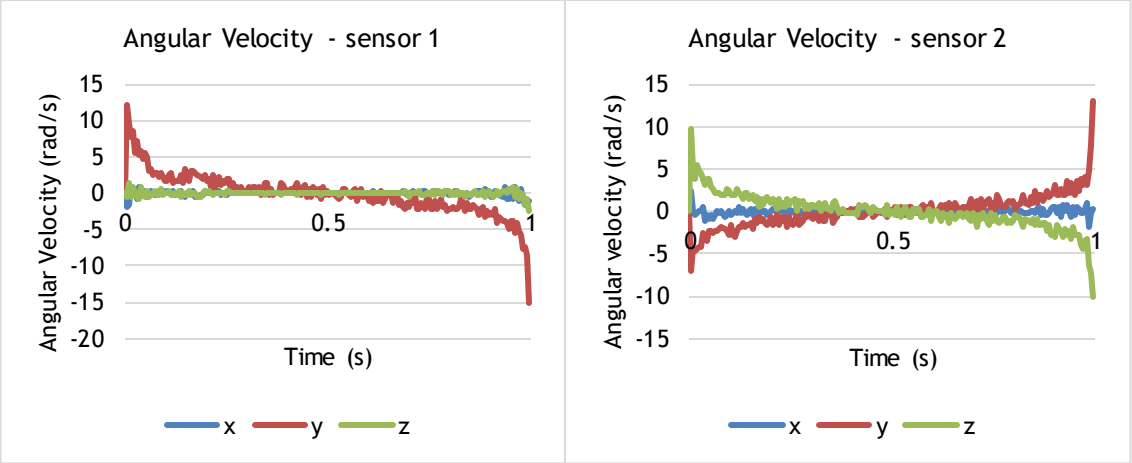

Figure A.20 - Input signal for the simulated data validation. $\phi_1 = 0°, \phi_1 = -45°$, noise 20%

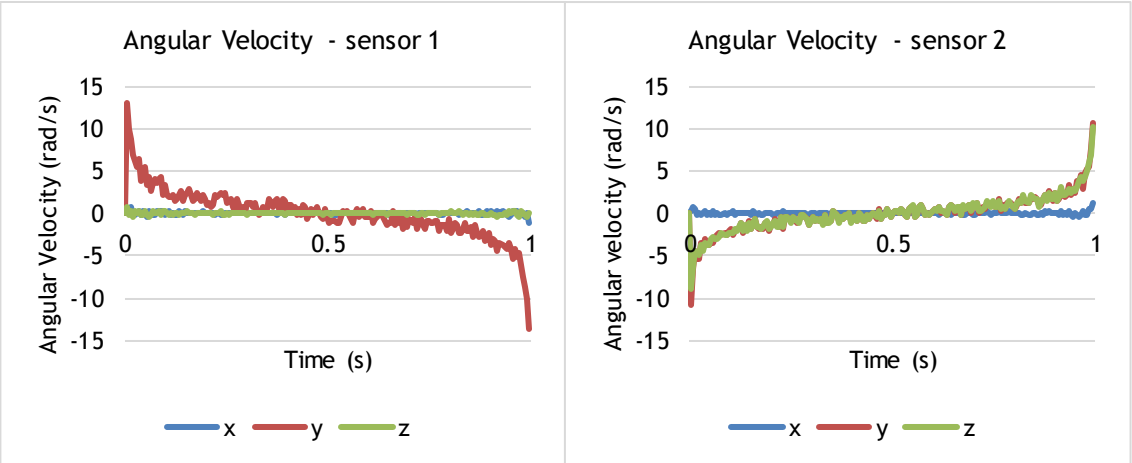

Figure A.21 - Input signal for the simulated data validation. $\phi_1 = 0°, \phi_1 = 45°$, noise 10%

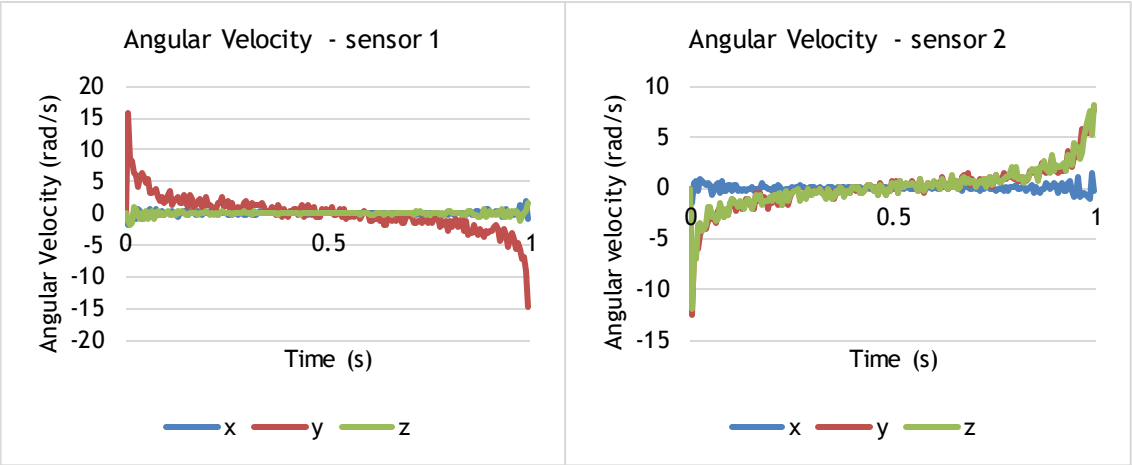

Figure A.22 - Input signal for the simulated data validation. $\phi_1 = 0°, \phi_1 = 45°$, noise 20%

