UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXVI Ciclo

Inertial Sensing for Human Motion Analysis: Processing, Technologies, and Applications

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Gianluigi Ferrari

Dottorando: *Matteo Giuberti*

Gennaio 2014

alla tata,
per lei avrò sempre 20 anni
ma è stata la prima a dire che sarei diventato un ingegnere..

Table of Contents

List of Figures

List of Tables

List of Acronyms

ADL	Activities of Daily Living
BAN	Body Area Network
BSN	Body Sensor Network
fps	frames per second
GPS	Global Positioning System
***k*-NN**	*k*-Nearest Neighbors
LA	Leg Agility
LDA	Linear Discriminant Analysis
LLA	Left Leg Agility
MDS	Movement Disorder Society
MEMS	Micro ElectroMechanical Systems
NCC	Nearest Centroid Classifier
PCA	Principal Component Analysis
PD	Parkinson's Disease
PE	Position Error
PMF	Probability Mass Function
QDA	Quadratic Discriminant Analysis
RLA	Right Leg Agility
RSS	Received radio Signal Strength
NN	Nearest Neighbor
NNS	Nearest Neighbors Search
UPDRS	Unified Parkinson's Disease Rating Scale

UWB	Ultra-WideBand
WSN	Wireless Sensor Network

Introduction

Nothing happens until something moves.

– Albert Einstein

Human motion has always attracted significant interest and curiosity. In particular, the last two centuries have seen a fast and great development of innovative techniques and technologies for the scientific analysis of human motion. If initially this was mainly due to the large interest in biomedical fields, a growing number of other leading applications has kept this interest alive until today. These applications emerge, for instance, in sport, entertainment, and industrial contexts.

The first motion capture systems, appeared along the nineteenth century, were typically based on optical technologies and their development was profoundly interlaced with the contemporary development of photography and cinematography. Since then, many other different technologies have been employed to develop new motion capture systems, such as (but not limited to) inertial, mechanical, magnetic, and acoustic. In particular, inertial motion capture systems, based on the use of inertial sensors (such as the accelerometer, which measures the acceleration, and the gyroscope, which measures angular velocity), are likely to replace the previous ones and become a standard technology. This is mainly favored by the recent great improvement in the large-scale development of accurate inertial sensors ever cheaper.

When referring to inertial human motion analysis, several application areas are driving current research and development efforts. A tentative list may include, for instance, the following: clinical and home monitoring and/or rehabilitation; ambient

assisted living; computer graphics and computer animation; gaming and virtual reality; sport training; pedestrian navigation; and robotics. Furthermore, human motion analysis often implies a transversal investigation of many aspects of human motion, at different levels of abstraction and at different detail depths. For instance, one may just be interested in recognizing and estimating the pose of a person as well as in identifying the activities and/or the gestures that he/she is performing. Furthermore, one may be just interested in analyzing a restricted part of the body rather than focusing on the full body.

Due to this heterogeneity of topics and intents, this thesis does not focus on a specific application or method, but aims at investigating different aspects of inertial human motion analysis, by specifically discussing the corresponding data processing approaches and the involved technologies. Four research areas have been taken into account which correspond to four types of applications: *arm posture recognition*; *activity classification*; *evaluation of functional motor tasks*; and *motion reconstruction*. In particular, these applications have been chosen in order to cover topics with different levels of abstraction and different detail depths.

Chapter 1

Inertial Sensing for Human Motion Analysis

There is more wisdom in your body than in your deepest philosophy.

– Friedrich Nietzsche (Thus Spoke Zarathustra)

1.1 Inertial Sensing

Inertial sensing involves the measurement of inertial properties of a body through sensors. To this end, in the following, two inertial sensors, i.e., the accelerometer and the gyroscope, and their principles of operation are briefly described.

1.1.1 Accelerometer

An accelerometer is an inertial sensor able to measure and quantify acceleration. More specifically, an accelerometer produces an (electrical) output proportional to the rate of the acceleration. A proper calibration is then expedient to convert this output into an actual acceleration measurement.

In physics, acceleration can be defined as the rate at which a body velocity

changes with time. Since velocity, as well as acceleration, is a vector quantity and is, thus, characterized by a magnitude and a direction, a change in acceleration will happen if either the magnitude or the direction of the velocity changes. Note, however, that acceleration and velocity are two distinct concepts. Indeed, an object can move very fast (and, thus, can have a high velocity) and still not be accelerating.

Furthermore, the velocity of an object will only change if a net force is acting on it. A change in the forces acting on an object will thus lead to a non-zero acceleration. More specifically, according to Newton's Second Law, the magnitude of the acceleration a (dimension: [m/s^2]) is proportional to the object mass m (dimension: [kg]) and the net force F (dimension: [N]) which it is experiencing as follows:

$$a = \frac{F}{m}. \tag{1.1}$$

There exist fundamentally two types of forces acting on a body: *static forces*, such as the constant force of gravity, and *dynamic forces*, caused by external stimulation. An accelerometer can be used to measure these forces (and, thus, the corresponding accelerations). In fact, an accelerometer measures any force, such as shocks, vibrations, rotations, and tilting, that causes acceleration.

As specified earlier, according to SI, acceleration should be measured in m/s^2. However, it can be sometimes expressed in g units (where "g" stands for "gravity"), where 1 g is equivalent to 9.81 m/s^2, i.e., the acceleration of a body in free fall.

The simplest way of measuring acceleration is by using a spring mass system. This system is based on two main principles: the Second Law of Newton (shown in Equation (1.1)) and the Hooke's law. To this end, let us assume to have a mass which is connected via a relaxed spring to the object whose acceleration has to be measured. If the object is accelerated to the left, the mass will experience a force which will cause it to move to the right. According to the Hooke's law, this force can be defined as follows:

$$F = k\Delta x \tag{1.2}$$

where k (dimension: [N/m]) is a positive real constant value (which depends on the specific spring behavior) and Δx (dimension: [m]) is the mass displacement. Under constant acceleration, Equations (1.1) and (1.2) lead to the following equality of

forces:

$$k\Delta x = ma \tag{1.3}$$

and, therefore,

$$a = \frac{k}{m}\Delta x \tag{1.4}$$

which allows to measure the object acceleration just by quantifying the mass displacement.

Many different types of accelerometers are commercially available, each of them relying on different operation principles. Among them, the following accelerometer technologies are valuable to be cited.

- *Piezoelectric*. These accelerometers contain microscopic quartz or ceramic crystal structures that, when stressed by forces, cause a voltage to be generated. This voltage is then used to produce an electrical output proportional to the acceleration.
- *Micro ElectroMechanical Systems (MEMS)*. These accelerometers have a mechanical system which moves in response to an acceleration. This movement causes a change in an electrical characteristic such as a capacitance. This change is further amplified to produce a voltage proportional to the acceleration. The size of these structures is typically of the order of micrometers.
- *Heat Gas*. In such accelerometers, hot gas is distributed in a chamber also equipped with temperature sensors. When the device is rotated, the sensors will experience different temperatures based on their positions. In this way, the static tilt of the device can be measured.

A detailed review of MEMS accelerometers is further provided in [9].

1.1.2 Gyroscope

A gyroscope is an inertial sensor able to directly measure and quantify angular motion of an object with respect to an inertial frame of reference. More specifically, depending on whether the angular velocity or orientation is being measured, they

can be divided into *rate gyroscopes* and *angle gyroscopes*, respectively. The angular velocity can be expressed in terms of either $^{o}/s$ or rad/s.

Gyroscopes are based on the principle of the conservation of angular momentum. The angular momentum of an object does not change unless it is acted upon by a torque. The fundamental equation describing the behavior of the gyroscope is the following:

$$\boldsymbol{\tau} = \frac{d\boldsymbol{L}}{dt} = \frac{d(I\boldsymbol{\omega})}{dt} = I\boldsymbol{\alpha} \tag{1.5}$$

where: $\boldsymbol{\tau}$ (dimension: [N·m]) is the torque on the gyroscope; $\boldsymbol{L}$ (dimension: [N·m·s]) is its angular momentum; I (dimension: [kg·m^2]) is its moment of inertia; and $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ (dimension: [rad/s^2]) are, respectively, its angular velocity and angular acceleration.

Mechanically, a gyroscope is a spinning wheel or disc in which the axle is free to assume any orientation. The first mechanical gyroscope was built, in 1852, by Foucault, as a gimbaled wheel that stayed fixed in space due to angular momentum while the platform rotated around it. However, due to their large size and cost, such mechanical gyroscopes are not suitable for human motion analysis. Instead, *vibrating structure gyroscopes* (which belong to the family of MEMS sensors) are small and inexpensive sensors with low power requirements able to measure the angular rate of an object by exploiting the Coriolis effect. Indeed, according to the Coriolis effect, a vibrating element, when rotated, experiences a secondary vibration (which is an apparent force, i.e., the Coriolis force), which is orthogonal to the original vibrating direction. The Coriolis force $\boldsymbol{F}_{\mathrm{C}}$ (dimension: [N]), associated to the latter vibration, is defined as follows:

$$\boldsymbol{F}_{\mathrm{C}} = -2m(\boldsymbol{\omega} \times \boldsymbol{v}) \tag{1.6}$$

where: m is the mass of the vibrating element; $\boldsymbol{v}$ (dimension: [m/s]) is its velocity; and $\boldsymbol{\omega}$ is the angular velocity applied to the object. Note that the Coriolis force is a result of Newton's laws of motion (which describe the motion of an object in a non-accelerating inertial frame of reference) applied to rotating frames.

Nowadays, gyroscopes are developed exploiting different technologies. A tentative list may include:

- Gyrostat;

- MEMS gyroscope;
- Ring Laser Gyroscope (RLG);
- Fiber Optic Gyroscope (FOG);
- Vibrating Structure Gyroscope (VSG) or Coriolis Vibratory Gyroscope (CVG);
- Dynamically Tuned Gyroscope (DTG);
- gyroscope with MHD-sensors (based on the MagnetoHydroDynamics effect);
- electro-chemical gyroscope (based on Molecular Electronic Transducer, MET, technology);
- London moment gyroscope.

Concerning MEMS gyroscopes, a detailed review is further provided in [9].

1.2 Human Motion Analysis

1.2.1 Historical Background

The interest in motion (and, more specifically, the birth of the modern biomechanics) has very ancient roots [10, 11]. History of science usually begins with the ancient Greeks, who first left a record of human inquiry concerning the nature of the world in relationship to our powers of perception. In particular, Aristotle (384-322 BCE), who

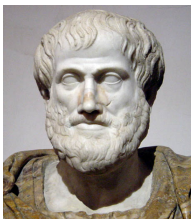
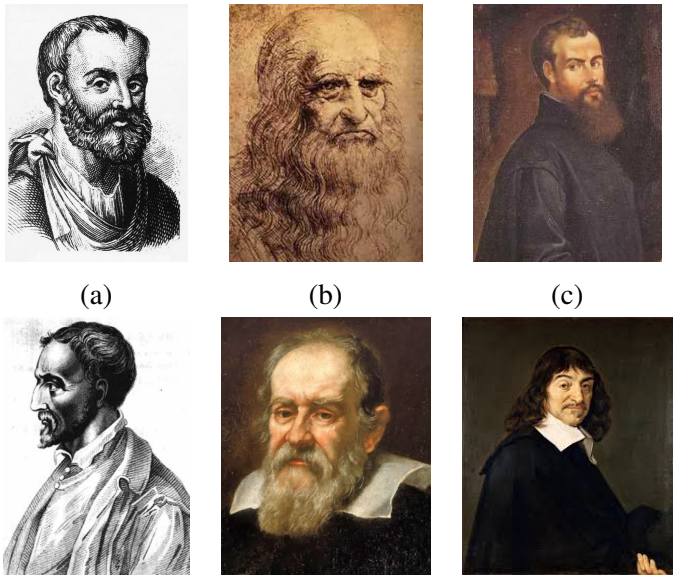


Figure 1.1: Aristotle.

was a student at the Plato's academy and father of the deductive reasoning, might be considered the first biomechanician. Its book called "De Motu Animalium" [12] (translated: "On the Movements of Animals"), where animal bodies are seen as mechanical systems and the difference between imagining performing and action and actually doing it is questioned, is indeed considered one of the earliest writings on biomechanics.

Although other scientists (among which we can cite Galen, Leonardo da Vinci, Andreas van Wesel, Girolamo Cardano, Galileo Galilei, and René Descartes) have



(a) (b) (c)

(d) (e) (f)

Figure 1.2: (a) Galen, (b) Leonardo da Vinci, (c) Andreas van Weasel, (d) Girolamo Cardano, (e) Galileo Galilei, and (f) René Descartes.

differently contributed to the evolution of this discipline, Giovanni Alfonso Borelli (1608-1679) is still considered the actual father of modern biomechanics. Borelli, who at age 16 went to Rome to become a student of Benedetto Castelli (Galileo's former student), also knew Galileo himself and was surely fashioned by the scientific method, which Galileo used to follow in all of his studies. Borelli, at age 50, became

teacher of Mathematics at Pisa, where Galileo previously taught as a young man and there he worked closely with Marcello Malpighi, the much younger chair of theoret-

(a) (b) (c)

Figure 1.3: (a) Giovanni Alfonso Borelli, (b) Benedetto Castelli, and (c) Marcello Malpighi.

ical medicine. Favored by this collaboration, Borelli was the first who performed the first experiment in gait analysis and who understood the musculoskeletal system as a set of levers that magnified motion rather than force (many years before Newton actually produced his laws of motion). His great treatise, called (after that of Aristotle) "De Motu Animalium," [13] was published posthumously in 1680.

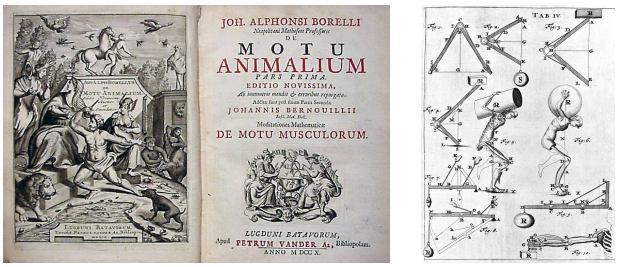
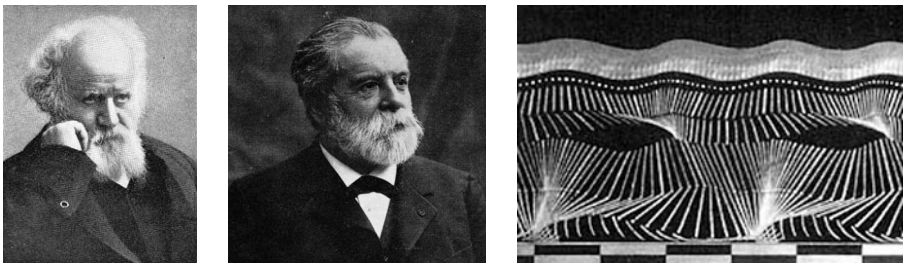


Figure 1.4: Some extracts from Borelli's treatise "De Motu Animalium."

After Borelli, there is a little sign of biomechanics in the literature until the latter half of the 19th century, where the first techniques of motion capture were developed. In particular, the idea of investigating locomotion using cinematography and photography, suggested by the French astronomer Pierre Janssen, was taken and scientifically readapted by the French Étienne-Jules Marey (1830-1904), who first correlated

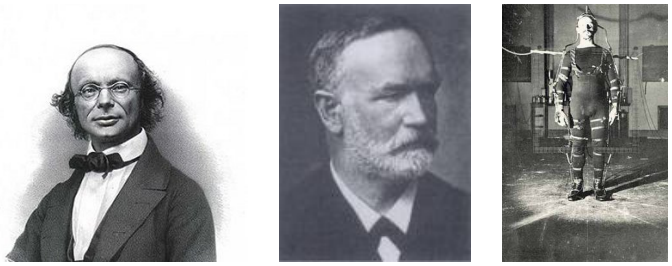
ground reaction forces with movement. His works about chronophotography, made in collaboration with his students Carlet and Demeny, allowed to accurately record the progression of the human limbs while walking and pioneered the modern mo-



(a) (b) (c)

Figure 1.5: (a) Pierre Janssen, (b) Étienne-Jules Marey, and (c) Joinville soldier walking (Marey, 1883).

tion analysis. In the meantime, in Germany, the Weber brothers were hypothesizing a great deal about human gait [14], but it was Christian Wilhelm Braune (1831-1892) and his student Otto Fischer (1861-1917), while studying three-dimensional gait analysis and inertial parameters of human body, who significantly advanced the science using recent advances in engineering mechanics [15].



(a) (b) (c)

Figure 1.6: (a) Wilhelm Eduard Weber, (b) Christian Wilhelm Braune, and (c) Braune and Fischer's subject wearing their experimental suit.

Finally, in the 20^{th} century, the number of scientists who contributed to expand

and improve the knowledge of human kinematics and kinetics was increasingly larger.

1.2.2 Motion Capture Technologies

Inspired by the first motion capture systems presented in Subsection 1.2.1, many other different technologies have been so far employed to develop new motion capture systems. A (possibly incomplete) list of motion capture technologies may include the following:

- Optical;
- Inertial;
- Mechanical;
- Magnetic;
- Acoustic;
- Hybrid.

In the following, a description of these technologies is given.

Optical

Probably the one with the oldest roots (as anticipated in Subsection 1.2.1), optical motion capture systems are based on the use of cameras to record and infer the user motion. Optical systems can be mainly distinguished into two main categories: marker-based (or optoelectronic) and markerless. Optoelectronic systems, which determine the positions of markers on the basis of measurements of time-of-flight infrared light intensity and triangulation, can use active or passive markers. In order to guarantee distinction from each other when tracked by multiple cameras, active markers (usually infrared leds) are excited at unique frequencies, whereas passive markers (usually reflective spheres) rely on different sizing and/or accurate calibration procedures. On the other hand, markerless systems capture the user motion by mainly exploiting

(a) (b)

Figure 1.7: Commercial examples of (a) optoelectronic (namely, Vicon [1]) and (b) markerless (namely, Microsoft Kinect [2]) optical motion capture systems.

computer vision techniques of pattern recognition and often require high computational resources. In some cases, depth cameras and structured light projection are also used to get more information about the sensed environment.

Optoelectronic systems are extremely accurate and, for this reason, they are typically used as golden references for the validation of other motion capture technologies. However, they require specialized laboratories and are complex to be used. On the opposite, markerless systems are less expensive and complex at a price, however, of a lower accuracy. Furthermore, both categories share, as drawbacks, the large sensitivity to markers' occlusion and different light conditions. Finally, the user movements must be limited to the area captured by the cameras.

Two examples of optical motion capture systems (namely, the Vicon system [1] and Microsoft Kinect [2]) are shown in Figure 1.7.