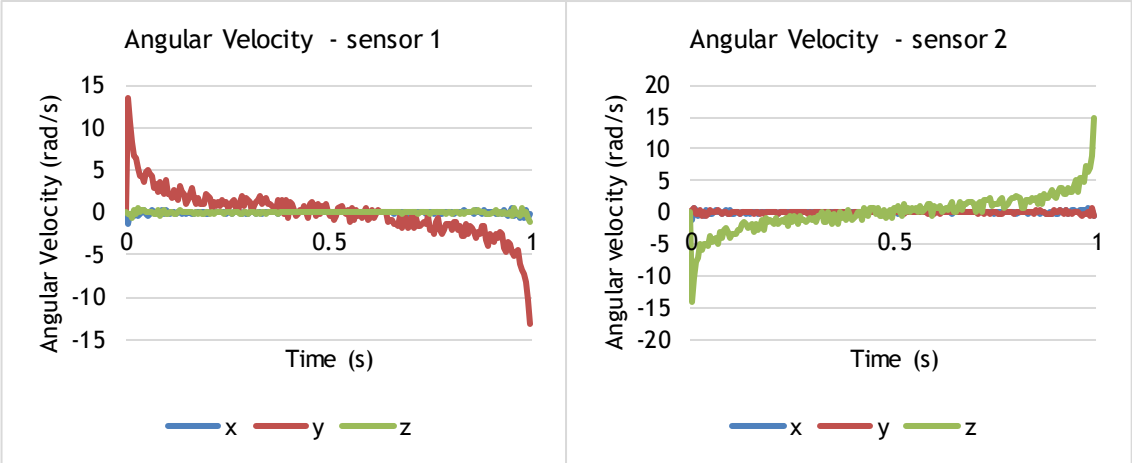

Figure A.23 - Input signal for the simulated data validation. $\phi_1 = 0°, \phi_1 = 90°$, noise 10%

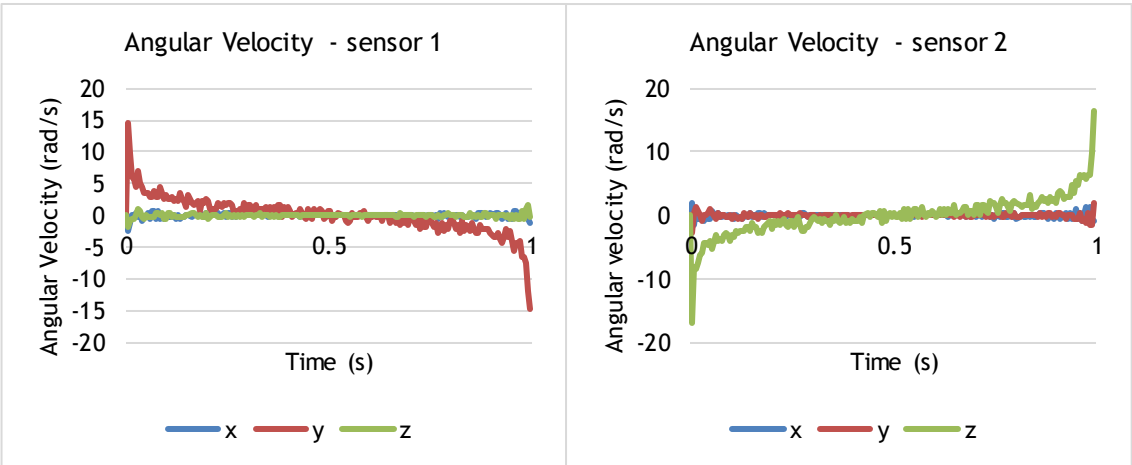

Figure A.24 - Input signal for the simulated data validation. $\phi_1 = 0°, \phi_1 = 90°$, noise 20%