Inertial

After optical systems, inertial systems probably represent the second most popular way of achieving motion capture and they are increasingly attracting interest. Inertial motion capture systems are composed of sensor devices, typically equipped with accelerometers and gyroscopes, placed on every rigid body segment of the user. The angular velocity measured by gyroscopes is integrated to estimate the orientation (with respect to an initial reference orientation). However, gyroscopes suffer from drift and the previous estimate requires to be continuously corrected and compensated by absolute orientation measurements. This can be done by using an accelerometer which, by observing the direction of the measured gravity acceleration, allows to estimate the inclination of the device with respect to the vertical direction. This is however no longer true when the sensor experiences also linear accelerations due to the user motion which may cause inaccuracies in the orientation estimation process. Knowing the orientation of a device, the contribution of the gravity acceleration is removed so that the remaining measured acceleration (now only related to the user motion) can be twice integrated to infer the device position.

One advantage of these systems is that they are self-contained. The major drawback however is that low-cost sensors are typically very sensitive to noise and drift which produce to unavoidable errors over long time periods. Nevertheless, recent advances in sensor fusion algorithms and in the miniaturization process have allowed the development of small and very accurate drift-corrected inertial sensors.

A commercial example of an inertial motion capture system (namely, Xsens MVN [16]), is shown in Figure 1.8.

Mechanical

Mechanical motion capture systems consist of rotational encoders (such as goniometers and potentiometers), which are able to measure relative joint angles, rigidly mounted on a body exoskeleton. The knowledge of the relative joint angles allows to estimate the user body pose through the use of simple biomechanical model-based algorithms.

(a) (b)

Figure 1.8: Commercial example of an inertial motion capture system from Xsens: the (a) wired (MVN [3]) and (b) wireless (MVN Awinda [4]) solutions are shown.

Since these systems do not depend on any external infrastructure, they can be used to track users without working area constraints. Moreover, they do not suffer from shadowing or interference problems. However, they are relatively cumbersome to wear and present serious difficulties in aligning external instrumented joints with the body joints, especially when the motion of joints with multiple degrees of freedom (e.g., the shoulder) needs to be captured. In addition, they are very sensitive to soft tissue artifacts which lead to variations of the instruments position as user motion occurs. Mechanical motion capture systems require an independent calibration for each user.

A commercial example of a mechanical motion capture system (namely, Gypsy from Meta Motion [5]), is shown in Figure 1.9.

Figure 1.9: Gypsy [5]: a commercial example of a mechanical motion capture system from Meta Motion.

Magnetic

Magnetic motion capture systems exploit body-worn trackers to measure magnetic fields generated by a transmitter source, which is composed of three orthogonal coils emitting a magnetic field when a current (either direct or alternating) is applied. Since the strength of the measured fields is proportional to the distance from each coil of the source, on the basis of these measurements, the positions and orientations of the body-worn trackers can be estimated.

Advantages of this technology include the small size of the trackers and the absence of line-of-sight requirements (since the body is mostly transparent to magnetic fields). On the opposite, magnetic field strength decreases rapidly with distance from the generating source and can be easily distorted, especially when the field is produced by alternating current, by electromagnetic interferences caused by ferromagnetic materials in the vicinity.

Figure 1.10: MotionStar [6]: commercial example of a magnetic motion capture system from Ascension Technology Corporation.

A commercial example of a magnetic motion capture system (namely, MotionStar [6] from Ascension Technology Corporation), is shown in Figure 1.10.

Acoustic

Acoustic systems require the user to wear acoustic emitters (i.e., speakers) on his/her body and to distribute acoustic receivers (i.e., microphones) in his/her surroundings (or, without loss of performance, viceversa). Motion capture is then achieved using ultrasonic pulses that allow to determine the body devices' positions through time-of-flight multilateration (or phase-coherence). In case only the relative positions of body segments are needed, acoustic transducers may be just placed on the user body enabling an improved freedom of motion.

Due to the physics of sound waves, a line-of-sight between acoustic devices is always required (otherwise, acoustic occlusions occur) and devices should not be too

distant. Furthermore, these systems are very sensitive to audio interference (caused by reflections of sound waves with the environment) and present a relatively low transmission rate. Finally, since the efficiency of an acoustic transducer is proportional to the active surface area, one has to deal with a trade-off between the size of the devices and the inherent performance of the system.

Hybrid

The previously presented motion capture systems always present both advantages and drawbacks, typically inherited from the component technology. To this end, hybrid motion capture systems rely on the combination of two (or more) of the previous technologies, in order to exploit their advantages while tackling and overcoming the related limitations. For instance, inertial systems, typically characterized by inherent drift in the estimation of the devices' orientation, have been combined, for compensation purposes, with magnetic [17, 18], Global Positioning System (GPS) [19], optical [20], and acoustic [21] systems.

1.2.3 Classification of Human Motion Analysis

The analysis of human motion typically involves a multi-level investigation of different aspects of human motion. For instance, we can imagine to map it onto a two-dimensional plane where the two dimensions represent its *level of abstraction* and its *detail depth*. For ease of clarity, a tentative classification of human motion analysis (with a specific focus on the use of inertial technologies) at varying abstraction level and detail depth is represented in Table 1.1. According to Table 1.1, the increase of the abstraction level corresponds to move the focus of the analysis from basic *inertial parameters* (low abstraction level) toward human *actions* (high abstraction level), passing through the concept of *pose* (medium abstraction level). Furthermore, each of these level may be investigated with varying depth of detail (from low to high).

In particular, at the highest level of abstraction, we are interested in analyzing the actions of a person, i.e., if the user is doing something and what he/she is doing. In this case, the following areas of analysis can be distinguished.

Table 1.1: Classification of human motion analysis (focusing on inertial technologies).

		Detail Depth		
		Low	**Medium**	**High**
Abstraction Level	**High (Action)**	Activity Detection	Activity Recognition	Gesture Recognition
	Medium (Pose)	(Partial/Full) Pose Classification	Partial Pose Estimation	Full Pose Estimation
	Low (Inertial Parameters)	Basic Signal Analysis	Partial Signal Analysis	Complete Signal Analysis

- *Activity Detection* (low detail depth). The aim is to know whether a person is actually performing some activities or not. It can be used to segment a sequence of unlabeled inertial data in order to perform further investigations about the types of the performed activities and/or simply to measure the activity level of that person.

- *Activity Recognition* (medium detail depth). It implies to recognize activity primitives (such as modes of locomotion, i.e., sit, stand, lie, and walk) from sequences of inertial data. In particular, the time occurrence, duration, and type of these activities need to be estimated.

- *Gesture Recognition* (high detail depth). It is of interest to recognize specific gestures performed by the user (e.g., open/close a drawer, open/close a door, drink a cup of coffee, and read a newspaper) from sequences of inertial data. As before, the time occurrence, duration, and type of these gestures need to be estimated.

Moving down to a medium level of abstraction, the focus is about the person pose, i.e., the orientation and position information about (part of) his/her body segments. According to the required level of detail, the following areas of analysis can be distinguished.

- *(Partial/Full) Pose Classification* (low detail depth). The aim is to classify the (partial or full) pose of a person, by choosing among a predefined set of known poses. Since the choice is limited to the considered set of poses, the pose estimate is rough and cannot be used in applications where an accurate estimate of the pose is required.

- *Partial Pose Estimation* (medium detail depth). It requires the accurate estimation of part of the person pose, i.e., the orientations and/or positions of part of his/her body segments. This is typically employed in applications where the full body pose is not necessary and the focus is, for instance, on a body limb (e.g., an arm or a leg).

- *Full Pose Estimation* (high detail depth). It involves the accurate estimation of the full body pose of a person, in terms of orientations and positions of his/her body segments. It is typically required in scenarios where a complete virtual reconstruction of a human body is necessary (e.g., for computer animation).

Finally, at the lowest level of abstraction, a characterization of inertial (spatio-temporal) parameters of the person motion is investigated. This means that the person pose or activity is not necessarily of interest. Instead, the analysis of raw inertial signals from all or part of sensor devices is mostly sufficient. In this case, the following areas of analysis can be distinguished.

- *Basic Signal Analysis* (low detail depth). The signal analysis is here limited to the estimation of basic inertial parameters, mainly related to the motion of a specific body segment.

- *Partial Signal Analysis* (medium detail depth). It implies a partial characterization of the person motion in order to extract some significant spatio-temporal parameters.

- *Complete Signal Analysis* (high detail depth). Advanced signal processing techniques are here employed to extract a complete and accurate time-frequency characterization of the person motion.

1.2.4 Areas of Application

Generally, the analysis of human motion is increasingly attracting interest in many application areas and contexts such as biomedical, sport, entertainment, and industrial. More specifically, focusing on inertial sensing of human motion, the following areas of application mainly motivates the current research and development efforts.

- *Clinical and home monitoring and/or rehabilitation*. Many pathologies require the patient conditions to be continuously monitored and evaluated. Due to the lack of specialized personnel, to the high cost involved, and to the large number of patients, this is not typically possible in traditional healthcare systems, where such evaluations are instead scheduled with lower frequency, leading sometimes to situations where a successful intervention is no longer possible. To this end, there is a rapid shift from a clinical-based setting to a patient or home centered setting with the help of sensor technologies. In addition, since functional motor tasks are often expedient to get information about the health status of a patient (e.g., gait analysis), the use of inertial sensors allows to continuously and remotely keep track of the patient conditions. This is especially useful during post-injury rehabilitation periods or during monitoring of motion impaired (elderly) people, when a home therapy contributes to improve both the recovery and the quality of life of the patient.

- *Ambient assisted living*. People who live alone, especially elderly or impaired people, need a continuous supervision so that unusual and/or critical events may be detected and possibly prevented. Smart environments, where unobtrusive sensors are distributed in the living area, may already allow to get high-level information about the monitored person behavior. However, inertial sensors enable a more accurate tracking of the person and of his/her activities.

- *Computer graphics and computer animation*. In traditional animation, each frame was carefully hand drawn. Several months were then required to complete even short animated movies. The advent of computers has automated and, consequently, accelerated this process, but it is the advent of motion capture techniques which definitely reduced the computational efforts in the production of animated sequences. Motion sequences performed by professional actors are indeed used to quickly get motion data which are retargeted to character models of varying sizes. This is particularly useful in the creation of animated features, special effects for movies, and video games.

- *Gaming and virtual reality*. One of the aims of gaming is providing the gamers with a captivating and interactive experience. To fulfill this objective, new generation video games consoles are already developing advanced control systems where the user can interact with the game by using his/her body or part of it (e.g., Nintendo Wii and Microsoft Kinect). To this end, inertial sensors are very suitable tools that can be used to capture the motion of the user and project it onto a virtual character. Virtual reality applications, especially used for rehabilitation purposes where the interaction of a patient with a virtual environment is investigated, are other motivating applications.

- *Sport training*. In a world where athletes are reaching the limits of the inherent performance of the human body system, inertial systems are valuable tools that can be used to improve athletes performance in cases where the standard coaching system may be slow and inaccurate. Using inertial sensors, in a fast and automatic way, the athlete can easily spot mistakes in the execution of sportive gestures and correct them accordingly.

- *Pedestrian navigation*. Due to frequent GPS outages, pedestrian (especially indoor) navigation, especially used for routing people in fairs and exhibitions, urban contexts, and/or industrial environments, may be critical in such contexts. Inertial sensors can thus be exploited to track the people and provide them with updated location information.

- *Robotics*. In this area, many applications have been developed which, through the use of inertial sensors, allow to control a mechanical prosthesis or a robotic arm. These types of applications are particularly of interest in dangerous industrial environments where the safety and the health of a person is at risk if directly present.

1.3 Thesis Outline

As observed in Section 1.2, the analysis of human motion broadly covers (even very) different fields of research and application. Due to this heterogeneity of topics, a plethora of technologies and approaches have been so far explored in order to address the specific requirements of the considered applications.

Therefore, this thesis does not focus on a specific application, but aims at investigating different aspects of human motion analysis, by specifically discussing the corresponding data processing approaches and the involved technologies. In particular, four research areas have been taken into account which correspond to four types of applications. These applications have been chosen so that topics with different levels of abstraction and different detail depths (according to Table 1.1) are covered.

Specifically, this thesis is organized as follows.

- **Chapter 2:** ***Arm Posture Recognition***. This chapter investigates a hybrid radio/inertial approach to arm posture recognition. A radio fingerprinting-based approach, through measurements of the Received radio Signal Strengths (RSSs) from anchor nodes, is first used to localize (i.e., determine the positions of) target nodes properly placed on a user arm. Accelerometric signals generated by the target nodes are then used to estimate the pitch of every device in order to refine the radio fingerprinting estimates. The system performance is experimentally evaluated taking into account different fingerprinting-based localization algorithms (namely, deterministic and probabilistic) and a comparison between the performance of the proposed system and that of a low-cost optical arm posture recognition system (namely, Kinect) is presented.

According to Table 1.1, this topic belongs to "Partial Pose Estimation", with *medium* abstraction level and *medium* detail depth.

- **Chapter 3:** ***Activity Classification***. In this chapter, a novel body sensor network (BSN)-based low-complexity activity classification algorithm is presented. The proposed algorithm is able to detect and classify a sequence of activities, choosing from a limited set of known activities, by observing the outputs generated by inertial sensor devices (namely, accelerometers) placed on the body of a user. To this end, a preliminary (computationally intensive) training phase, performed once, is expedient to automatically optimize the key parameters of the algorithm used in the following (computationally light) online phase for activity classification. The algorithm performance is finally experimentally evaluated, together with that of known activity classification algorithms, as a function of the number of nodes and features.

  According to Table 1.1, this topic belongs to "Activity Recognition", with *high* abstraction level and *medium* detail depth.

- **Chapter 4:** ***Evaluation of Functional Motor Tasks***. In patients with Parkinson's Disease (PD), the pathology's degree of severity is typically evaluated by means of functional motor tasks conceived to monitor specific PD symptoms. In particular, this chapter focuses on the characterization of the Leg Agility (LA) task, which is used to specifically observe the motion impairments of lower limbs. By extracting relevant kinematic variables, such as the angular amplitude and speed of thighs' motion, a comparative analysis is carried out, where the motion performance of a PD patient is compared to that of a healthy subject. The measurements are obtained through the use of two inertial sensor devices attached to the subject thighs and their accuracy is confirmed by direct comparison with an optoelectronic system (namely, Vicon). Although preliminary, the proposed analysis allows to derive significant insights in possible approaches to accurately evaluate the degree of severity of PD.

  According to Table 1.1, this topic belongs to "Partial Signal Analysis", with *low* abstraction level and medium detail depth.

- **Chapter 5:** ***Motion Reconstruction***. In this chapter, a motion reconstruction framework, where the full body motion of a user is estimated by using a small number of sensor devices (namely, 5 devices placed on the wrists, the ankles, and the pelvis of a user), is described. Due to the very limited number of devices, a data-driven approach is proposed where low-dimensional control signals (from the sensor devices) are used to estimate high-dimensional full body poses, based on a Nearest Neighbors Search (NNS) among a training dataset of previously recorded human poses. Furthermore, a key characteristic of the proposed system is that it does not rely on any external infrastructure. In the experimental analysis, the system performance is investigated considering different types of features (which, practically, correspond to different types of sensors or technologies) and different numbers and placement strategies of sensor devices.

  According to Table 1.1, this topic belongs to "Full Pose Estimation", with *medium* abstraction level and *high* detail depth.

Chapter 2

Arm Posture Recognition

Raise your arms, Maurice!
It's more fun when you raise your arms up like this!

– Julien (Madagascar: Escape 2 Africa)

In Chapter 1, several motion capture technologies have been described, each of them offering both advantages and drawbacks. To this end, the use of hybrid motion capture systems, where advantages of a component technology are exploited to compensate for the drawbacks of the other one and viceversa, represents an interesting solution that allows to obtain improved motion capture performance with respect to that obtained using the component systems. To the best of our knowledge, no works have appeared so far, in the literature, concerning the investigation of a hybrid motion capture system based on the joint use of radio and inertial signals. Therefore, in this chapter, a hybrid radio/accelerometric approach, inspired by the use of a radio (indoor) localization technique (i.e., fingerprinting), is described and preliminary experimentally investigated, with the specific aim of performing *arm posture recognition*.

2.1 Introduction

Localization systems are becoming more and more important in pervasive wireless technologies for their roles in location-aware services. GPS has been one of the milestones for outdoor localization [22]. However, its use for indoor localization is often impaired by phenomena typical of indoor scenarios, such as reflections, multipath, and fading. Therefore, indoor localization systems, which do not rely on the use of GPS, have been developed. In particular, recent research has been devoted to the so-called "fingerprinting" technique [23, 24, 25]. Fingerprinting is a localization technique where target nodes' positions are estimated on the basis of measurements, by reference nodes (anchor nodes), of the RSSs from target nodes. In particular, a radio map of the environment is first constructed through a specific offline training phase and is then used to estimate the target nodes' positions by best matching newly collected RSS values with those saved in the radio map. This technique implicitly takes into account the presence of reflections and multipath and is then particularly effective in indoor scenarios. Nevertheless, despite the appeal of fingerprinting for indoor localization applications, few works have so far appeared in the literature.

2.1.1 Chapter Contribution

Unlike existing works where fingerprinting is used to localize subjects in large (indoor) areas, in this work fingerprinting-based radio localization is exploited to perform arm posture recognition by estimating the positions of (i.e., localizing) target wireless sensor nodes properly placed on a user arm (i.e., body area localization). Different fingerprinting-based localization algorithms, either deterministic or probabilistic, are considered to estimate targets' positions. Furthermore, the proposed radio-based posture recognition system is extended to integrate the use of inertial measurements. More precisely, measurements obtained from accelerometers (available on the target nodes) are used to estimate the pitch of every node in order to refine the accuracy of the position estimation provided by the fingerprinting-based radio localization.

We remark that the key contribution of this work mainly consists in experimen-

tally investigating the feasibility of a hybrid approach which combines radio localization and inertial signals.[1] In particular, to the best of our knowledge, besides the novelty of the use of a radio localization technique (i.e., fingerprinting) for posture recognition applications, no work on the joint use of radio and inertial signals for these types of applications has appeared in the literature. Moreover, while in the literature inertial posture recognition approaches tend to use inertial devices equipped with accelerometers, gyroscopes, and/or magnetometers [28, 29, 30], our low-complexity approach relies only on acceleration measurements. The proposed system is extensively studied (optimizing key parameters) and its performance is also compared with that of a low-cost optical system (namely, Kinect). Finally, even if the proposed system is conceived to recognize simple postures of a static (not walking) user, the design and implementation of a fully portable ambulatory posture recognition system is an appealing research extension.