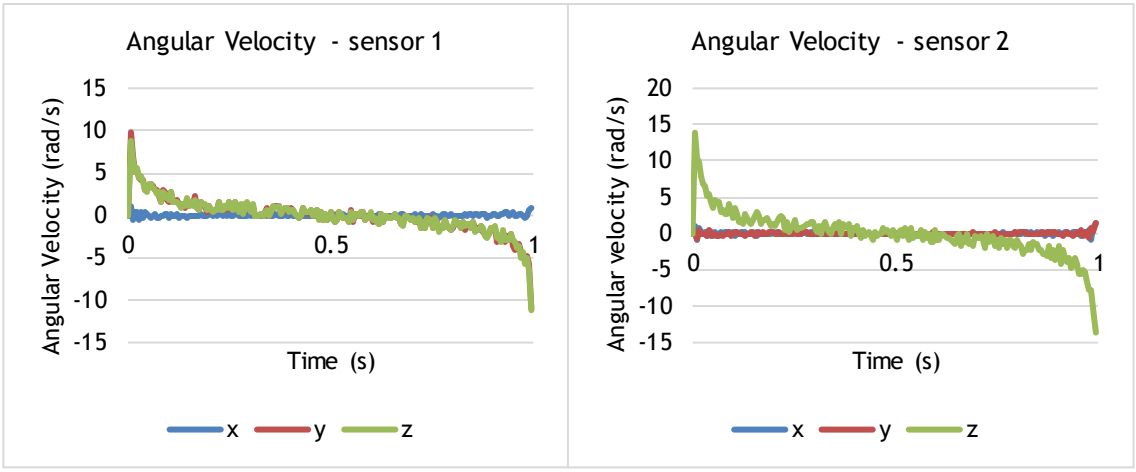

Figure A.25 - Input signal for the simulated data validation. $\phi_1 = 45°$,$\phi_1 = -90°$, noise 10%

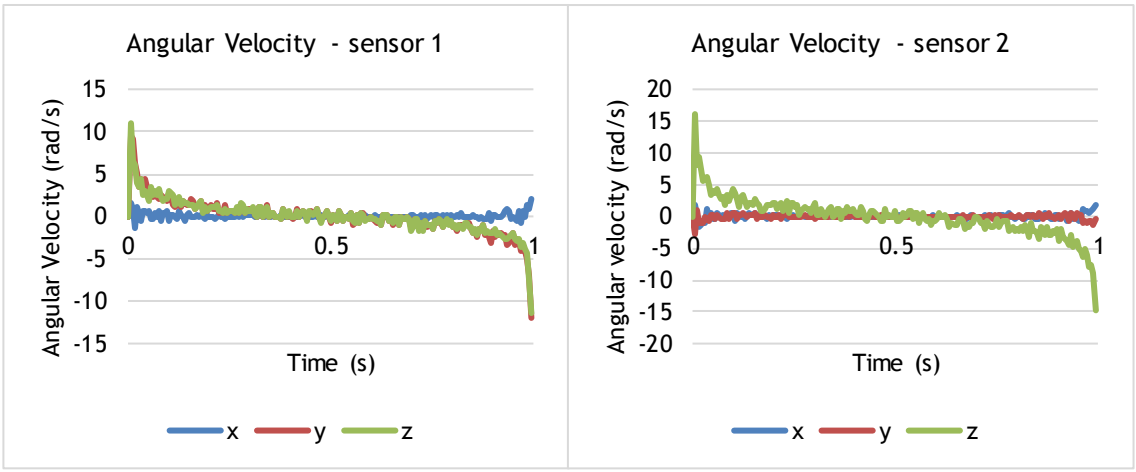

Figure A.26 - Input signal for the simulated data validation. $\phi_1 = 45°$,$\phi_1 = -90°$, noise 20%

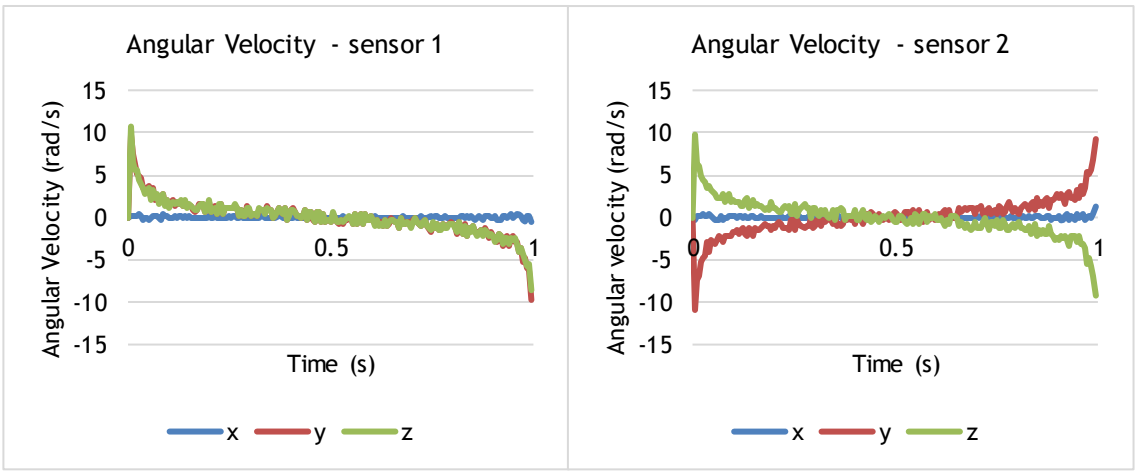

Figure A.27 - Input signal for the simulated data validation. $\phi_1 = 45°$,$\phi_1 = -45°$, noise 10%