Chapter Outline

The rest of this chapter is structured as follows. Section 2.2 is dedicated to related work. In Section 2.3, the proposed radio/accelerometric hybrid approach is presented and described, specifically focusing on the problem of arm posture recognition. In Section 2.4, the considered experimental set-up is presented and the performance metrics of interest are then introduced. In Section 2.5, the system performance, after parametric optimization, is analyzed. Finally, in Section 2.6, advantages, disadvantages, and possible future extensions of the proposed system are discussed.

2.2 Related Work

In the literature, two main approaches, already introduced in Subsection 1.2.2, have been proposed for posture recognition: optical and inertial.

Concerning *optical posture recognition*, the widely used technology is optoelectronic (e.g., Vicon system [1]). Optoelectronic systems require the user to wear reflec-

[1] Preliminary results obtained with radio fingerprinting (without using inertial signals) can be found in [26, 27].

tive markers and to move in a space completely visible by a set of cameras. Because of their accuracy, these systems are typically used as ground truth reference for other posture recognition systems. On the other hand, their use is typically limited to clinical environments or specialized laboratories, due to their large cost and complexity.

Other optical posture recognition systems comprise the class of markerless systems. Microsoft Kinect [2], which features an RGB camera and a depth sensor (composed of an infrared camera and projector), can be considered one of the most significant example of markerless systems. Its low cost (with respect to systems like Vicon), along with its still quite good performance, has made it a widely used solution. Nevertheless, it also has some spatial and temporal limitations, which could be critical in the context of some applications. More specifically, concerning spatial limitations, it is generally known that the z axis of Kinect, which is related to the direction perpendicular to its sensor camera, has poorer resolution with respect to those of its x and y axes, which instead define the frontal plane [31]. Moreover, Kinect's sampling rate, which is typically around 30 frames per second (fps), can be a limiting factor when monitoring fast movements.

Typically, optical systems suffer from problems related to different lighting conditions and markers' occlusion. Moreover, the user movements must be limited to the area captured by the cameras. The reader is referred to [32, 33] for accurate surveys on optical posture recognition.

Concerning *inertial posture recognition*, inertial sensors are typically used to estimate the orientation of rigid body segments and, thus, to recognize the posture of a user. One of the most successful and accurate commercial products is the Xsens MVN [16], which comprises 17 inertial sensors (equipped with triaxial accelerometers, gyroscopes, and magnetometers) attached to the body of the user by a Lycra suit. The major advantage of this technology, with respect to optical systems, is that the user is completely free to move everywhere because no camera is needed. Moreover, the visibility of the nodes placed on the user body is not an issue. However, the accuracy of these systems is typically lower than that of optical systems and the cost, particularly for systems which rely on a large number of nodes and types of sensors (such as Xsens MVN), is not significantly lower than that of optoelectronic systems.

Finally, especially when used for a long time, a significant drift in the sensors' measurements can be typically observed, leading to a performance degradation. It should also be observed that, unlike optical systems, inertial systems cannot *directly* provide information about the sensor nodes' positions. Instead, estimated sensor devices' orientations are used together with properly defined biomechanical models, comprising the lengths of all body segments, to reconstruct the full body posture of the user through forward kinematics techniques [34]. The user absolute position is then estimated using advanced contact detection techniques.

Finally, solutions based on the joint use of different technologies, designed in order to tackle and overcome the limitations which characterize component technologies considered independently, have also been widely investigated. For instance, inertial/GPS, inertial/optical, and inertial/acoustic joint measurements are considered in [19], in [20], and in [21], respectively. In particular: in the inertial/GPS approach, inertial nodes equipped also with GPS receivers are used to track the user, providing better performance (especially) in outdoor scenarios (e.g., sport sessions as skiing); in the inertial/optical approach, inertial sensor nodes and cameras are jointly used, leading to a decreased freedom of movement of the user but also to an improved robustness to occlusions; in the inertial/acoustic approach, nodes equipped with microphones and speakers are used in addition to inertial sensor nodes, providing better performance (thanks to the estimation of relative distances between nodes) but suffering from "acoustic occlusions."

Unlike the above approaches, our solution represents the first attempt (to the best of our knowledge) of investigating the effectiveness of the joint use of radio and inertial (specifically, accelerometric) measurements. To this end, an arm posture recognition system is developed in order to evaluate the feasibility of the use of such hybrid approach for posture recognition. Advantages and disadvantages are inherited from the considered technologies and our solution compares to existing ones as follows.

- *Optoelectronic systems* (e.g., Vicon) provide a better accuracy than that of our approach, at the price of (i) a much higher cost and (ii) problems related to different lighting conditions and markers' occlusion.

- *Markerless systems* (e.g., Kinect) suffer from problems similar to optoelectronic systems (i.e., related to different lighting conditions and markers' occlusion). They have a cost and a performance in the order of those of our approach. However, unlike our approach, a benefit of these systems is that no specific hardware has to be placed on the user body.

- *Inertial systems* have a performance which highly depends on the used inertial sensors. However, standard inertial systems (i.e., which use devices with accelerometers, gyroscopes, and magnetometers) provide better performance than that of ours and allow the user a higher freedom of movement, at the price of a higher cost.

- *Inertial/GPS* hybrid solutions have the same advantages/disadvantages of inertial systems, but requires necessarily to be outdoor. Of course, they are very suitable for outdoor posture recognition sessions (i.e., improved freedom of movement).

- *Inertial/optical* hybrid solutions, when considering devices with costs comparable to that of our system, are almost similar (in terms of advantages and disadvantages) to our approach.

- *Inertial/acoustic* hybrid solutions are also very similar to our approach. However, even if they suffer from problems related to "acoustic occlusions," they allow improved portability and freedom of movement.

Unlike our solution, all other systems do not need a training phase—more precisely, the radio-based component of our approach relies on a (short) training phase. However, preliminary calibration phases must be typically taken into account.

To summarize, the proposed radio/inertial hybrid approach is conceived as a low-cost and low-complexity approach which overcomes typical limitations of existing systems, such as those related to different lighting condition and occlusions. On the other hand, the proposed posture recognition system requires that the user does not walk (i.e., allows a limited freedom of movement to the user). To overcome these

limitations, the design of an entirely portable posture recognition system will be discussed.

2.3 Arm Posture Recognition

In this work, "arm posture recognition" refers to the continuous estimation of the three-dimensional positions of sensor devices placed on the user arm and, therefore, to the continuous estimation of the arm orientation—this differs from "arm posture classification," where arm postures must be detected choosing from a discrete set of predefined postures. Indeed, even if our hybrid approach uses a localization technique (i.e., radio fingerprinting) to classify "known" (trained) arm postures (*posture classification*), it then properly interpolates these postures in order to recognize whole transitional movements of the arm (and not just "still postures") between two or more of the previous trained postures (*posture recognition*).

2.3.1 Fingerprinting-based Radio Localization

As anticipated in Section 2.1, fingerprinting is a robust localization technique for indoor scenarios, which are typically characterized by reflections, multipath, and fading. We now provide some intuition on the fingerprinting technique—the interested user can find more details in [26, 27]. Fingerprinting requires three kinds of nodes: target nodes, anchor nodes, and a base station. Target nodes have to be localized, whereas anchors nodes have fixed known positions and are used to generate a reference system. Finally, the base station is the processing center. Two phases are considered: a training phase, during which a radio map of fingerprints is generated, and an online phase, during which localization is performed.

During the training phase, the target node continuously broadcasts packets to be received by the anchor nodes. The latter, upon reception of the packets sent by the target, measure the RSSs and relay this information to the base station. The base station collects the RSS values and groups them into "fingerprint vectors." Finally, the base station generates a "fingerprint," i.e., a vector containing the arithmetic average of the received fingerprint vectors (namely, the vector whose elements are the average RSSs

measured by the anchors). Different physical positions of the target (i.e., "fingerprint positions") in the monitored area (properly chosen depending on the considered application) correspond and lead to different fingerprints. The entire set of fingerprints created in the training phase represents the "radio map" of the environment.[2] This radio map can be then used to run *deterministic* (i.e., based on simple comparisons between newly measured fingerprint vectors and the fingerprints of the radio map) localization algorithms. Furthermore, if one wants to use *probabilistic* versions of these algorithms (i.e., based on a more accurate statistical characterization of the RSS), during the training phase the entire Probability Mass Functions (PMFs) of the RSSs from all anchors need to be also computed and stored. Even if in the context of indoor localization a log-normal distribution of the RSS seems to be widely accepted [24], this is generally not our case, especially due to the fact that sensor nodes are placed on a human body. Therefore, the PMFs will be experimentally evaluated using the so-called histogram method, where the normalized histogram of the actual RSS measurements (during the training phase) for each fingerprint position are used [35].

After the training phase is completed, the online phase starts (following the same operations of the training phase, but now building a so-called "online vector," i.e., a single fingerprint vector containing a time snapshot of newly measured RSS values). In particular, in the online phase the radio map (deterministic approach) or the PMFs (probabilistic approach) created in the training phase are used to localize the target. Note that, once the training phase is over (i.e., the online phase starts), the target node can move freely and should not necessarily be placed in the previously trained fingerprint positions.

Given the measured online vectors, different algorithms can be used to estimate the positions of the target node. One of the simplest *deterministic* fingerprinting algorithm is the Nearest Neighbor (NN) algorithm, whose generalization is known as

[2]The major strength of fingerprinting consists of the fact that the fingerprints implicitly take into account the impact of reflections and multipath on the RSSs, i.e., this technique is "tailored" to the specific indoor environment. This makes fingerprinting virtually insensitive to indoor propagation limitations—provided that the propagation environment remains quasi-static.

kNN[3] [36]— we remark that "neighbors" here refer to fingerprints and are thus associated with specific fingerprint positions. The kNN algorithm estimates the target positions by computing a specific distance metric between the online vector and every fingerprint contained in the radio map. By applying the Shepard method [37] in order to compute a weighed interpolation of the closest neighbors, the estimated target position $\hat{\boldsymbol{s}} = (x, y, z)$ (dimension: [cm]) is given by

$$\widehat{\boldsymbol{s}} = \sum_{i=1}^{k} \frac{w_i}{\sum_{j=1}^{k} w_j} \cdot \widehat{\boldsymbol{s}}_i \tag{2.1}$$

where $\{\widehat{\boldsymbol{s}}_i\}_{i=1}^{k}$ are the fingerprint positions (i.e., physical positions) of the k closest neighbors (i.e., the fingerprints with shortest distances from the online vector) and

$$w_i \triangleq \frac{1}{d_i^{p_s} + 0.0001} \tag{2.2}$$

where: d_i is the distance computed between the i-th closest neighbor and the online vector (defined in the space of RSS vectors); and p_s is an integer larger than 0. The term 0.0001 at the denominator of Equation (2.2) is used to prevent a division by zero if the online vector is equal to one of the fingerprints. In our system, we will consider only two definitions of distance: Euclidean and Manhattan [36]. Other distance definitions can be applied to the kNN algorithm, e.g., the Mahalanobis distance, which takes also into account the contribution of covariance matrix computed for every fingerprint [36]—little performance differences are, however, observed. Observe that, when $k = 1$ (i.e., with the NN algorithm), Equation (2.1) reduces to the coordinates of the closest fingerprint position and, then, p_s has no influence on the system. Finally, note that, due to the interpolation between fingerprint positions in Equation (2.1), the estimated position $\widehat{\boldsymbol{s}}$ may (likely) differ from any of the considered fingerprint positions.

Unlike the deterministic approach, in the *probabilistic* approach (straightforwardly called p-kNN) the RSSs measured at the anchor nodes are characterized, using the

[3]Note that the term kNN, in the context of radio fingerprinting, is used to indicate the identification of the k closest fingerprints. Therefore, unlike in typical machine learning scenarios, the NN search is performed in a database of a very limited size (i.e., the number of rows is equal to the number of trained fingerprint positions) and, therefore, is not computationally intensive.

samples received in the training phase, through their entire PMFs. In this case, the estimated target position can be expressed as follows:

$$\widehat{\boldsymbol{s}} = \sum_{i=1}^{k} \frac{P(\widehat{\boldsymbol{s}}_i|\boldsymbol{r})}{\sum_{j=1}^{k} P(\widehat{\boldsymbol{s}}_j|\boldsymbol{r})} \widehat{\boldsymbol{s}}_i \tag{2.3}$$

where: $\boldsymbol{r}$ is the "online" vector; $P(\widehat{\boldsymbol{s}}_i|\boldsymbol{r})$ is the a-posteriori probability of the i-th (out of k) closest neighbor; and the k closest neighbors are chosen so that the corresponding a-posteriori probability is maximized. More specifically, using Bayes theorem, the a-posteriori probability that the target node is in the i-th fingerprint position, given that the online vector $\boldsymbol{r}$ is received, can be expressed as

$$P(\widehat{\boldsymbol{s}}_i|\boldsymbol{r}) = \frac{P(\boldsymbol{r}|\widehat{\boldsymbol{s}}_i) P(\widehat{\boldsymbol{s}}_i)}{P(\boldsymbol{r})} = \frac{P(\boldsymbol{r}|\widehat{\boldsymbol{s}}_i) P(\widehat{\boldsymbol{s}}_i)}{\sum_{\ell=1}^{L} P(\boldsymbol{r}|\widehat{\boldsymbol{s}}_\ell) P(\widehat{\boldsymbol{s}}_\ell)} \tag{2.4}$$

where: L is the number of trained fingerprint positions; $P(\boldsymbol{r}|\widehat{\boldsymbol{s}}_i)$ is computed (as anticipated earlier in this subsection) using the histogram method; and $P(\widehat{\boldsymbol{s}}_i)$ is the a-priori probability of being in the i-th fingerprint position. Since, with no movement restriction, all fingerprint positions are equally likely, it holds that $P(\widehat{\boldsymbol{s}}_i) = 1/L$. Note again that, when $k = 1$ (and thus reducing to a so-called p-NN), Equation (2.3) returns exactly the coordinates of the closest fingerprint position.

Concerning arm posture recognition, fingerprinting can be used to estimate the positions of (i.e., localize) multiple target nodes, properly placed on a user arm (e.g., one on the upper arm and one on the forearm), and straightforwardly derive the arm posture. To this end, the reference system origin must be properly chosen (e.g., the shoulder) and the user must try to keep this origin fixed during the evaluation. In this application scenario, the fingerprint positions correspond to targets' physical positions related to predefined arm postures that must be held by the user during the training phase. During the following online phase, the user can instead move his/her arm freely (always recalling not to move the reference system origin, i.e., his/her shoulder).

2.3.2 Accelerometer-based Pitch Estimation

The radio fingerprinting-based approach to posture recognition described in Subsection 2.3.1 may introduce errors, especially when a target node is in a position that differs from the trained fingerprint positions (i.e., when the arm is in an untrained posture). A possible way to improve the system performance is to estimate the arm orientation by making use of other inertial sensors (e.g., accelerometers, gyroscopes, and/or magnetometers), which the target nodes can be equipped with. In particular, considering proper combinations of these sensors (e.g., an accelerometer and a gyroscope), the orientation of a device (and, thus, of the arm) can be estimated [28].

The orientation of a device can be described by three parameters: yaw (or heading), pitch (or elevation), and roll (or bank) [38]. Specifically, it is known that a rigid body can be arbitrarily rotated by first rotating it around its z axis by an angle ψ (the yaw), then around its y axis by an angle θ (the pitch), and finally around its x axis by an angle ϕ (the roll) [38]. Observing that the acceleration measured by a still device is only due to the gravity acceleration, it can be shown that, using just an accelerometer (in order to minimize the cost of the system), the pitch of a *still* device (i.e., following the previous notation, the angle between its x axis and the horizontal plane perpendicular to the gravity direction) can be determined by observing how the gravity vector is rotated with respect to the x axis of the device. More precisely, by exploiting acceleration measurements and following simple trigonometric equations, the device pitch θ (dimension: [deg]) can be computed as follows [39]:

$$\theta = \arcsin \frac{a_x}{g} = \arcsin a_x \tag{2.5}$$

where a_x is the acceleration (in g units) measured along the x axis and g is the gravity acceleration (obviously, equal to 1 g). For ease of clarity, in Figure 2.1 a graphically intuitive representation of the geometrical meaning of Equation (2.5) is provided. Note that Equation (2.5) holds since the device is still and, therefore, the measured acceleration vector $\boldsymbol{a}$ has the same direction and norm of the gravity vector $\boldsymbol{g}$.

As the above approach is valid only if the device is still, static (constant force of gravity) and dynamic (movements or vibrations of the accelerometer itself) accelerations need to be discriminated in the case the device is moving. This problem

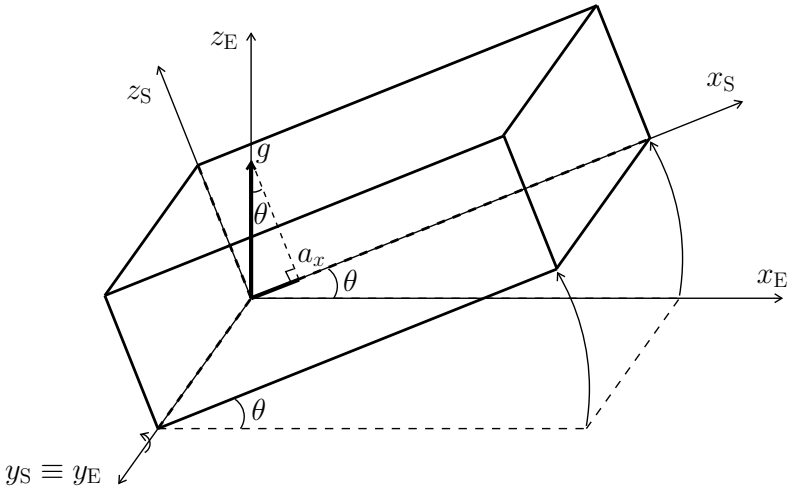


Figure 2.1: Measured acceleration when the device is rotated around its y axis (i.e., its pitch θ is different than 0). Both Earth (E subscript of axes) and device (S subscript of axes) coordinate systems are shown.

cannot be easily solved, but it can be at least mitigated by taking into account only acceleration measurements with amplitudes in $[g-\xi, g+\xi]$, where g is the gravity acceleration (i.e., 9.81 m/s^2) and ξ needs to be properly chosen with respect to the application context. This can be carried out by considering only data portions in which the user is not moving (and, thus, $\xi \cong 0$ m/s^2) or by simply applying a low-pass filter to the output acceleration signals.

When the device is moving, since it is no longer true that $|\boldsymbol{a}| = g = 1$ g and since the arcsin function accepts only values whose norm is equal to or lower than 1, Equation (2.5) should be properly rearranged. To this end, the estimated pitch of the device can be estimated as follows:

$$\theta = \arcsin \overline{a}_x = \arcsin \frac{a_x}{|\boldsymbol{a}|} \tag{2.6}$$

where $\overline{a}_x$ is the normalized acceleration (in g units) measured along the x axis.