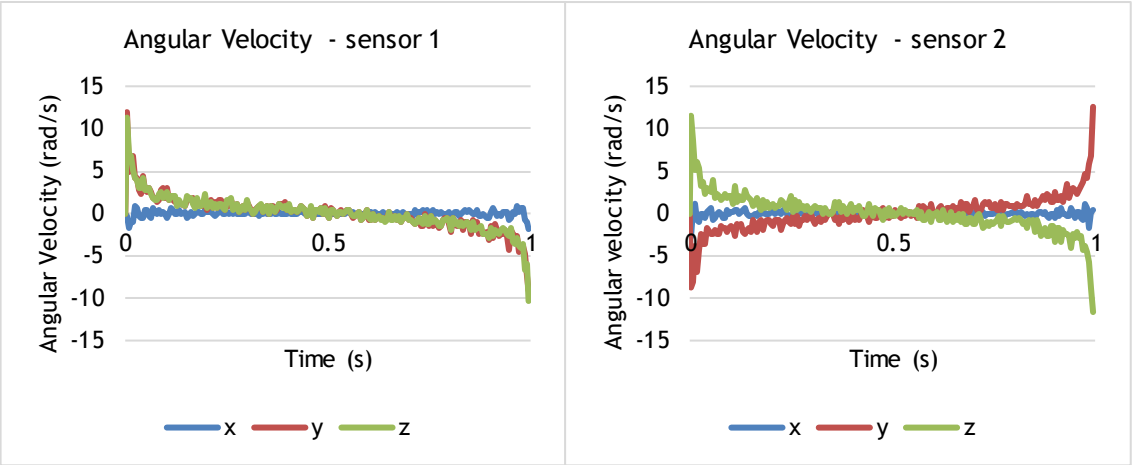

Figure A.28 - Input signal for the simulated data validation. $\phi_1 = 45°$,$\phi_1 = -45°$, noise 20%

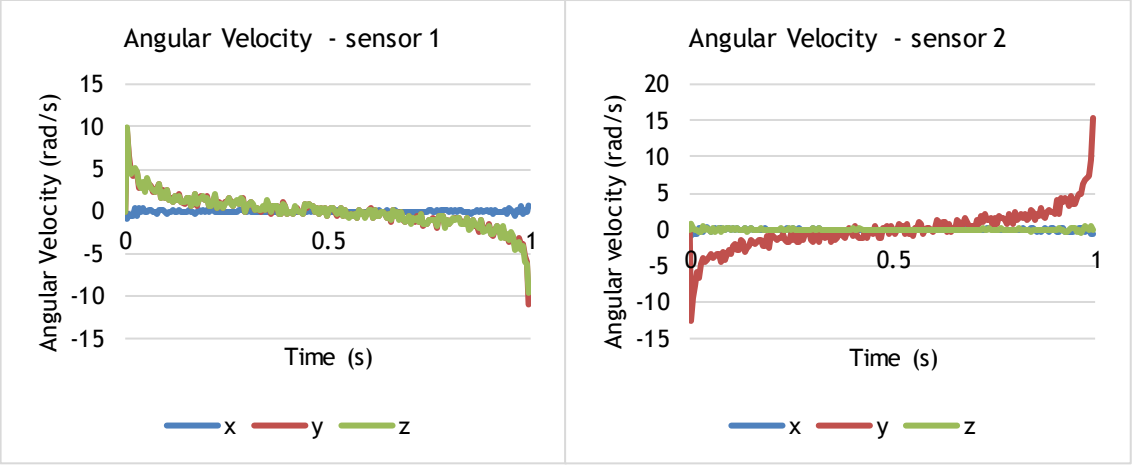

Figure A.29 - Input signal for the simulated data validation. $\phi_1 = 45°$,$\phi_1 = 0°$, noise 10%