Similarly, when the sensor device is attached to a rigid body segment of the user (e.g., the arm), the pitch (i.e., the inclination) of the considered body segment can be

computed as follows:

$$\theta = \arcsin \overline{a}_{\text{bs}} \tag{2.7}$$

where $\overline{a}_{\text{bs}}$ is now the normalized acceleration (in g units) measured along the device axis aligned with that of the considered body segment. Note that, according to Equations (2.5), (2.6), and (2.7), the pitch always belongs to $[-90^{\text{o}}, 90^{\text{o}}]$.

Since the estimated inclination of a body segment is not sufficient alone to provide information about its position, a proper human biomechanical model, which assumes to know the involved body segments' lengths and how they are linked together, should be considered.

2.3.3 Recursive Estimation Process

Referring to the posture recognition of a user arm, it will be now described how estimates obtained from (i) radio fingerprinting and (ii) pitch estimation should be properly combined. In particular:

- The pitch θ of each device (and, thus, of the corresponding body segment) is computed using Equation (2.7). This will lead to the estimation of the z coordinate of the target node.
- Estimates of the (x, y, z) coordinates of the devices are available from radio localization (through fingerprinting).

Furthermore, we preliminary introduce a few geometric assumptions.

- A (properly defined) fixed origin **O** is chosen.
- One device is attached to each adjacent rigid body segment between the previously chosen origin **O** and the farthest distal point **Z** of the body part whose posture needs to be estimated. More precisely, for arm posture recognition, **O** could be the shoulder and **Z** could be the wrist. Therefore, two devices should be used, attached, respectively, to the upper arm and to the forearm.
- The length d of each body segment on which the devices are mounted is known (e.g., the lengths of the upper arm and of the forearm).

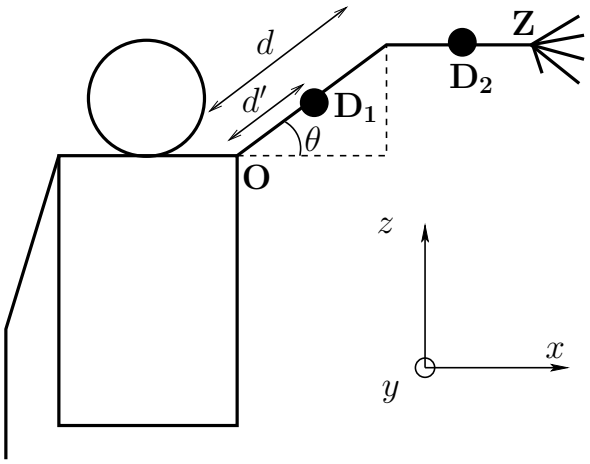


Figure 2.2: Pictorial description of the considered scenario. The points of interest used in the algorithm are highlighted. In particular: **O** and **Z** are the three-dimensional coordinates of the origin point and the farthest distal point that needs to be estimated, respectively; $\mathbf{D}_1$ and $\mathbf{D}_2$ are the three-dimensional coordinates of target 1 and target 2, respectively; d is the actual length of the arm segment considered at the first step (in this case, the upper arm); d' is the distance between the considered sensor device (with coordinates $\mathbf{D}_1$) and the corresponding body segment's joint proximal to the fixed origin (in this case, the origin point with coordinates **O**, i.e., the shoulder); and θ is the pitch of the considered sensor device and, thus, of the corresponding body segment (in this case, the upper arm).

- The distance d' between each device and the corresponding body segment's joint proximal to the fixed origin is known (e.g., the distance between the shoulder and the device attached to the upper arm).

In Figure 2.2, a pictorial description of the scenario, with highlighted points of interest, is shown.

In the following, we introduce a recursive estimation process, according to which adjacent body segments are considered at consecutive steps. Specifically, all adjacent body segments comprised between **O** and **Z** are considered, starting from the body

segment starting in **O** and ending with the body segment ending in **Z**. Furthermore, for each considered body segment, the positions of its two extremes (i.e., the body segment joints) are chosen as reference points: the position of the proximal joint (i.e., the joint closest to the body) is denoted as point $\mathbf{A} = (x_\mathrm{A}, y_\mathrm{A}, z_\mathrm{A})$ and the position of the distal joint (i.e., the joint farthest from the body) as point $\mathbf{B} = (x_\mathrm{B}, y_\mathrm{B}, z_\mathrm{B})$. In the specific case of the arm shown in Figure 2.2, at the first step **A** corresponds to **O**, whereas at the last step **B** corresponds to **Z**.

In order to estimate the posture of the user arm, the postures of the upper arm and of the forearm have to be estimated (or, in other words, the positions of the elbow and the wrist, with respect to the shoulder, need to be identified). In this case, the proposed recursive estimation strategy involves two steps. To this end, two target nodes need to be used: the first on the upper arm and the second on the forearm—more details on the experimental set-up will be given in Subsection 2.4.1. For ease of clarity, let us denote their physical positions as $\{\mathbf{D}_i\}_{i=1}^{2}$, where $\mathbf{D}_i = (x_{\mathrm{D}_i}, y_{\mathrm{D}_i}, z_{\mathrm{D}_i})$.[4] The origin $\mathbf{O} = (0,0,0)$ will be the shoulder, whereas the farthest distal point **Z** will be the wrist.

As shown in Figure 2.2, at the first step, since we are focusing on the upper arm, **A** corresponds to the shoulder (i.e., the origin **O**), whereas **B** corresponds to the elbow (whose physical position is currently unknown). We then consider the first device, denoted as point $\mathbf{D}_1$ and (in our scenario) positioned in the middle of the upper arm, i.e., between **A** and **B**. Its initial position estimate, denoted as $\widehat{\mathbf{D}}_1$, is recovered through radio localization, whereas its pitch θ (which corresponds to the inclination of the upper arm) is estimated using Equation (2.7). Finally, d is the distance between **A** and $\mathbf{D}_1$, whereas d' is the length of the upper arm (i.e., the distance between **A** and **B**).

Taking into account a polar coordinates framework, the arm posture is then estimated through the computation of the pitch and the heading of all the body segments of the arm. In particular, the main idea is to use the initial position estimate $\widehat{\mathbf{D}}_1$ (recov-

[4]Note that we assume that the z axis of the reference system is in the vertical direction (i.e., that of the gravity vector). The x and y axes may be instead conveniently chosen as perpendicular axes in the horizontal plane. In particular, we will chose the y axis as the forward direction and the x axis as the direction normal to the user sagittal plane.

ered from radio localization) to infer information about the heading ψ (dimension: [deg]) of the considered body segment. The heading information is then combined with that of the pitch θ of the same body segment (estimated using the accelerometer) in order to refine and correct $\widehat{\mathbf{D}}_1$ (taking also into account a proper biomechanical model). Therefore, the hybrid approach consists in using the (i) radio fingerprinting to estimate the heading of each arm segment and the (ii) inertial signals to estimate the inclination.

More precisely, the heading ψ of the considered body segment can be computed as

$$\psi = \arctan 2\left(y_{\widehat{\mathrm{D}}_1} - y_{\mathrm{A}}, x_{\widehat{\mathrm{D}}_1} - x_{\mathrm{A}}\right) \tag{2.8}$$

where the function $\arctan 2$ behaves as the standard arctan function but, in addition, provides information about the quadrant of the computed angle, so that the heading ψ belongs to $[0^{\mathrm{o}}, 360^{\mathrm{o}}]$.

Finally, considering the estimated pitch (θ) and heading (ψ), the adjusted estimated coordinates of the first device, denoted as $\overline{\mathbf{D}}_1$, can be expressed as

$$\overline{\mathbf{D}}_1^{\mathrm{T}} = \begin{pmatrix} x_{\overline{\mathrm{D}}_1} \\ y_{\overline{\mathrm{D}}_1} \\ z_{\overline{\mathrm{D}}_1} \end{pmatrix} = \begin{pmatrix} x_{\mathrm{A}} + d' \cdot \cos\theta \cdot \cos\psi \\ y_{\mathrm{A}} + d' \cdot \cos\theta \cdot \sin\psi \\ z_{\mathrm{A}} + d' \sin\theta \end{pmatrix} \tag{2.9}$$

where $(\cdot)^{\mathrm{T}}$ denotes vector transposition. A graphical intuition of Equation (2.9) is given in Figure 2.3. Furthermore, the estimated coordinates of $\mathbf{B}$ (namely, the elbow) can be similarly expressed as follows:

$$\mathbf{B}^{\mathrm{T}} = \begin{pmatrix} x_{\mathrm{A}} + d \cdot \cos\theta \cdot \cos\psi \\ y_{\mathrm{A}} + d \cdot \cos\theta \cdot \sin\psi \\ z_{\mathrm{A}} + d \sin\theta \end{pmatrix}. \tag{2.10}$$

The above estimation process can be repeated to estimate the position of the second device $\mathbf{D}_2$, positioned in the middle of the forearm (as shown in Figure 2.2), and of the wrist. To this end, $\mathbf{A}$ and $\mathbf{B}$ should be updated (i.e., $\mathbf{A}$ would correspond to the elbow, whereas $\mathbf{B}$ would correspond to the wrist) and Equations (2.9) and (2.10) can be used again considering proper values of θ, ψ, d', and d (namely, the ones related

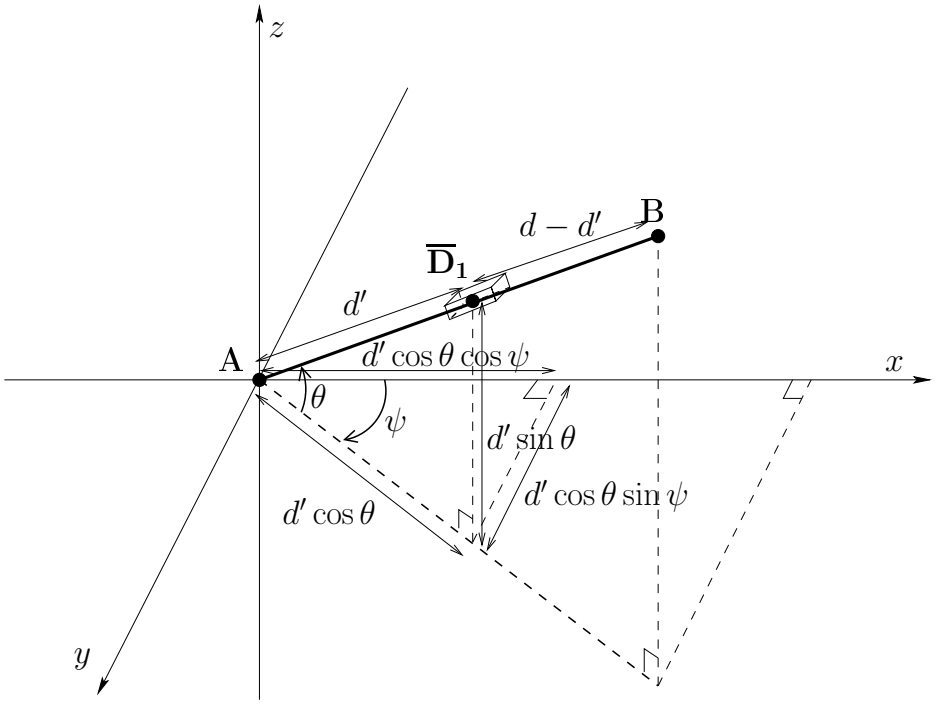


Figure 2.3: Graphical intuition of Equation (2.9) in the specific case of the first considered body segment (i.e., the upper arm). The heading (i.e., ψ) and the pitch (i.e., θ) of the segment are highlighted. For ease of clarity, the device $\mathbf{D}_1$ is pictured along segment **AB** and the projection of the segment on the horizontal plane is also shown. Note that, for the case of the upper arm, **A** and **B** correspond, respectively, the shoulder and the elbow of the user.

to the arm segment between the updated points **A** and **B**, i.e., the forearm). Finally, since after this second step the updated point **B** (i.e., the wrist) corresponds to **Z**, the recursive process ends and the arm posture is estimated.[5] Note that as the position of **B** is estimated and, therefore, may have errors, it is likely that the estimated coordinates of $\mathbf{D}_2$ will be less accurate than those of $\mathbf{D}_1$.

Finally, we remark that the integration of the accelerometers into our system does not have any impact on the fingerprinting training phase. Indeed, all the acceleration measurements are taken into account only during the online phase.

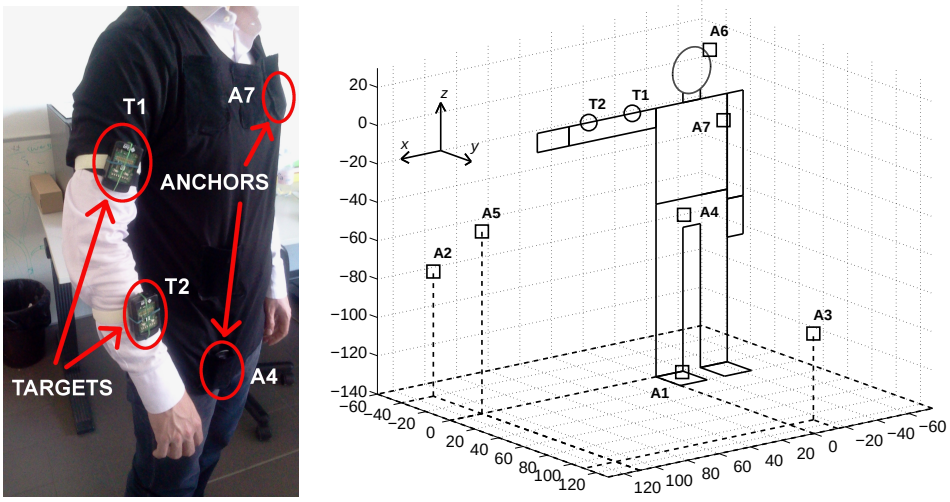
2.4 Experimental Set-up

2.4.1 Experimental Testbed

SunSPOT devices have been used for the experimental testbed. SunSPOTs are wireless devices equipped with a triaxial accelerometer and an IEEE 802.15.4 compliant radio interface with an on-board antenna and up to 100 m transmission range [40].

In the current chapter, we consider a testbed which extends the one proposed in [26]. In particular, a Body Area Network (BAN) with target nodes on the user arm is still considered, but now the anchor nodes are in part placed (and fixed) in the surroundings of the user and in part attached on his/her body—in [26], all anchors are fixed and outside the body. This is obtained through the use of (i) a home-made t-shirt with folders where anchors can be placed and (ii) a hat with an anchor attached to it. Although the presence of some anchor nodes at fixed (outside the body) positions still forces the user to remain in its initial position in the room (in order to make the fingerprinting technique work consistently), the proposed testbed is a first step toward a fully portable arm posture recognition system—this extension is the subject of our current research activity. Furthermore, a noisier (and, thus, more realistic) environment is here taken into account (e.g., with more than one people moving in the room while testing the system and in the presence of multiple active WiFi net-

[5] Observe that this process can be recursively repeated, should more consecutive body segments be considered (e.g., by adding a device for the hand in our testbed).



(a) (b)

Figure 2.4: Considered experimental posture recognition set-up: (a) plot of on-body set-up; (b) scheme of the overall set-up. The positions of the anchor nodes (i.e., A1 to A7) and of the target nodes (i.e., T1 and T2) are highlighted. Note that A2, A3, and A5 are fixed in the surroundings of the user, whereas A1, A4, A6, and A7 are attached on his/her body.

works), whereas in [26, 27] the system performance is analyzed in more controlled (interference-free) scenarios. Therefore, the presented results are already reflecting the performance of a system in a quite adverse (thus realistic) home scenario.

In the present experimental testbed, 2 SunSPOTs, acting as targets (i.e., target 1 and target 2), are placed on the right arm of the user, as shown in Figure 2.4 (a) (where it is also possible to see the home-made t-shirt): the first node (i.e., target 1) is on the upper arm and the second (i.e., target 2) is on the forearm. $N = 7$ anchor nodes are considered: 4 of them are placed on the user body, whereas the remaining 3 are placed in its proximity.[6] For ease of clarity, an illustrative representation of the

[6]We remark that, in the presented experimental analysis, the user is forced to keep his/her body still. Therefore, the anchor nodes attached on his/her body can be actually considered as fixed (still) nodes. The choice of attaching them directly on the user body will be further discussed and motivated

Table 2.1: Fingerprint positions coordinates (considered during the training phase) with respect to the user shoulder. Every fingerprint position corresponds to a specific arm posture.

Fingerprint Positions	Target 1			Target 2		
	x [cm]	y [cm]	z [cm]	x [cm]	y [cm]	z [cm]
P1	0	0	-15	0	0	-42.5
P2	0	0	15	0	0	42.5
P3	0	15	0	0	42.5	0
P4	15	0	0	42.5	0	0
P5	-11	11	0	-34	21	0

overall experimental set-up is shown in Figure 2.4 (b), where a map of the positions of the nodes is shown. Fixing the origin of our new reference system on the shoulder (and thus forcing the user to move the arm and keep the shoulder still), 5 fingerprint positions per target node (i.e., P1, P2, P3, P4, and P5), whose coordinates are defined in Table 2.1 and shown in Figure 2.5, are considered, corresponding to 5 simple arm postures. The choice of the 5 arm postures (and, thus, of the 5 fingerprint positions) is expedient to cover as uniformly as possible the surroundings of the user arm.

The packet transmission rate of target nodes is around 30 pck/s. The acceleration, locally (at each target node) sampled at 100 Hz, is also consistently down-sampled to the same rate (i.e., 30 Hz). Therefore, every second, the base station receives, for each target, around 30 new fingerprint vectors (containing new RSS measurements at anchor nodes) and new acceleration measurements. The synchronization between the sensor nodes in the system is performed by aligning the internal clocks of every sensor node to that of the base station, at application deployment stage, and by inserting a timestamp field inside each transmitted packet.[7]

in Section 2.6.