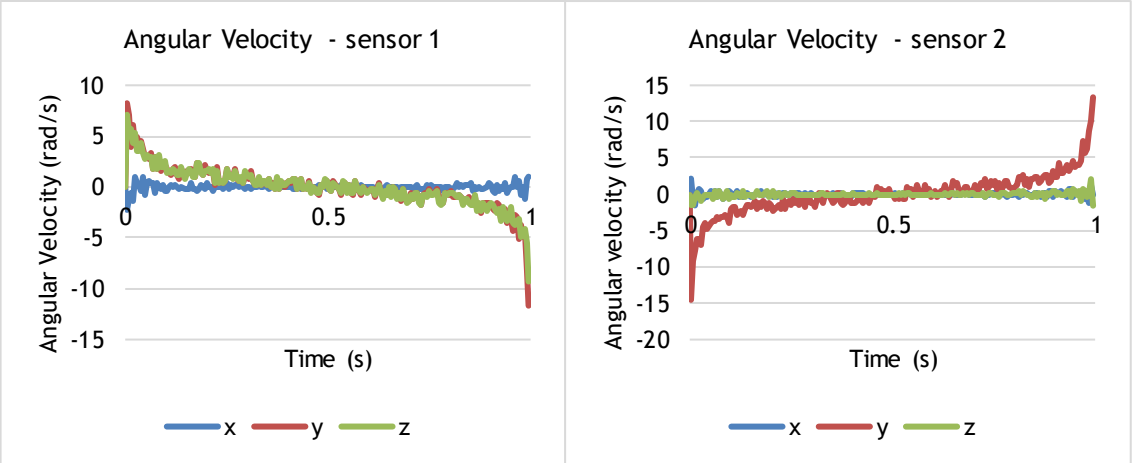

Figure A.30 - Input signal for the simulated data validation. $\phi_1 = 45°$,$\phi_1 = 0°$, noise 20%

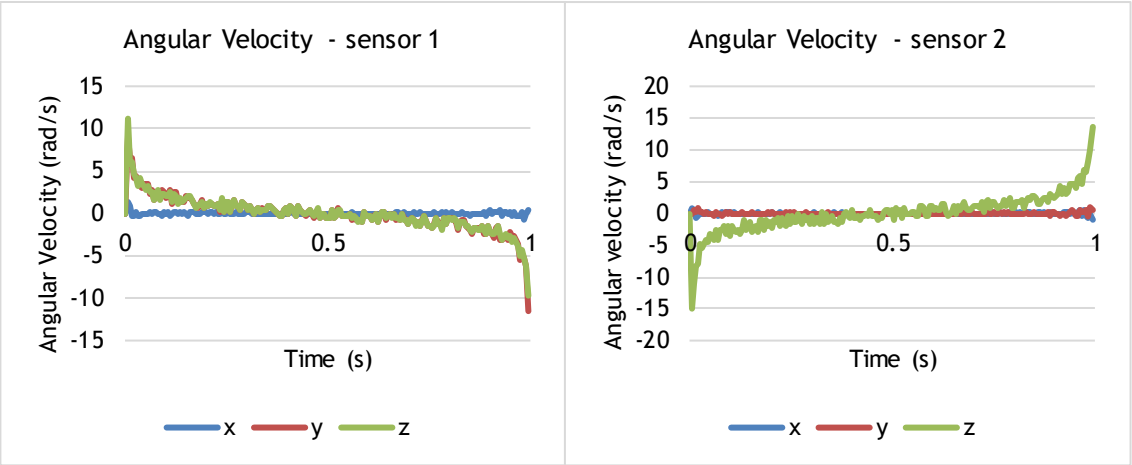

Figure A.31 - Input signal for the simulated data validation. $\phi_1 = 45°$,$\phi_1 = 90°$, noise 10%

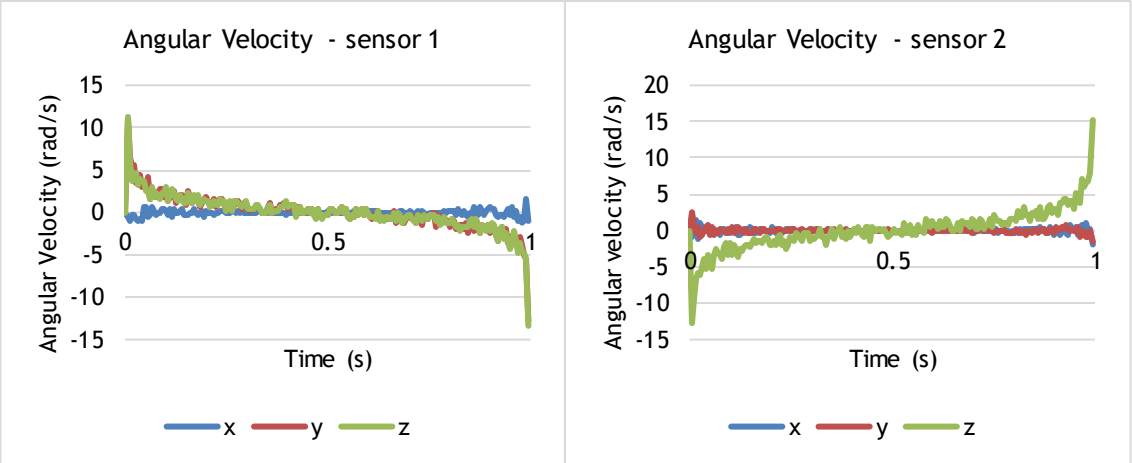

Figure A.32 - Input signal for the simulated data validation. $\phi_1 = 45°$,$\phi_1 = 90°$, noise 20%