[7]Note that, even if for the purpose of our experimental analysis this is sufficient to guarantee a fair synchronization for a few hours, a future system should try to define specific communication protocols/mechanisms able to achieve, e.g., through a sparse exchange of synchronization packets, a long-

In the training phase, the user has been asked to keep each of the 5 arm postures for about 30 s. As shown in [27], the time interval, during which each arm posture should be kept fixed to train each corresponding fingerprint position (i.e., 30 s), has been chosen in order to collect a number of fingerprint vectors (i.e., about 1000 for each target node) sufficient for the convergence of the evaluated fingerprints. During these 30 s, the considered fingerprint position is trained as explained in Subsection 2.3.1 and the related fingerprint is generated.[8] Once the training phase has terminated, the online phase is split into two parts in order to properly test the system performance. In particular: the user is first asked to replicate the 5 trained arm postures (in order to test the system in *static* conditions); furthermore, a few transitional movements, which start and end at two of the trained arm postures, are also executed (in order to test the system in *dynamic* conditions). Note that, for the purpose of localization, fingerprinting is performed independently for each target node. With reference to Table 2.1, this means that: P1 corresponds to a specific posture of the arm but, obviously, to two fingerprint positions (i.e., one for each target node) and, therefore, to two fingerprints; P2 - P5 can be interpreted likewise. The transitional movements that the user is asked to perform are shown in Figure 2.5 and can be summarized as follows:

- lower the arm from position P2 to position P1, passing through position P3 (Figure 2.5 (a));
- move horizontally the arm from position P4 to position P5, passing through position P3 (Figure 2.5 (b));
- raise the arm from position P1 to position P3 (Figure 2.5 (c));
- move horizontally the arm from position P3 to position P4 (Figure 2.5 (d)).

lasting synchronization of system devices.

[8] Note that, even if some slight variations of the kept arm postures are unavoidable, the measured fingerprint vectors should be relatively close to each other (in the RSS space). Therefore, the error introduced in the computation of the fingerprints is typically negligible.

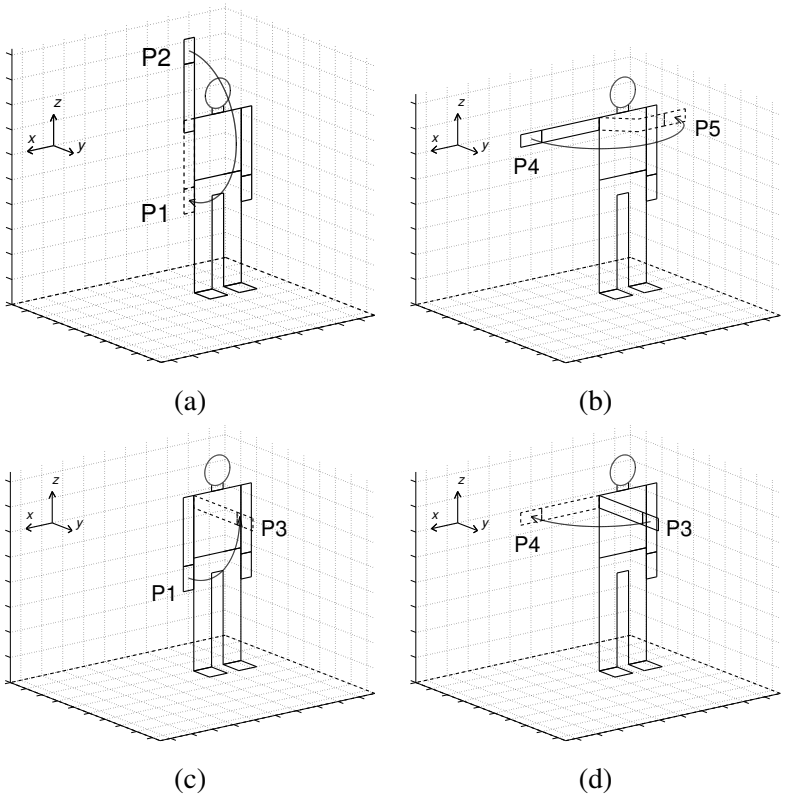


Figure 2.5: Transitional movements, performed by the user during the online phase, between the 5 trained arm postures (used to train the corresponding 5 fingerprint positions, i.e., P1, P2, P3, P4, and P5) considered in the training phase.

The previous sequence of transitional movements is repeated by the user twice, resulting in a total of 8 movements. The user is asked to perform each transitional movement at constant speed (typically leading to movements with duration of $2 \div 4$ s). Since the segmentation of the time portions containing each transitional movement is only expedient to the performance analysis (and not to the correct system behavior), it has been performed manually. During the online phase, each new fingerprint vector (i.e., the online vector) is used to evaluate the current arm posture.

Finally, for the purpose of a *trend-wise* comparison, we run at the same time an

arm posture recognition session performed through the Microsoft Kinect system [2]. In particular, we put the Kinect device in front of the user (at a distance of about 3 m) and run the Skeletal Viewer demo application (included in the Kinect libraries). Thank to this application, we were able to save the coordinates of the joints of the user arm for each time sample and, therefore, the coordinates of the target nodes, which have been compared with the ones estimated with our system. Given that Kinect samples video frames at around 30 fps, a direct comparison with our system is straightforward. Five users have been considered for the experimental analysis.

2.4.2 Performance Metrics

Typically, the performance of an arm posture recognition system is evaluated by comparing the estimated arm position with that estimated with optoelectronic systems (e.g., Vicon) and considered as "ground truth." For ease of clarity, we remark that our goal is not to derive the *exact* performance of the proposed system, but, rather, to have a rough idea of its achievable performance, in a comparative way with respect to Kinect. Since we do not have access to any optoelectronic system and since we are interested on *trend-wise meaningful results*, the performance of our system has been evaluated with respect to predefined arm postures and trajectories of the considered body segments, where true target positions have been manually derived—we believe that this is reasonable, as the considered postures and transitions (described in Subsection 2.4.1) are very regular. Due to the geometry of the human body (and, specifically, the arm model) and the simplicity of the chosen and tested transitional movements (described in Subsection 2.4.1), the considered trajectories are simple horizontal/vertical arcs in the three-dimensional space (as shown in Figure 2.5). Therefore, these trajectories can be accurately reproduced by a careful user. Even if, in this way, some errors are certainly introduced in the measurements, due to the simplicity of the chosen postures and movements these errors tend to be limited to a few centimeters. The performance of Kinect is similarly evaluated.

The performance of a localization algorithm can be characterized in terms of *accuracy* (dimension: [cm]) and *precision* (dimension: [%]) [41]. In particular: *accuracy* is defined as the distance between the estimated position and the true target

position; *precision* is the percentage of successful position estimates within a given accuracy. For instance, a precision of 60% with an accuracy of 10 cm means that 60% of the estimation errors made by the system are lower than or equal to 10 cm. The performance of a localization algorithm can be then fully described by evaluating the precision as a function of the accuracy, i.e., through a curve. Note that the best achievable performance corresponds to the (0, 1) point in the precision/accuracy graph, whereas the performance decreases as much as the distance between the precision/accuracy curve and this point increases.

Though the accuracy/precision curve provides significant insights on the system performance, in order to provide a more concise (yet insightful) system performance metric, the position error (PE), averaged along the duration of the considered arm movements, is also evaluated. Practically, given a movement of duration T and starting at t_0, PE (dimension: [cm]) is computed as follows:

$$\begin{aligned} \mathrm{PE} &\triangleq \frac{\sum_{t=t_0}^{t_0+T-1} d_{\mathrm{E}}\left(\hat{\boldsymbol{s}}_t, \boldsymbol{s}_t\right)}{T} \\ &= \frac{\sum_{t=t_0}^{t_0+T-1} \sqrt{\left(\hat{x}_t - x_t\right)^2 + \left(\hat{y}_t - y_t\right)^2 + \left(\hat{z}_t - z_t\right)^2}}{T} \end{aligned} \tag{2.11}$$

where: $\hat{\boldsymbol{s}}_t = (\hat{x}_t, \hat{y}_t, \hat{z}_t)$ and $\boldsymbol{s}_t = (x_t, y_t, z_t)$ are the estimated and the "true" target positions, respectively, at the t-th epoch; and d_{E} stands for Euclidean distance.

2.5 Results

The performance of the proposed posture recognition system has been evaluated in both static and dynamic conditions. Concerning the system performance in *static* conditions (not reported here for lack of space), our results show that the deterministic approach performs slightly better than the probabilistic one: the average PEs are 3 cm and 4.8 cm for the deterministic and the probabilistic approaches, respectively. This confirms previous results in [26, 27].

We now focus on the performance of our system in *dynamic* conditions (i.e., when the user arm moves over time). To this end, each user is asked to perform the sequence of the four transitional movements defined in Subsection 2.4.1 and shown in Figure 2.5. Moreover, for repeatability purposes, the same sequence is repeated twice (leading to a total of 8 transitional movements).

In order to evaluate the system performance, the system parameters used in Equations. (2.1) and (2.3) (i.e., the number of anchors N and the related optimal subset, the values of k and p_s, and the distance metrics) are optimized in order to minimize the PE between the real targets' positions and their estimates. In particular, the optimal system parameters are determined both (i) considering independently each distinct transitional movement (i.e., determining 8 set of optimal parameters, one per movement) and (ii) considering jointly all transitional movements (i.e., determining a unique set of optimal parameters to be used for every possible movement).[9] Note that the presented results are averaged over the five users. Therefore, the optimal parameters are chosen as the ones which jointly maximize the average performance of all the users (i.e., they are not optimized independently for each user). Both deterministic and probabilistic approaches are considered. Finally, the performance of Kinect is also evaluated.

The system performance obtained by *optimizing independently each distinct transitional movement* is shown in Table 2.2.[10] It is easy to see that the posture recognition system works at its best when the user performs movements 1 (or 5) and 3 (or 7). This provides insightful information about the system behavior which can be easily understood by observing the nature of these movements. Indeed, these movements are all

[9]Note that, for a practical implementation of the system, a unique set of optimal parameters must be considered. Nevertheless, the optimization of the system parameters performed considering independently each transitional movement allows to evaluate an insightful lower bound of the performance of our system.

[10]Observe that, given that the user performs twice the same sequence of 4 movements, the movements from 5 to 8 (as referred to in the "Transitional Movements" column) are repetitions of the same movements that are identified by the labels from 1 to 4. Therefore, it is expected that the performance (and optimized parameters) in correspondence to movement 1 will be quite similar to that in correspondence to movement 5—the same comment applies to the pairs of movements 2-6, 3-7, and 4-8.

Table 2.2: Posture recognition performance (optimized independently for each distinct transitional movement).

Target Nodes	Transitional Movements	Deterministic Approach						Probabilistic Approach				Kinect
		PE [cm]	N	anchors subset	distance	k	p_s	PE [cm]	N	anchors subset	k	PE [cm]
Target 1	1	**1.2**	3	{1, 5, 6}	Manhattan	5	1	**2.6**	1	{5}	3	**8.4**
	2	**4**	3	{5, 6, 7}	Euclidean	2	2	**6.7**	2	{6, 7}	3	**7.1**
	3	**1.3**	4	{2, 4, 5, 6}	Manhattan	1	-	**1.3**	1	{1}	3	**7.5**
	4	**4.5**	4	{3, 4, 5, 7}	Manhattan	4	6	**7.4**	1	{4}	4	**11.4**
	5	**1.3**	3	{1, 5, 6}	Manhattan	5	1	**3.4**	1	{1}	3	**8.1**
	6	**4.2**	3	{5, 6, 7}	Euclidean	3	2	**6.3**	2	{6, 7}	3	**6.9**
	7	**0.4**	2	{2, 4}	Manhattan	1	-	**0.4**	1	{1}	3	**7.8**
	8	**4.4**	4	{1, 4, 5, 7}	Euclidean	3	2	**6.3**	2	{6, 7}	2	**7.7**
Target 2	1	**3.3**	2	{2, 3}	Manhattan	1	-	**10.7**	2	{2, 7}	1	**18.1**
	2	**15**	4	{1, 2, 3, 6}	Manhattan	1	-	**21.9**	2	{3, 4}	1	**12**
	3	**3.6**	3	{3, 4, 7}	Euclidean	5	4	**4.5**	2	{4, 7}	1	**15.6**
	4	**12.5**	6	{1, 2, 3, 4, 5, 6}	Euclidean	2	6	**19.9**	4	{1, 2, 3, 4}	2	**25.9**
	5	**3.6**	2	{2, 3}	Euclidean	1	-	**9.4**	3	{2, 3, 7}	1	**17.2**
	6	**13.9**	3	{2, 3, 6}	Euclidean	1	-	**22.8**	4	{2, 5, 6, 7}	1	**12.5**
	7	**1.2**	3	{3, 4, 7}	Euclidean	1	-	**2.4**	3	{3, 4, 7}	3	**17.1**
	8	**12.7**	5	{3, 4, 5, 6, 7}	Manhattan	2	4	**15.1**	4	{2, 4, 6, 7}	2	**17.6**
Target 1 + Target 2	$1 \div 8$	**average PE = 5.4 cm**						**average PE = 8.8 cm**				**average PE = 12.6 cm**

"vertical" movements (i.e., the majority of the movements of the arm is concentrated on a vertical plane), whereas the others are all "horizontal" movements (i.e., the majority of the movements of the arm is concentrated on a horizontal plane). Taking into account our system, it is easy to notice that the "vertical" movements benefit more from pitch estimation (performed through the accelerometers) than from heading estimation (which derives entirely from the radio localization). On the other hand, the opposite comment stands for the "horizontal" movements, where the heading of the arm varies typically more than its pitch. This is then a strong symptom of the importance of the role of the accelerometers in our system.[11] As expected, in Table 2.2, it can also be observed that the nature of the movements (i.e., "vertical" or "horizontal") has no influence on the Kinect performance (which is reported in the last column of the table). Finally, in order to provide a general and concise indication of the system performance, the last row of the table reports the average PE (over all movements). It can be noticed that, in terms of average PE, our system slightly outperforms Kinect.

[11] In order to improve the accuracy of estimation of "horizontal" movements the use of a gyroscope and/or a magnetometer is expedient.

Table 2.3: Posture recognition performance (optimized on the whole sequence of movements).

Target Nodes	Deterministic Approach						Probabilistic Approach				Kinect
	PE [cm]	N	anchors subset	distance	k	p_s	PE [cm]	N	anchors subset	k	PE [cm]
Target 1	**4.2**	4	{3, 5, 6, 7}	Euclidean	4	2	**6.1**	2	{6, 7}	5	**8.1**
Target 2	**14**	3	{2, 3, 6}	Euclidean	1	-	**17.9**	4	{2, 3, 4, 7}	1	**17.1**
Target 1 + Target 2	**average PE = 9.1 cm**						**average PE = 12 cm**				**average PE = 12.6 cm**

Nevertheless, we remark that the comparison with Kinect is only trend-wise meaningful and, thus, a limited effort has been dedicated to arrange the best acquisition conditions (for example, considering more favorable light conditions or optimized user's distance for Kinect). However, no significant performance improvement is expected with Kinect. In fact, previous investigations on the purpose of Kinect, in similar application scenarios, have also shown the presence of significant error peaks (even higher than $10 \div 15$ cm) associated with similar arm movements [42, 31]. As already anticipated, since it is impractical to tune the system differently for each movement, we remark that the average results in Table 2.2 are not directly applicable. Nevertheless, they can be used as "best-case" benchmark values and, therefore, to lower bound the performance of the posture recognition system.

Considering now a *global* (more practical) *optimization* computed upon the whole sequence of movements (i.e., by selecting a single set of parameters for all movements), the obtained system performance is shown in Table 2.3. As expected, the average PE is higher than in the previous case. In particular, it can be seen that the proposed arm posture recognition system guarantees an average PE around 10 cm. Specifically, the deterministic approach (with average PE equal to 9.1 cm) performs better than the probabilistic approach (whose corresponding average PE is 12 cm). As before, our system slightly outperforms the Kinect system (whose performance is again indicated in the last column of the table). The fact that the deterministic approach outperforms the probabilistic one (even if the opposite is intuitively expected) is likely due to the fact that the probabilistic version of the proposed system may need a longer training time or, at least, a larger number of training samples. In fact, we have observed that some RSS PMFs (determined during the training phase) have

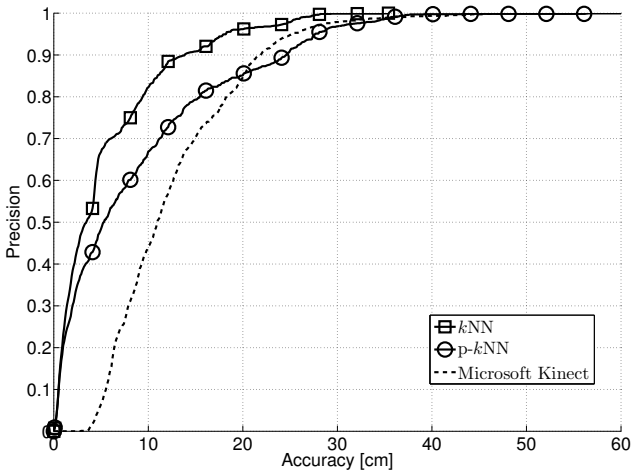
almost zero variance, which implies a lower flexibility (in the online phase) to handle small variations and noise in the measured RSS values.

In order to better investigate our system behavior, in Figure 2.6 we finally evaluate the accuracy/precision performance of the proposed system. Both deterministic and probabilistic approaches are considered. The Kinect performance is also shown for comparison purposes. In Figure 2.6 (a) (which is related to results shown in Table 2.2), the parameters are optimized independently for each distinct transitional movement and, therefore, the obtained performance corresponds to a best-case scenario, whereas, in Figure 2.6 (b) (which is related to results shown in Table 2.3), the parameters are optimized considering jointly all possible transitional movements. As already observed in Tables 2.2 and 2.3, it can be observed that, in both cases (a) and (b), the deterministic approach slightly outperforms the probabilistic one and the Kinect system. Note that, in case (b), the Kinect system outperforms the probabilistic version of our system for accuracy values higher than 15 cm (in other words: if a PE larger than 15 cm is tolerable the Kinect system is more suitable than the probabilistic approach of our system).

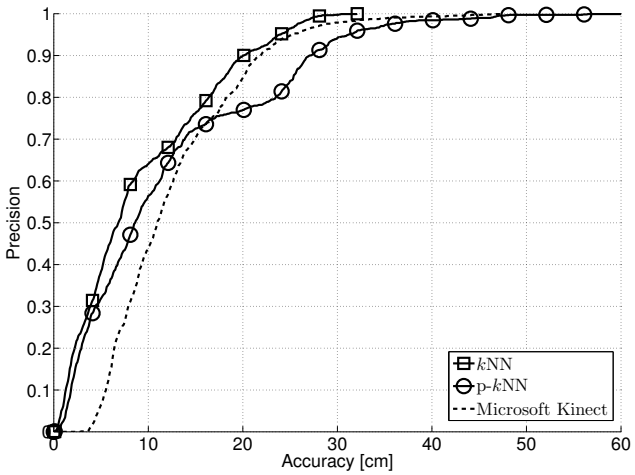
2.6 Discussion and Conclusions

As already anticipated throughout the chapter, the main goal of this work is to investigate the feasibility of the combination of two known technologies, i.e., *radio (fingerprinting-based) localization* and *inertial (accelerometric) measurements*. More specifically, the efficacy of the considered hybrid radio/inertial approach in recognizing arm postures and movements has been evaluated. Our experimental analysis has been carried out using a SunSPOT testbed. Nevertheless, the proposed approach is more general and can be applied to any device equipped with an accelerometer and a radio interface.