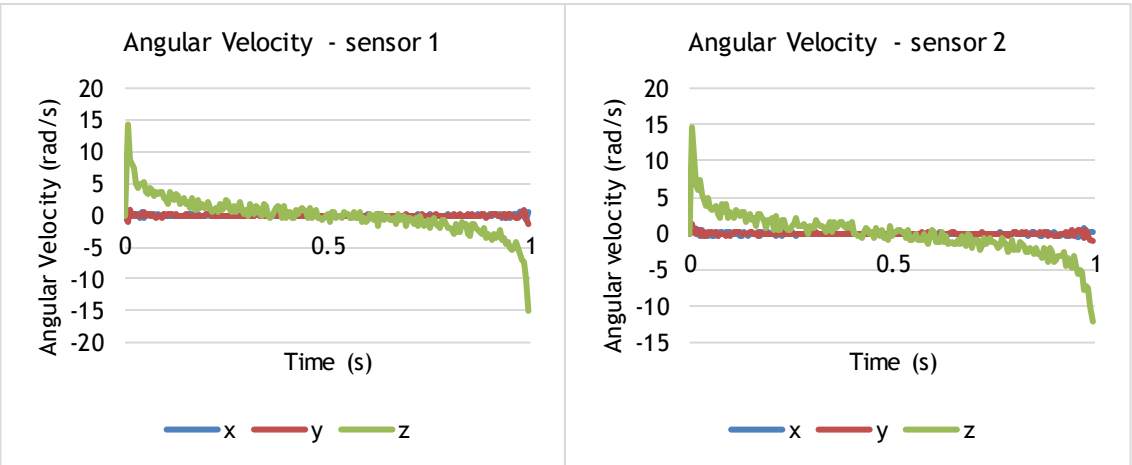

Figure A.33 - Input signal for the simulated data validation. $\phi_1 = 90°$,$\phi_1 = -90°$, noise 10%

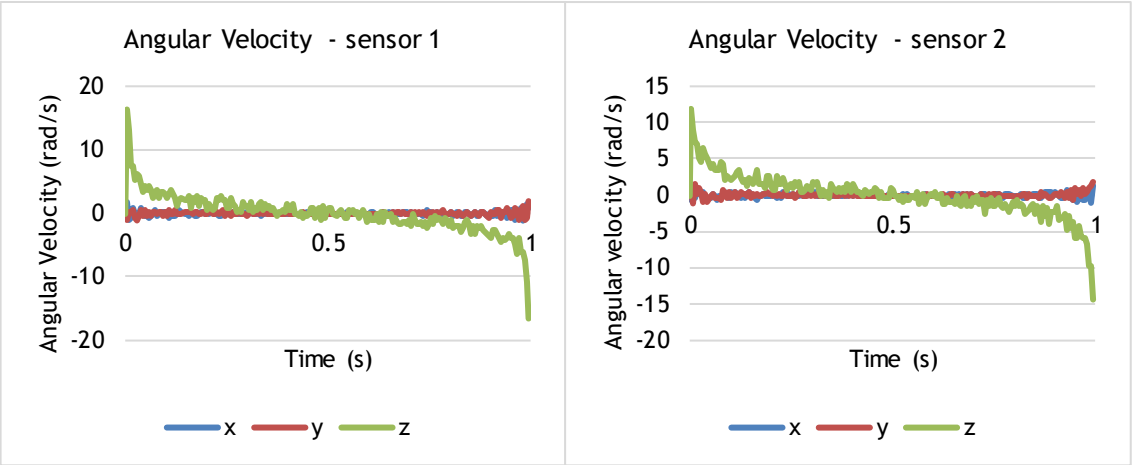

Figure A.34 - Input signal for the simulated data validation. $\phi_1 = 90°$,$\phi_1 = -90°$, noise 20%

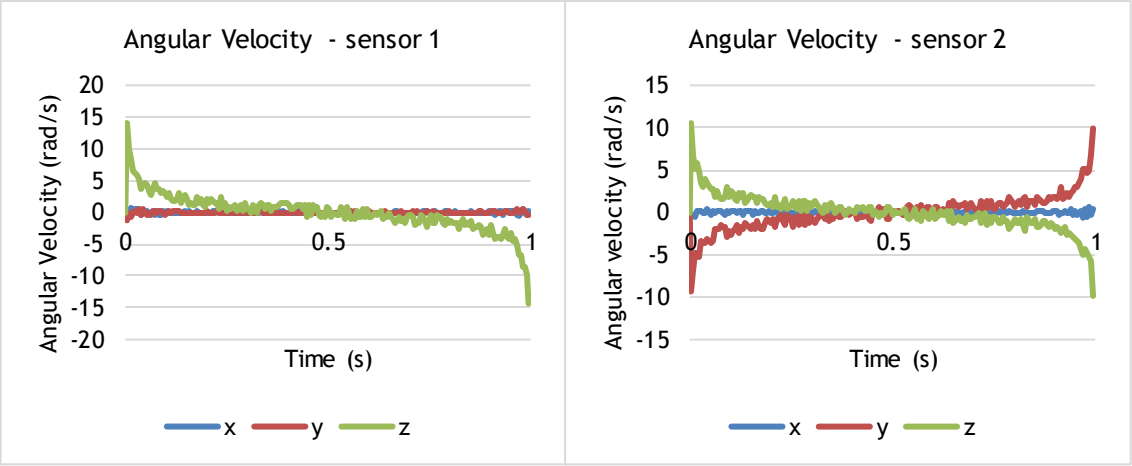

Figure A.35 - Input signal for the simulated data validation. $\phi_1 = 90°$,$\phi_1 = -45°$, noise 10%