As previously discussed in Section 2.2, most of current state-of-the-art solutions for arm posture recognition rely on optical and/or inertial approaches. In particular, optical approaches, by using costly systems composed of a set of cameras and reflective markers placed on the user, allow to obtain an exceptional accuracy, but suffer



(a)



(b)

Figure 2.6: Precision, as a function of the accuracy, for the arm posture recognition experimental testbed, considering both deterministic and probabilistic approaches. The optimal configuration of the parameters is considered for every curve. In particular, in (a) every distinct transitional movement is optimized independently, whereas in (b) the optimization is carried out for the whole sequence of transitional movements. A comparison with the Kinect performance is also offered.

from problems related to different lighting conditions and markers occlusions. On the other hand, these limitations are effectively tackled by inertial systems, which do not depend on cameras and, thus, allow an improved freedom of movement and portability, at the price of a lower accuracy. Inertial posture recognition is typically based on the estimation of the orientations of wireless devices placed on the body of a user, carried out by a proper fusion of measurements from different sensors (i.e., accelerometers, gyroscopes, and magnetometers). The performance of inertial systems is good when all types of sensors are used but the cost tends to increase when using a large number of sensors.[12] However, if one attempts to reduce system costs (related to the use of multiple sensors per device) by using only partial subsets of the previous sensors, the performance of inertial systems decreases because of implicit limitations in determining a stable and complete orientation of the devices when using independently each considered sensor. For instance: accelerometers cannot provide information about the devices' heading; magnetometers suffer from magnetic fields disturbance; and gyroscopes present a bias drift.

The idea behind our work is that of reducing the cost (and complexity) of standard inertial systems by designing a BAN-based system where devices are equipped with only an accelerometer (and a radio interface). To this end, note that the proposed system does not require computationally intensive operations. Indeed, the operations involved in the algorithms can be executed at most in polynomial time complexity (mostly due to the trigonometric operations on the inertial measurements). Note also that the proposed system "as-is," when deployed on SunSPOTs, is already able to run with a time delay of a few tens of milliseconds (i.e., almost real-time), even with a not completely optimized code implementation.

The proposed system would be suitable to estimate the arm inclination but cannot provide information about its heading. To this end, radio localization is also used in order to localize (upon proper training of the system) the devices and, then, to esti-

[12]Note, for instance, that the cost of the majority of commercial devices used for motion capture purposes (i.e., typically equipped with an accelerometer, a gyroscope, and a magnetometer) is at least twice (or even three times) that of a simpler device equipped just with an accelerometer. In fact, the cost of a gyroscope (or of a magnetometer) alone is already twice that of an accelerometer.

mate the corresponding arm segment heading. The proposed hybrid approach allows to inherit the advantages of inertial systems (in terms of robustness against typical limitations of optical systems) but, on the other hand, introduces a major limitation. Indeed, due to the fact that radio localization relies on a preliminary training phase and anchor nodes are fixed in the surroundings, the user is forced to keep his/her body (with the exception of the arm) still during the arm posture recognition session (or, at least, he/she is forced to keep the shoulder, corresponding to the system origin, still).

Even if the requirement of still users would not be a true limitation for several rehabilitation applications (where users are not supposed to move around while performing specific exercises), some applications may require that the user moves while performing an exercise. Therefore, an appealing extension of the proposed system should aim at allowing the user to move freely while performing the exercises. As a first step in this direction, in this chapter we have also considered anchor nodes placed directly on the user body and verified that they can be effectively used as reference nodes. Furthermore, a future design of the proposed system could exploit radio localization to directly estimate distances between (mobile) nodes placed only on the user body, possibly removing the distinction between anchor and target nodes (i.e., every node could be at the same time an anchor and a target node). This could have a direct relevance to rehabilitation engineering and is currently under investigation.

Another possible limitation of the proposed system, especially if the application scenario involves the evaluation of elderly people movements, is the required training phase, which could be sometimes hard to be correctly performed by the user. To this end, a possible solution may consist in allowing an impaired user to perform it and then rely on techniques able to "map" the training output in order to be used with the actual patient (similar techniques have already been taken into account for Kinect). In general, since this work represents a feasibility study on the hybrid integration of radio and inertial signals, the system "as-is" is not intended to be directly applied in real scenarios (e.g., rehabilitation applications) and, therefore, further extensions and methods should be considered in order to improve its robustness.

Our results show that the proposed system, in its current design, can provide a performance similar to that of Kinect (which is a concurrent low-cost posture recog-

nition system). In addition, simple localization algorithms, namely deterministic (the *k*NN algorithm) and probabilistic (p-*k*NN), have been considered to recognize arm postures—due to their low computational complexity, these algorithms can be implemented on the majority of current low-cost devices. The system parameters have been optimized in order to minimize the average PE and values around 10 cm have been obtained. A PE around 10 cm can surely be sufficient for the majority of posture *classification* applications. The system performance has also been evaluated in terms of precision and accuracy. In particular, the deterministic version of the proposed hybrid localization algorithm outperforms the probabilistic one and slightly outperforms an optical Kinect system.

Since the aim of posture *classification* applications is just to determine the posture of a user choosing among a discrete set of previously trained postures (and, thus, to discriminate between known postures), our system works properly if the considered postures are sufficiently spatially distinct (i.e., the distance, for each pair of different postures, of at least two corresponding body segments equipped with sensor nodes is higher than 10 cm). Concerning posture *recognition* (where one is not interested on the discrimination between priorly known discrete and well-distinguished postures but, rather, on the recognition of generic postures), the performance of our system highly depends on the application requirements. In particular, it could be used to monitor the recovery improvement after an arm surgery, in the cases where it is relevant to discriminate a few cases (e.g., the arm can be raised half-way or all the way). For applications where it is necessary to discriminate between very close positions, then the accuracy needs to be improved. To this end, possible ways to improve the system performance may reside, for instance, in the use of an outlier rejection technique to handle misleading RSS measurements and/or in the use of advanced filtering techniques (which should give adaptive weights to every different measurement) in order to properly fuse together inertial and radio signals [43].

Chapter 3

Activity Classification

Nothing is more revealing than movement.

– Martha Graham (The American Dance)

In Chapter 2, it has been shown how inertial sensors can be effectively employed to estimate and track the arm posture of a person. However, we can also assume to move at a higher level of motion analysis abstraction and focus instead more generally on the behavior of that person, trying to recognize, still using inertial sensors, the activities he/she is performing. To this end, in this chapter, a novel BSN-based low-complexity *activity classification* algorithm is presented, where user activities are detected and recognized, choosing from a list of known activities, just by analyzing acceleration measurements obtained from a few body-worn inertial sensor devices (namely, accelerometers). The algorithm performance is experimentally evaluated and compared to that of existing algorithms using a publicly available dataset, thus providing a fair and unbiased benchmark for comparisons with other algorithms.

3.1 Introduction

Wireless sensor networks (WSNs) are attracting a relevant interest in many applications, typically associated with monitoring of particular environments. BSNs are a special class of WSNs, where wireless nodes are applied to a user body in order to monitor and detect some activities, e.g., activities of daily living (ADL), performed by the user. Relevant applications of these systems include long-term remote monitoring (e.g., at home) of the activities performed by a user (e.g., elderly people or post-rehabilitation patients), typically for medical purposes [44].

Past work on BSN activity classification algorithms has relied on accelerometers placed in multiple locations over the body [45, 46]. A performance improvement can be observed using multiple types of sensors [47, 48, 49, 50]. However, regardless of the considered type of sensor, an activity classification algorithm is generally composed of two phases: a *training* phase, typically used for calibration and parameters estimation purposes; and an *online* (classification) phase, possibly executed in real time. The training phase aims at identifying activity-specific features from the signals generated at each sensor, after manual [49] or automatic [51, 52, 53] signal segmentation. Regarding classification, most of the works in the literature tend to adopt thresholding or to use k-Nearest Neighbors (k-NN) algorithms, because of their simplicity and applicability on low-cost mobile devices [46, 50]. However, more sophisticated techniques have also been considered, such as those based on the use of decision trees [45] or hidden Markov models [48].

3.1.1 Chapter Contribution

In this chapter, we design a novel low-complexity BSN-based activity classification algorithm, which allows to detect and classify a sequence of activities, choosing from a list of known activities, by observing accelerometric data. A preliminary training phase allows to automatically optimize key parameters of the algorithm. The goal of the training phase is that of selecting a proper subset of nodes in order to minimize the number of relevant features, yet guaranteeing an accurate activity classification degree. The algorithm performance is analyzed using publicly available experimental

data [54] (in part generated in the context of the Opportunity Challenge [7, 8]), thus providing a valid and unbiased benchmark for comparisons with other algorithms. The proposed algorithm outperforms, especially when using a limited number of nodes, other known low-complexity algorithms, such as the k-NN, the Nearest Centroid Classifier (NCC), the Linear Discriminant Analysis (LDA), and the Quadratic Discriminant Analysis (QDA) [8, 36, 55]. The obtained results are very promising, making the proposed algorithm suitable for healthcare real-time monitoring applications.

Chapter Outline

The rest of this chapter is structured as follows. In Section 3.2, the experimental set-up and the performance metrics are preliminary introduced, followed by the derivation of the proposed algorithm. Section 3.3 is dedicated to performance analysis. Finally, in Section 3.4 concluding remarks are given.

3.2 Method

3.2.1 Experimental Set-up and Performance Metrics

As anticipated in Section 3.1, the experimental data used to test our algorithm are shared data collected in the context of the European project Opportunity and provided for the so-called Opportunity Challenge [54, 7, 8]. Figure 3.1 shows the experimental configuration of the sensor nodes in the considered BSN. The output of the BSN consists of mainly accelerometric data, integrated, for some nodes, with gyroscopic and magnetometric data—in this chapter, only accelerometric data will be used. For data collection, different users were asked to perform sequences of consecutive predefined movements and naturally executed daily activities. This allowed to generate highly realistic data upon which robust and flexible algorithms could be developed and trained.

Given a discrete set of predefined activities $\mathscr{A} = \{a_1, a_2, \ldots, a_A\}$ (with cardinality $|\mathscr{A}| = A$), the metric which will be used to evaluate the performance of the proposed

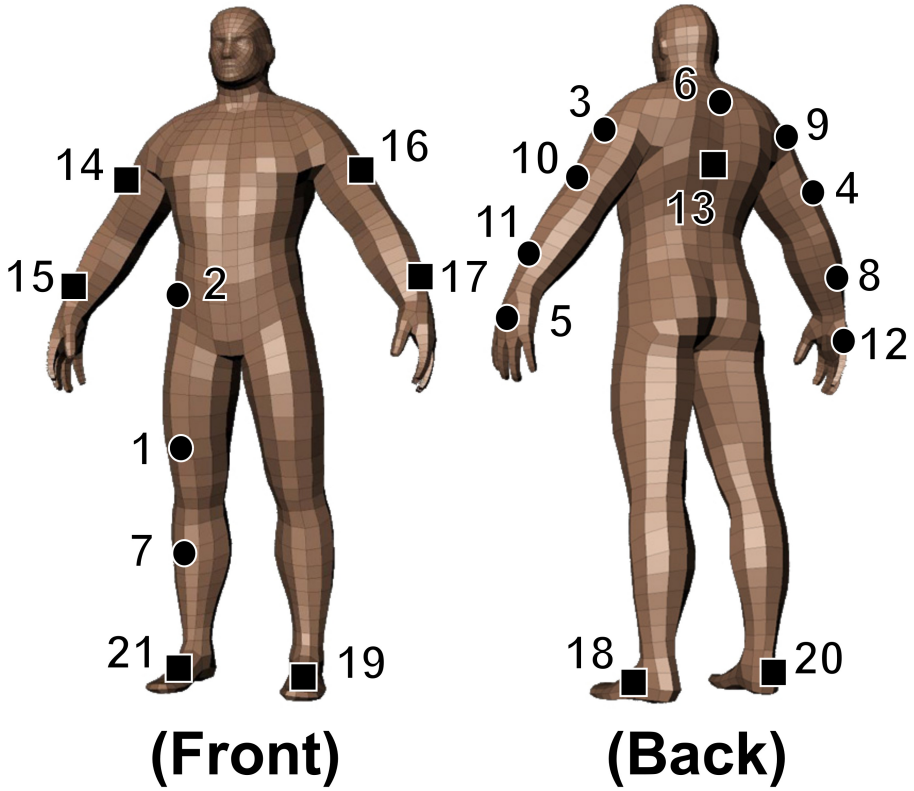


Figure 3.1: Opportunity Challenge setup [7, 8]. The position of accelerometers (●) and inertial measurement units (■) is highlighted.

classification algorithm is the weighted f1 score [8], denoted as $F1^{\mathrm{w}}$ and evaluated as follows:

$$\begin{aligned} F1^{\mathrm{w}} &= \sum_{a=1}^{A} 2 * \left(\frac{n_a}{N}\right) * F1_a \\ &= \sum_{a=1}^{A} 2 * \left(\frac{n_a}{N}\right) * \left(\frac{prec_a * rec_a}{prec_a + rec_a}\right) \end{aligned}$$

where: a is the considered activity; n_a is the number of samples of the inertial data sequence in correspondence to which a user is performing activity a; N is the total number of samples of the collected inertial data sequence; $F1_a$ is the f1 score computed for activity a; and $prec_a$ and rec_a are, respectively, the precision (defined as $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, where TP and FP are the numbers of true positives and false positives) and the recall (defined as $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, where FN is the number of false negatives) evaluated for activity a. In practical, $F1^{\mathrm{w}}$ is the average f1 score over all activities. For reproducibility purposes, the performance of the proposed algorithm is evaluated for multiple datasets collected from different subjects. To this end, in the Opportunity dataset, for each subject, golden labels are provided, containing a single stream of (manually) labeled activities.[1] In the proposed algorithm, which is described in the following subsections, we will only make use of accelerometric data.

3.2.2 Algorithm Description

Generally, an activity classification problem involves the design of an algorithm that can estimate the sequence of occurrences of specific activities choosing from a discrete set of predefined activities $\mathscr{A}$, directly working on (typically) inertial signals (e.g., accelerometric and gyroscopic signals). The proposed algorithm tries to concentrate most of its complexity in a (offline) *training* phase, which is performed once and aims at selecting the smallest subset of nodes to extract the smallest, yet sufficient, number of accelerometric time features to guarantee a good performance (high

[1] For the sake of generality, the samples containing "undefined" activities (e.g., transitions between classifiable activities), because of the temporal continuity of the collected data, are not taken into account for the evaluation of $F1^{\mathrm{w}}$.

$F1^{\mathrm{w}}$). Then, upon optimized setting of proper thresholds, the *online* phase is relatively light (in terms of time complexity), making real-time activity classification applications feasible.

In the following, a detailed description of the operational steps of the proposed algorithm are presented. After preliminaries on data preprocessing, the online and training phases are presented. In order to run properly, the online activity classification algorithm needs some parameters that have to be estimated and optimized during the training phase. Even though, practically, the training phase precedes the online phase, in the remainder of this subsection, after preliminaries on data processing and feature extraction, we first describe in detail, for ease of presentation, the online phase. In the training phase, the same steps of the online phase are considered, with the only difference that known (labeled) data are used to tune the key parameters of the algorithm, which are then kept constant in the (following) online phase.

Preliminaries on Data Preprocessing and Feature Extraction

At each node, an accelerometer outputs a stream of three-dimensional data, which corresponds to the acceleration measured by the sensor in its three reference axes. More formally, let us define the three-dimensional acceleration vector, measured at the i-th epoch, as

$$\boldsymbol{\alpha_i} = (\alpha_{xi}, \alpha_{yi}, \alpha_{zi}) \qquad i \in \{1, 2, \ldots, N\}$$

and the norm of $\boldsymbol{\alpha_i}$ as

$$\overline{\alpha}_i = |\boldsymbol{\alpha_i}| = \sqrt{\alpha_{xi}^2 + \alpha_{yi}^2 + \alpha_{zi}^2} \quad i \in \{1, 2, \ldots, N\}$$

where N is the number of samples in the stream. For the sake of simplicity, let us assume that the accelerometer is already calibrated and, thus, α_{xi}, α_{yi}, and α_{zi} are expressed in g units. Furthermore, the accelerometric data are low-pass filtered in order to deal with smoother data in both training and online phases.

Starting from this signal, at the i-th epoch, two types of simple features are of interest and can be extracted. The first one, denoted as *p-feature* (where "p" stands for "parallel") and indicated with $acc_i^{(\mathrm{p})}$, is a properly chosen component of the normalized (to unity) acceleration vector, i.e., $acc_i^{(\mathrm{p})} \in \{\alpha_{xi}/|\boldsymbol{\alpha_i}|, \alpha_{yi}/|\boldsymbol{\alpha_i}|, \alpha_{zi}/|\boldsymbol{\alpha_i}|\}$. In

particular, the chosen component is the one parallel to the "main" axis of the related body segment (e.g., considering Figure 3.1: for node 1, $acc_i^{(\mathrm{p})}$ is the acceleration value measured along the femur direction; for node 7, the one measured along the tibia). Observe that, due to the normalization, $acc_i^{(\mathrm{p})}$ can only assume real values in $[-1,+1]$.

The second considered feature, denoted as *dev-feature* and indicated with $\sigma_i(s)$, is the standard deviation of the norm of the acceleration computed within a sliding window (with fixed length $L = 2 \cdot s + 1$ and centered at the i-th sample[2]) that runs over all the acceleration samples. Neglecting border effects (the extension is straightforward, as shown in [56]), $\sigma_i(s)$ can be expressed as follows:

$$\sigma_i(s) = \sqrt{\frac{\sum_{k=i-s}^{i+s} [\overline{\alpha}_k - \mu_k(s)]^2}{L-1}} \tag{3.1}$$

where

$$\mu_k(s) \triangleq \frac{\sum_{k=i-s}^{i+s} \overline{\alpha}_k}{L-1}. \tag{3.2}$$

It can be observed that $\sigma_i(s)$ is always larger than or equal to 0—typically, it is not considerably larger than 1 (due to the expression of the accelerometric data in g units).

The General Idea

Considering a single activity $a \in \mathscr{A}$, the output of an activity classification algorithm is given by a binary sequence $\boldsymbol{w}(a) = (w_1(a), w_2(a), \ldots, w_N(a))$ where $w_i(a) = 0$ if a is not detected at epoch i (the index i runs over the samples of the collected accelerometric data sequence) and $w_i(a) = 1$ if a is detected. In particular, $\boldsymbol{w}(a)$ contains "activity windows" (disjoint groups of consecutive "1"s), within which the activity a has been detected. The length of each activity window is recursively determined—using three parameters ℓ_1, ℓ_2, and ℓ_3, optimized independently for each activity. This process will be described later.

[2] More details about the definition of the window, especially at the borders of the acceleration signal, are given in [56].

For each considered activity, the proposed activity classification algorithm aims at automatically selecting the best nodes in the BSN and the corresponding most significant feature types, that can best discriminate the occurrence of the considered activity (e.g., the thigh node is intuitively one of the best nodes to estimate a "sit" activity, whereas the feet nodes give relevant information about the "walk" activity). More formally, let us define as $\mathscr{F} = \{f_1, f_2, \ldots, f_F\}$ (with cardinality $|\mathscr{F}| = F$) a set of features where the generic feature $f \in \mathscr{F}$ corresponds to a feature "type" (p-feature or dev-feature) associated to a specific node. As an example, referring to Figure 3.1, $f = 1$p is the p-feature extracted from the thigh node (i.e., node 1) and $f = 7$d is the dev-feature extracted from the tibia node (i.e., node 7). In the following, when referring to the value of a feature, we will implicitly refer to the value of the "embedded" feature type. In particular, given a specific activity a, the combination $\mathscr{C}_a^* \subset \mathscr{F}$ represents the subset of the most significant features used to classify activity a.[3]

The proposed algorithm identifies activities by properly thresholding the selected features. In particular, referring to a given feature f, an activity a is considered as detected at the samples in correspondence to which the feature (type) is between the (lower and upper) thresholds $t_1(a,f)$ and $t_2(a,f)$, which are specifically derived and optimized independently for each activity a and for each feature f. The optimal values $\{t_1^*(a,f)\}_{a\in\mathscr{A},f\in\mathscr{C}_a^*}$ and $\{t_2^*(a,f)\}_{a\in\mathscr{A},f\in\mathscr{C}_a^*}$ are determined in the training phase, as described later.

In order for the activity classification algorithm to output a single sequence $\boldsymbol{w} = (w_1, w_2, \ldots, w_N)$ (where $w_i = 0$ if no activities are detected and $w_i = a$ if activity a is detected, i.e., $w_i(a) = 1$), since $w_i(a_j) = w_i(a_k) = 1$ may happen for some $i \in \{1,2,\ldots,N\}$ and for some $j \neq k$, with $j,k \in \{1,2,\ldots,A\}$, different priorities need to be assigned to the activities. The priorities are denoted as $\boldsymbol{q} = (q(a_1), q(a_2), \ldots, q(a_A))$, where each element (which indicates the priority of each considered activity) can as-

[3]Note that, throughout this chapter, the term "optimal" and the superscript "*" are equivalently used with reference to the tunable parameters considered in the proposed algorithm. Furthermore, if not stated otherwise, the optimality is always intended in terms of classification performance through the f1 score (when activities are independently considered) and the weighted f1 score (when activities are combined together).

sume real values in $[0,1]$. The priorities' list $\boldsymbol{q}$ must be interpreted as follows: if $q(a_i) > q(a_j)$, with $i, j \in \{1,2,\ldots,A\}$, activity a_i has a higher priority than activity a_j. The priority assigned to an activity is strongly related to the confidence in correctly detecting that activity and is based on the evaluation of specific performance metrics, which are evaluated in the training phase, as will be described later. In this way, the output sequence $\boldsymbol{w}$ is an $(A+1)$-ary sequence of activity labels, where the label of the activity with highest priority is selected in the cases where more than one activity is detected at the same epoch. The optimal priorities' list $\boldsymbol{q}^*$ is determined in the training phase, as described later.

Activity Classification

The online phase of the proposed algorithm is based on three main steps: (i) a first *coarse classification* step; (ii) a *refinement* step; (iii) and a final *priority-based activity combination* step.

Concerning the first coarse classification step, which is executed independently for each activity to be classified, a specific activity a is detected at epoch i (i.e., $w_i(a) = 1$) if *all* the features belonging to $\mathscr{C}_a^*$ are comprised between the corresponding optimal thresholds $\{t_1^*(a,f)\}_{f\in\mathscr{C}_a^*}$ and $\{t_2^*(a,f)\}_{f\in\mathscr{C}_a^*}$.

In the first coarse classification step, a single occurrence of an activity can be missed (because of little pauses or random movements). This can be avoided, or at least mitigated, by applying to the detected activity windows a refinement step which takes into account the length of the estimated activity windows. More specifically, given a specific activity a, in our implementation the refinement step is based on the following sequential operations: (1) every "null window" (a group of consecutive "0"s), with a length (in terms of its number of samples) shorter than ℓ_1^*, is switched to an activity window (and incorporated in the preceding and following activity windows, which are then fused together); (2) every activity window with a length shorter than ℓ_2^* is turned into a null window; (3) step 1 is repeated considering now every null window with a length shorter than ℓ_3^*.

Finally, the priority-based activity combination step consists in combining the sequences $\boldsymbol{w}(a_1)$, $\boldsymbol{w}(a_2)$, ..., and $\boldsymbol{w}(a_A)$ into a single sequence $\boldsymbol{w}$ of activity labels,

on the basis of the optimal priorities $\boldsymbol{q}^*$ assigned to the activities.

Algorithm Training

In order to work effectively, the proposed algorithm needs to be properly trained by exploiting the part of collected data for which the occurrences of the activity of interest are correctly (manually) labeled. The training phase aims at estimating the optimal values of the key parameters that will be used in the online phase. In particular, *for each activity a* the following parameters need to be estimated: (i) the optimal combination $\mathscr{C}_a^*$, along with the corresponding optimal thresholds $\{t_1^*(a,f)\}_{f\in\mathscr{C}_a^*}$ and $\{t_2^*(a,f)\}_{f\in\mathscr{C}_a^*}$; (ii) the optimal thresholds ℓ_1^*, ℓ_2^*, and ℓ_3^*, used to refine the estimated activity windows; (iii) the optimal activities' priorities list $\boldsymbol{q}^*$, *over all activities*.

A detailed flow diagram of the implementation steps of the training phase of the proposed algorithm is shown in Figures 3.2, 3.3, and 3.4 and will be now described, distinguishing between its three component blocks (the same of the online phase): (i) the first *coarse classification* step; (ii) the *refinement* step; and (iii) the final *priority-based activity combination* step.

At the beginning of the coarse classification step, shown in Figure 3.2 and executed for each activity a, the set $\mathscr{F} = \{f_1, f_2, \ldots, f_F\}$, whose elements are all the initial features, must be (manually) defined. Similarly, the set $\mathscr{S}$ (with cardinality $|\mathscr{S}| = S$) accounts for different ways (each way is identified by a partition $s = (\mathscr{S}_\mathrm{T}/\mathscr{S}_\mathrm{O})$) of separating the (training) dataset into the two subsets[4] $\mathscr{S}_\mathrm{T}$ and $\mathscr{S}_\mathrm{O}$ (s will be denoted as "separation"). Finally, $\mathscr{R}$ (with cardinality $|\mathscr{R}| = R$) accounts for possible values of confidence (in terms of percentages r_i of probability masses) considered to derive the thresholds $\{t_1(a,f)\}_{a\in\mathscr{A},f\in\mathscr{F}}$ and $\{t_2(a,f)\}_{a\in\mathscr{A},f\in\mathscr{F}}$.

The goal of the first part of the coarse classification step is to estimate and store the activity windows $\{\boldsymbol{w}(a,f,r,s)\}_{a\in\mathscr{A},f\in\mathscr{F},r\in\mathscr{R},s\in\mathscr{S}}$. As already explained in Subsection 3.2.2, the activity windows are derived, for a given activity a, by applying to each $f \in \mathscr{F}$ (considering the $\mathscr{S}_\mathrm{O}$ dataset associated to the selected s) the (prop-

[4]Note that the separation into the two subsets $\mathscr{S}_\mathrm{T}$ and $\mathscr{S}_\mathrm{O}$ is considered in order to cross-validate the statistical performance of the algorithm, i.e., to assess how the algorithms will generalize to an independent dataset.

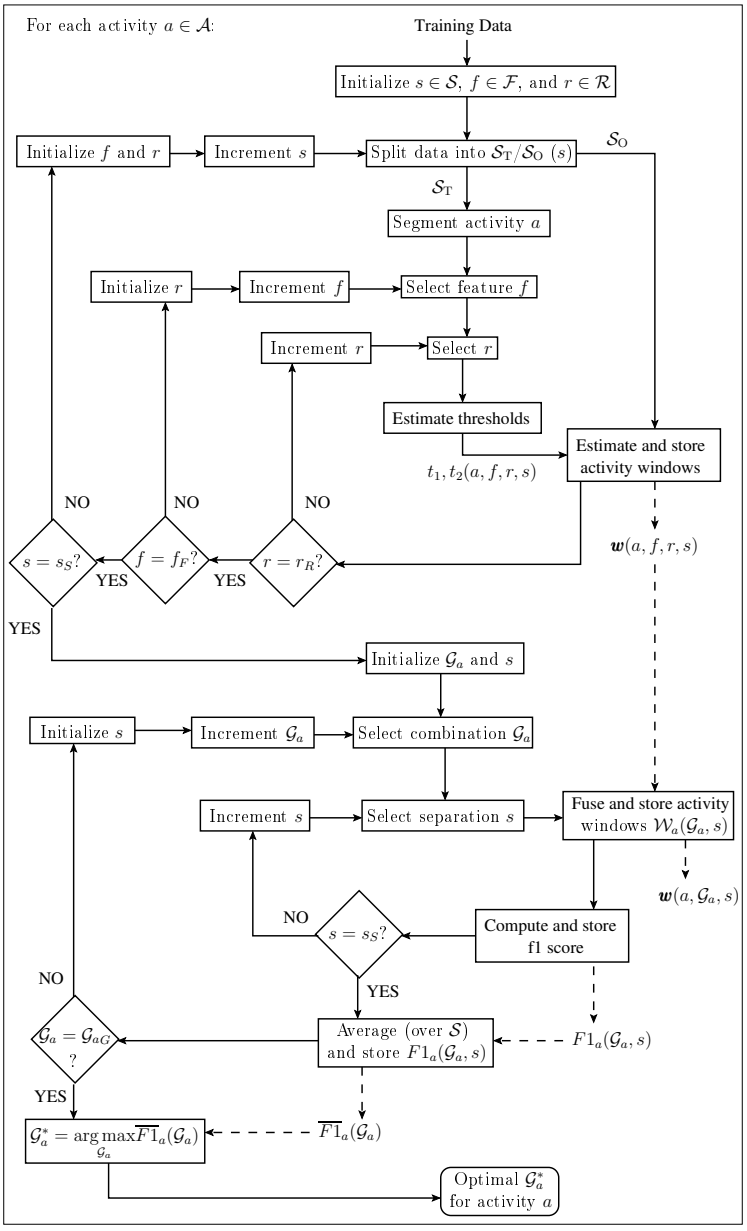


Figure 3.2: Detailed flow diagram of the implementation steps of the proposed algorithm's training phase: the first *coarse classification* step.

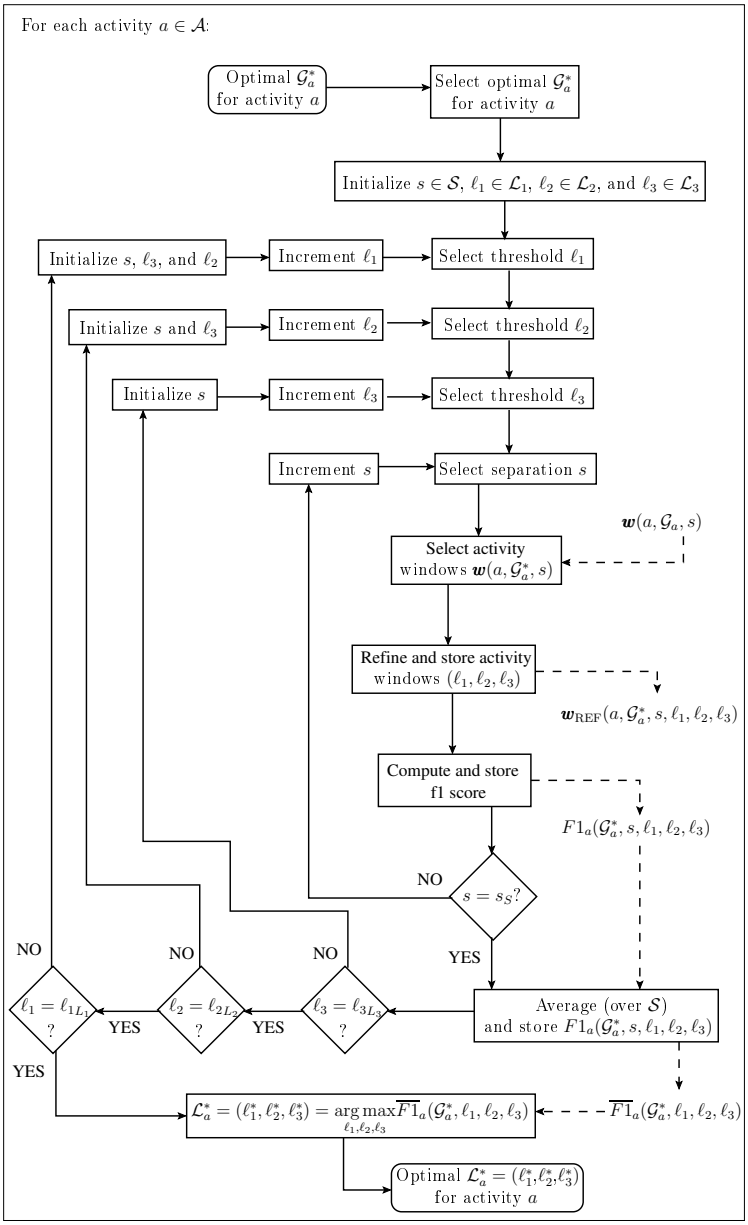


Figure 3.3: Detailed flow diagram of the implementation steps of the proposed algorithm's training phase: the *refinement* step.

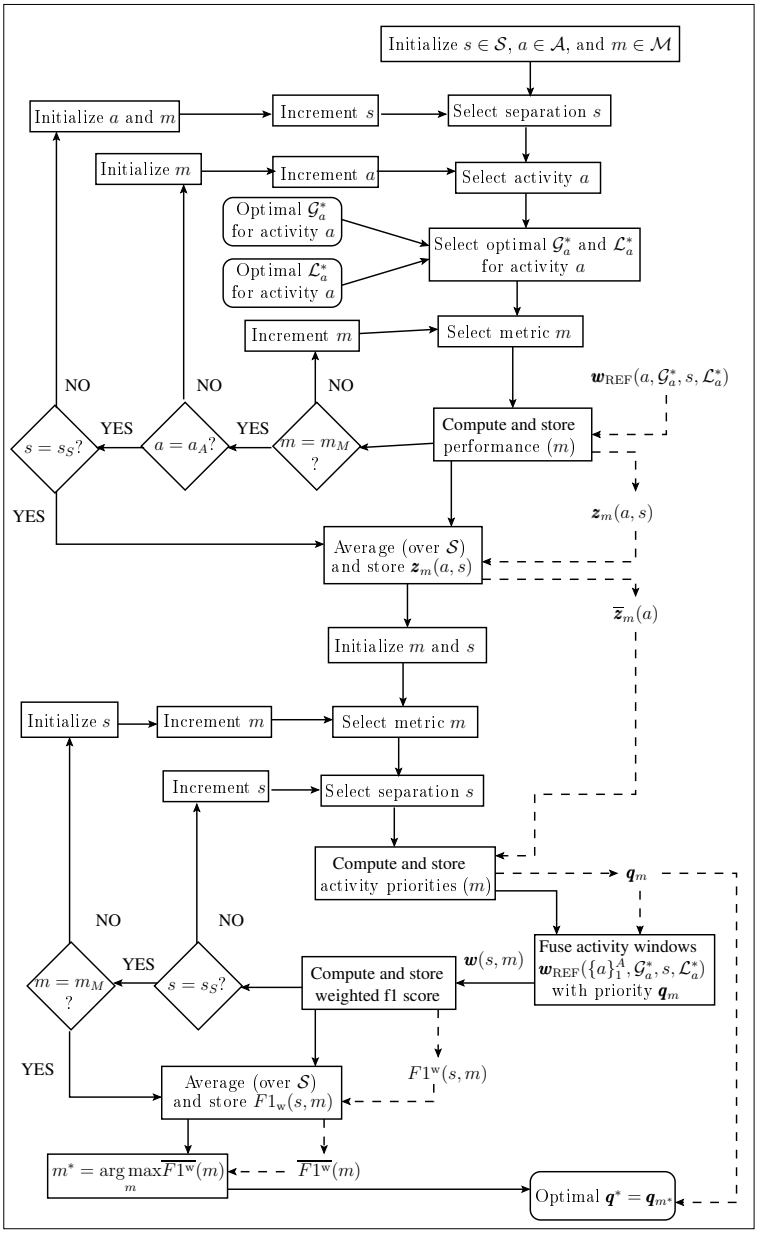


Figure 3.4: Detailed flow diagram of the implementation steps of the proposed algorithm's training phase: the final *priority-based activity combination* step.

erly estimated) thresholds $t_1(a,f,r,s)$ and $t_2(a,f,r,s)$—unlike in the online phase, t_1 and t_2 depend not only on a and f, but also on s and r. In particular, an illustrative description of the thresholds estimation step is shown in Figures 3.5 and 3.6 for an acceleration signal produced at the thigh node (i.e., node 1 in Figure 3.1): the p-feature (i.e., $f = 1\text{p}$) and the dev-feature (i.e., $f = 1\text{d}$), shown in Figure 3.5 (b), are first extracted from the acceleration signals (shown in Figure 3.5 (a), in the x, y, and z axes) and evaluated discretely in the "sit" intervals (assuming that $a = \text{sit}$ and for a given $s \in \mathscr{S}$) in order to obtain their PMFs in such intervals (the PMFs of the p-feature and the dev-feature are shown in Figures 3.6 (a) and 3.6 (b), respectively). Given a, f, s, for each value of $r \in \mathscr{R}$, the thresholds $t_1(a,f,r,s)$ and $t_2(a,f,r,s)$ are then chosen as the extremes of the shortest interval (t_1,t_2) which comprises at least $r\%$ of the probability mass.