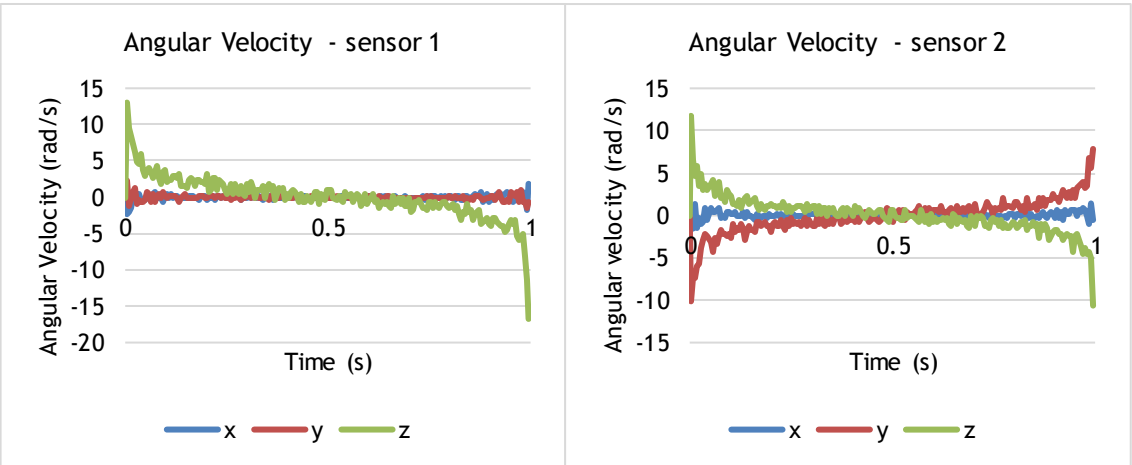

Figure A.36 - Input signal for the simulated data validation. $\phi_1 = 90°$,$\phi_1 = -45°$, noise 20%

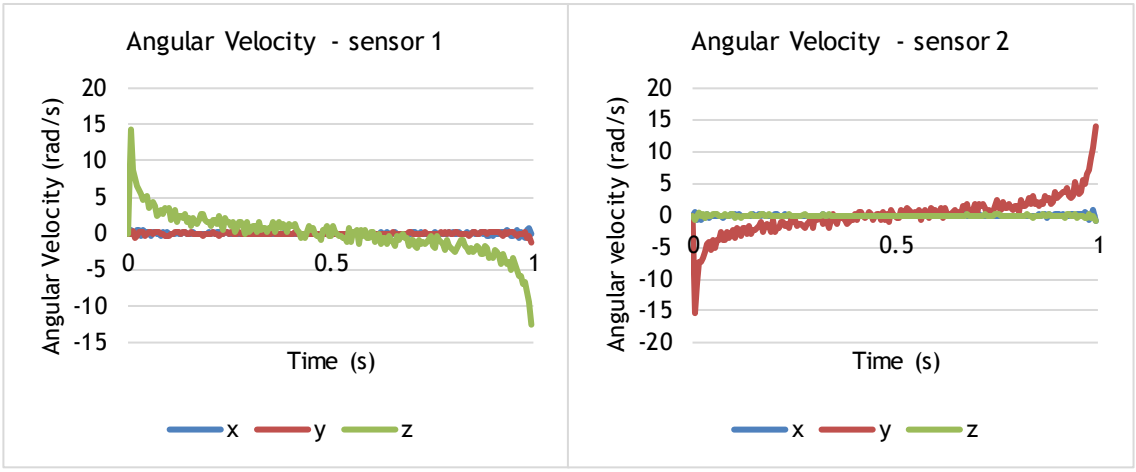

Figure A.37 - Input signal for the simulated data validation. $\phi_1 = 90°$,$\phi_1 = 0°$, noise 10%

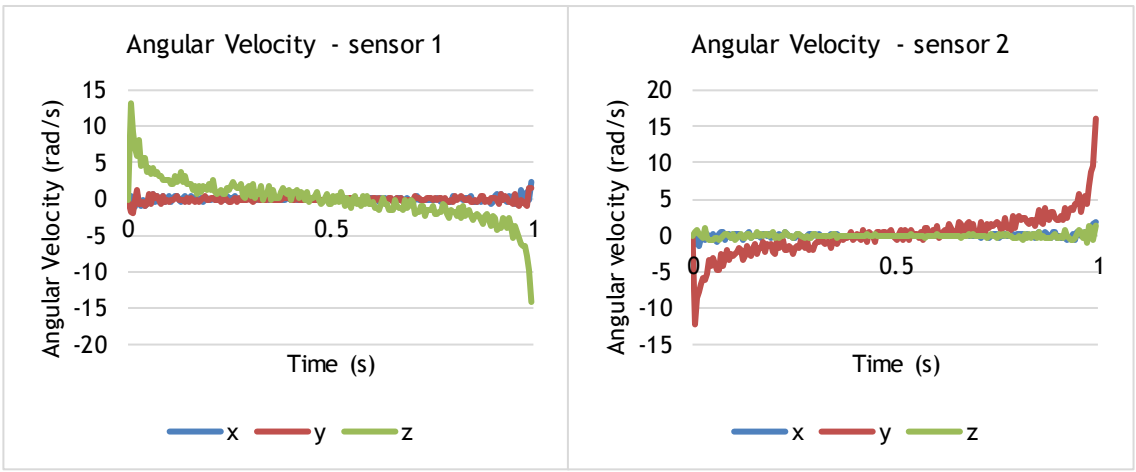

Figure A.38 - Input signal for the simulated data validation. $\phi_1 = 90°$,$\phi_1 = 0°$, noise 20%

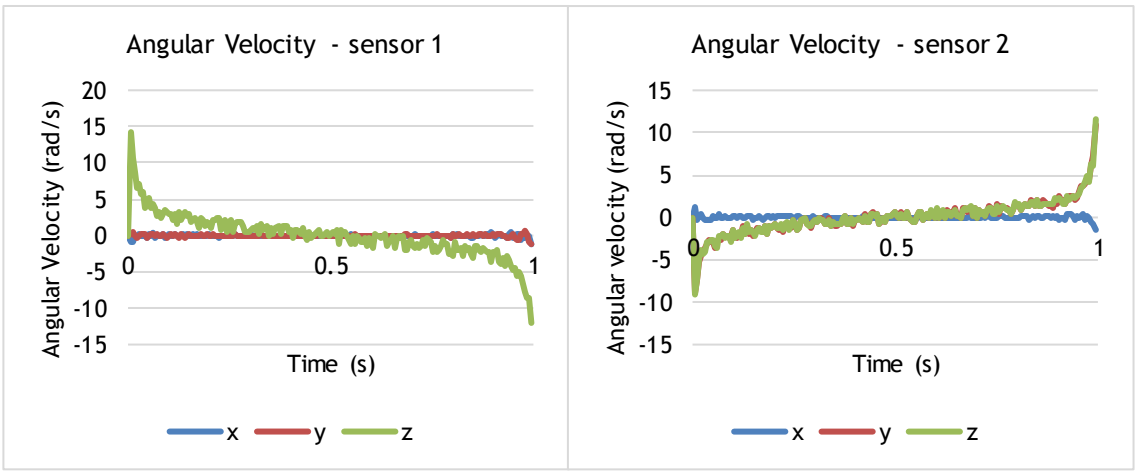

Figure A.39 - Input signal for the simulated data validation. $\phi_1 = 90°$,$\phi_1 = 45°$, noise 10%

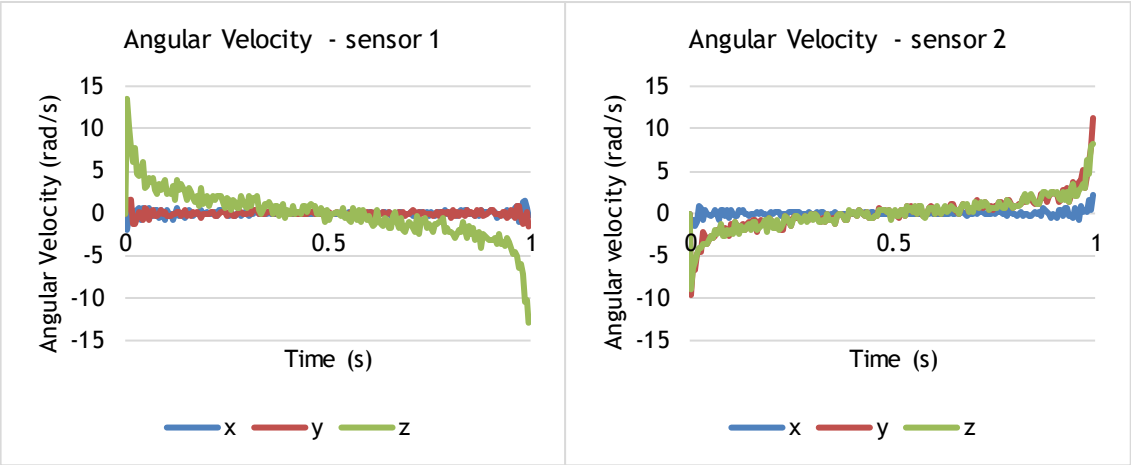

Figure A.40 - Input signal for the simulated data validation. $\phi_1 = 90°$,$\phi_1 = 45°$, noise 20%