The second part of the coarse classification step, still performed independently for each $a \in \mathscr{A}$, is based on the fusion of different combinations of the previously stored activity windows $\{\boldsymbol{w}(a,f,r,s)\}_{f\in\mathscr{F},r\in\mathscr{R},s\in\mathscr{S}}$. The goal is to find, for each activity, the optimal combination $\mathscr{G}_a^*$ (among the G possible combinations), where the generic combination $\mathscr{G}_a$ is associated with a possible subset of features $\mathscr{C}_a \subset \mathscr{F}$ and two values of r, i.e., r_p and r_dev, from which the thresholds $t_1(a,f,r,s)$ and $t_2(a,f,r,s)$ can be estimated when the considered f is, respectively, a p-feature or a dev-feature. As an example, an instance of $\mathscr{G}_a$ could be $\left(\mathscr{C}_a = \{1\text{d},1\text{p},4\text{p}\}, r_\text{p} = 93\%, r_\text{dev} = 97\%\right)$.

Given a combination $\mathscr{G}_a$ and a separation $s \in \mathscr{S}$, the set $\mathscr{W}_a(\mathscr{G}_a,s)$ is defined as the subset of $\{\boldsymbol{w}(a,f,r,s)\}_{f\in\mathscr{F},r\in\mathscr{R}}$ with $(f,r) \in \mathscr{G}_a$. The sequences of activity windows in $\mathscr{W}_a(\mathscr{G}_a,s)$ are then fused together (and stored for future use) into a single sequence of activity windows $\boldsymbol{w}(a,\mathscr{G}_a,s)$, obtained from the "logical AND" of all the considered sequences of activity windows, and its f1 score, denoted as $F1_a(\mathscr{G}_a,s)$, is computed and stored. By averaging $\{F1_a(\mathscr{G}_a,s)\}_{s\in\mathscr{S}}$ over all possible separations in $\mathscr{S}$ and denoting this average as $\overline{F1}_a(\mathscr{G}_a) = \sum_{s\in\mathscr{S}} F1_a(\mathscr{G}_a,s)/S$, the optimal $\mathscr{G}_a^*$ corresponds to the combination $\mathscr{G}_a$ which maximizes $\overline{F1}_a(\mathscr{G}_a)$.

The following refinement step, shown in Figure 3.3 and executed for each activity a, retraces the operations of the previous step in order to estimate the optimal thresholds $\mathscr{L}_a^* = (\ell_1^*,\ell_2^*,\ell_3^*)$ used to refine the previously estimated activity windows,

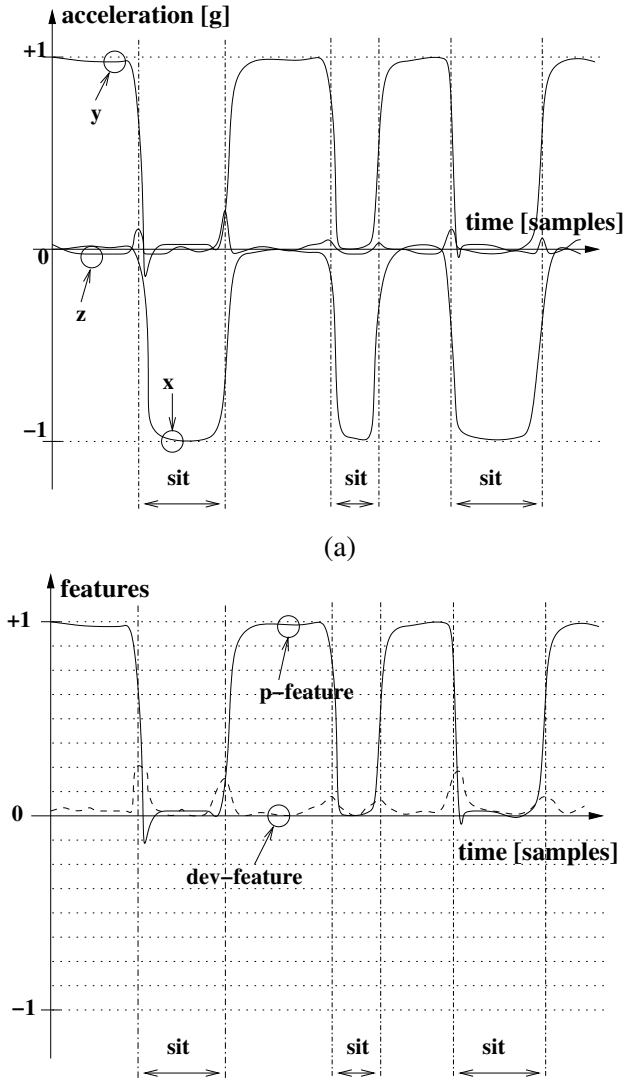
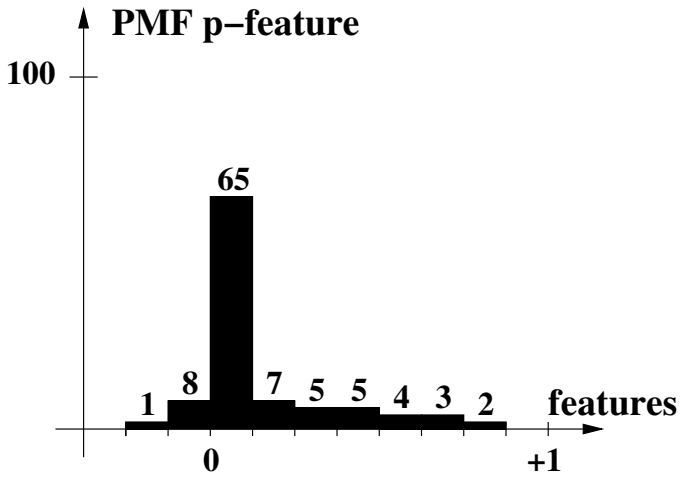
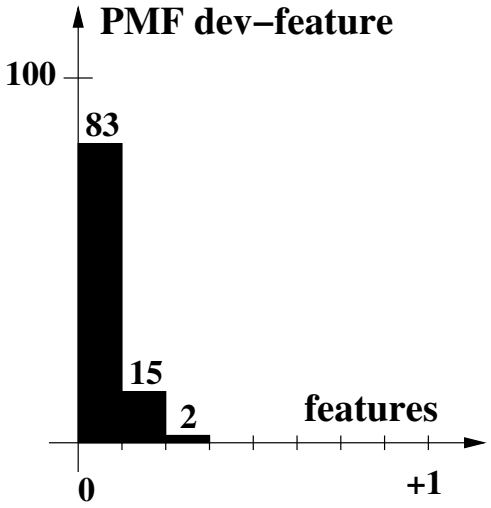


Figure 3.5: Description of the thresholds estimation step, which appears in Figure 3.2, for an illustrative acceleration signal produced at the thigh node: (a) acceleration signal (the x, y, and z components are highlighted); (b) p-feature (straight line) and dev-feature (dashed line) extracted from the previous acceleration signal.



(a)



(b)

Figure 3.6: Description of the thresholds estimation step, which appears in Figure 3.2, for an illustrative acceleration signal produced at the thigh node: PMFs of (a) p-feature and (b) dev-feature evaluated in the "sit" intervals. The features are extracted from the acceleration signal as shown in Figure 3.5.

but considering now just the optimal combination $\mathscr{G}_a^*$. To this end, three sets $\mathscr{L}_1$, $\mathscr{L}_2$, $\mathscr{L}_3$ of natural values are defined.[5] Given a separation $s \in \mathscr{S}$, all the possible combinations of $\ell_1 \in \mathscr{L}_1$, $\ell_2 \in \mathscr{L}_2$, and $\ell_3 \in \mathscr{L}_3$ are used to refine the previously estimated sequence of activity windows $\boldsymbol{w}(a,\mathscr{G}_a^*,s)$ as already described in Subsection 3.2.2. For every considered combination of $\ell_1 \in \mathscr{L}_1$, $\ell_2 \in \mathscr{L}_2$, and $\ell_3 \in \mathscr{L}_3$, the refined sequences of activity windows $\{\boldsymbol{w}_{\text{REF}}(a,\mathscr{G}_a^*,s,\ell_1,\ell_2,\ell_3)\}_{s\in\mathscr{S}}$ are then stored (for future use) and their f1 scores $\{F1_a(\mathscr{G}_a^*,s,\ell_1,\ell_2,\ell_3)\}_{s\in\mathscr{S}}$ are computed and stored. By averaging $\{F1_a(\mathscr{G}_a^*,s,\ell_1,\ell_2,\ell_3)\}_{s\in\mathscr{S}}$ over all possible separations in $\mathscr{S}$ and denoting this average as $\overline{F1}_a(\mathscr{G}_a^*,\ell_1,\ell_2,\ell_3) = \sum_{s\in\mathscr{S}} F1_a(\mathscr{G}_a^*,s,\ell_1,\ell_2,\ell_3)/S$, the optimal configurations $\mathscr{L}_a^* = (\ell_1^*,\ell_2^*,\ell_3^*)$ is estimated as the combination of ℓ_1, ℓ_2, and ℓ_3 which maximizes $\overline{F1}_a(\mathscr{G}_a^*,\ell_1,\ell_2,\ell_3)$.

During the final step, i.e., the priority-based activity combination, shown in Figure 3.4, the optimal list of activities' priorities $\boldsymbol{q}^*$ is estimated. To this end, a set of performance metrics $\mathscr{M} = \{m_1,m_2,\ldots,m_M\}$ (where $|\mathscr{M}| = M$) is considered. Specifically, in this chapter we consider the following performance metrics: (i) f1 score, (ii) precision, (iii) recall, (iv) specificity, and (v) accuracy. First, given an activity a and a separation s, every performance metric $m \in \mathscr{M}$ is used to evaluate the performance associated with the sequence of activity windows $\boldsymbol{w}_{\text{REF}}(a,\mathscr{G}_a^*,s,\mathscr{L}_a^*)$ and the corresponding value $\boldsymbol{z}_m(a,s)$ is stored. For given activity a and metric m, the obtained values $\{\boldsymbol{z}_m(a,s)\}_{s\in\mathscr{S}}$ are then properly averaged over all the possible separations $s \in \mathscr{S}$ and the resulting $\bar{\boldsymbol{z}}_m(a) = \sum_{s\in\mathscr{S}} \boldsymbol{z}_m(a,s)/S$ is used to compose (and store) a priority-based list of activities $\boldsymbol{q}_m = (q_m(a_1),q_m(a_2),\ldots,q_m(a_A)) \triangleq (\bar{z}_m(a_1),\bar{z}_m(a_2),\ldots,\bar{z}_m(a_A))$. Specifically, the priorities' lists $\{\boldsymbol{q}_m\}_{m\in\mathscr{M}}$ defines the priorities assigned to every activity according to the specific performance metric m.

At this point, for given $s \in \mathscr{S}$ and $m \in \mathscr{M}$, the A sequences of activity windows $\{\boldsymbol{w}_{\text{REF}}(a,\mathscr{G}_a^*,s,\mathscr{L}_a^*)\}_{a\in\mathscr{A}}$ are combined together, on the basis of the priorities' list $\boldsymbol{q}_m$, resulting in a single sequence of activity windows $\boldsymbol{w}(s,m)$ and the corresponding weighted f1 score, denoted as $F1^{\text{w}}(s,m)$, is computed and stored. By averaging $\{F1^{\text{w}}(s,m)\}_{s\in\mathscr{S}}$ over all possible separations in $\mathscr{S}$ and denoting this average as

[5] These values should be properly chosen in the order of at most a few seconds in order to filter out just the windows of samples which correspond to little pauses or random movements.

$\overline{F1^{\mathrm{w}}}(m) = \sum_{s \in \mathscr{S}} F1^{\mathrm{w}}(s,m)/S$, the optimal $\boldsymbol{q}^* = (q^*(a_1), q^*(a_2), \ldots, q^*(a_A))$ is then estimated as the $\boldsymbol{q}_m$ whose metric m maximizes $\overline{F1^{\mathrm{w}}}(m)$.

At the end of the training phase, for each activity a, the optimal G_a^* is used to further estimate the optimal thresholds $\{t_1^*(a,f)\}_{f \in C_a^*}$ and $\{t_2^*(a,f)\}_{f \in C_a^*}$ which will be actually used in the online phase, with the only difference that the entire training dataset is now considered to derive them. Therefore, for each activity a, the following optimal parameters have to be stored for future use in the online phase: C_a^*; $\{t_1^*(a,f)\}_{f \in C_a^*}$ and $\{t_2^*(a,f)\}_{f \in C_a^*}$; $\mathscr{L}_a^* = (\ell_1^*, \ell_2^*, \ell_3^*)$. Finally, the optimal list of priorities $\boldsymbol{q}^* = (q^*(a_1), q^*(a_2), \ldots, q^*(a_A))$, over all possible activities, is also stored.

3.3 Results and Discussion

The performance of the proposed algorithm has been evaluated for the classification of activities related to states of locomotion of the user. In particular, $A = 4$ activities are considered: *stand*, *walk*, *sit*, and *lie* (i.e., $\mathscr{A} = \{\text{stand}, \text{walk}, \text{sit}, \text{lie}\}$). In Subsection 3.3.1, we outline the considered configurations of nodes and features, for the proposed algorithm and the following existing classification algorithms (as anticipated in Section 3.1): the k-NN (with $k = 1$ and $k = 3$), the NCC, the LDA, and the QDA—more details about their practical implementation are given in [8, 36, 55]. All the algorithms (for all possible configurations) are tested on the same dataset, which, for the purpose of repeatability, takes into account four different subjects [54]. In all cases, $\mathscr{R} = \{90, 91, 92, \ldots, 100\}$, $\mathscr{L}_1 = \{0, 1, 2, \ldots, 60\}$, $\mathscr{L}_2 = \{0, 5, 10, \ldots, 200\}$, $\mathscr{L}_3 = \{0, 5, 10, \ldots, 200\}$, $\mathscr{M} = \{\text{f1 score}, \text{precision}, \text{recall}, \text{specificity}, \text{accuracy}\}$, and $S = 3$ different separations of the training dataset are considered. The obtained classification performance results are presented in Subsection 3.3.2. In Subsection 3.3.3, the time complexity of all considered algorithms is summarized. Subsection 3.3.4 investigates the robustness of the proposed algorithm against rotational noise.

3.3.1 Configurations of Nodes and Features

Overall, we consider 38 different configurations of nodes (and feature types): configurations 1-31 (with at most 7 nodes) apply to the proposed algorithms and the four

considered existing classification algorithms (k-NN, NCC, LDA, and QDA) and rely on the use of accelerometric data; configurations 32-38 (with more than 7 nodes) apply only to the four considered existing classification algorithms and rely on the use of other (besides accelerometers) inertial sensors (e.g, magnetometers). Each configuration involves a specific nodes' configuration, explicitly shown in the x axis of Figure 3.7 (which will be described in the next subsection) with reference to the node numbers in Figure 3.1. The choice of the nodes' feature types for each classification algorithm can be summarized as follows.

- **Configurations 1-31; proposed algorithm**. Only p-features and dev-features are considered. Specifically: for nodes 7, 1, 2, and 6 (namely, the nodes of the main vertical segment of the human body) both types of features are extracted; for nodes 13, 19, and 21 (feet nodes and a second node for the back) only the d-feature is extracted. In particular, at most $F = 11$ features are considered (for instance, the set of features of Configuration 1 is $\mathscr{F} = \{1\text{p}, 2\text{p}, 6\text{p}, 7\text{p}, 1\text{d}, 2\text{d}, 6\text{d}, 7\text{d}, 13\text{d}, 19\text{d}, 21\text{d}\}$).

- **Configurations 1-31; k-NN, NCC, LDA, and QDA algorithms**. For all the nodes, the mean of every acceleration component within a sliding window is considered, leading to 3 features extracted at each node. Therefore, at most $F = 21$ features are considered.

- **Configurations 32-38; k-NN, NCC, LDA, and QDA algorithms**. For all the nodes, the mean of every acceleration component within a sliding window is still considered (as in the previous case). In addition, for configurations from 34 to 38, the mean within a sliding window is also computed for the 3 components of the gyroscope and magnetometer signals, or some combinations of them, as summarized in the following.

  - for configuration 34: every node supplied with gyroscope and magnetometer (i.e., nodes from 13 to 21) produces 6 additional features (3 from gyroscope and 3 from magnetometer);

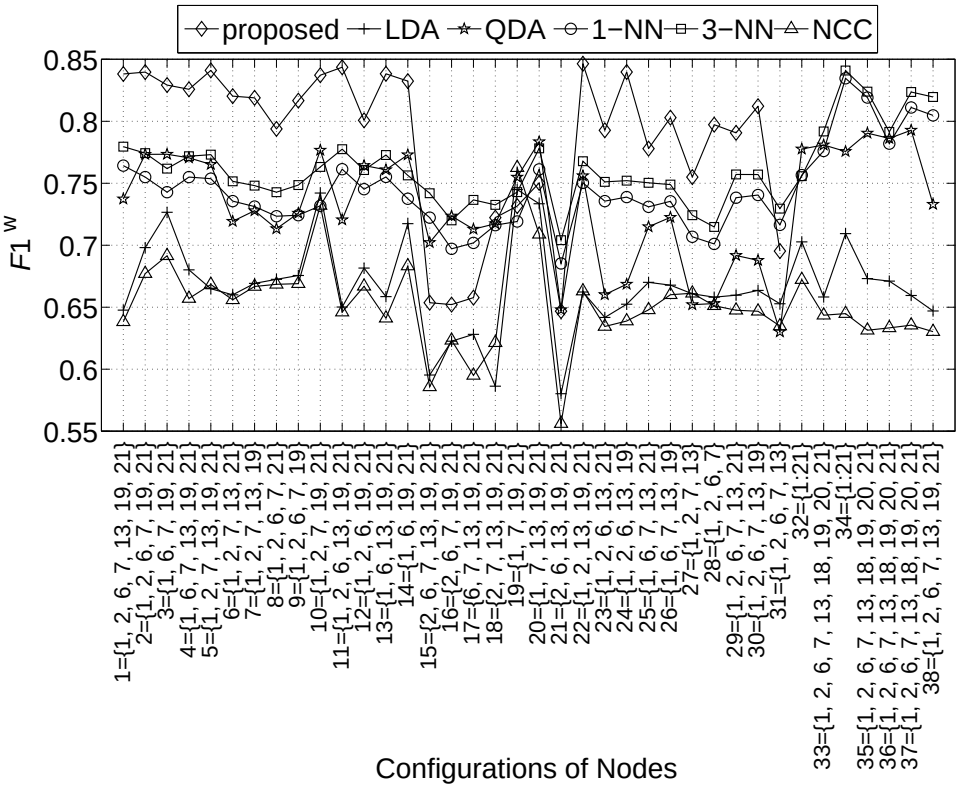


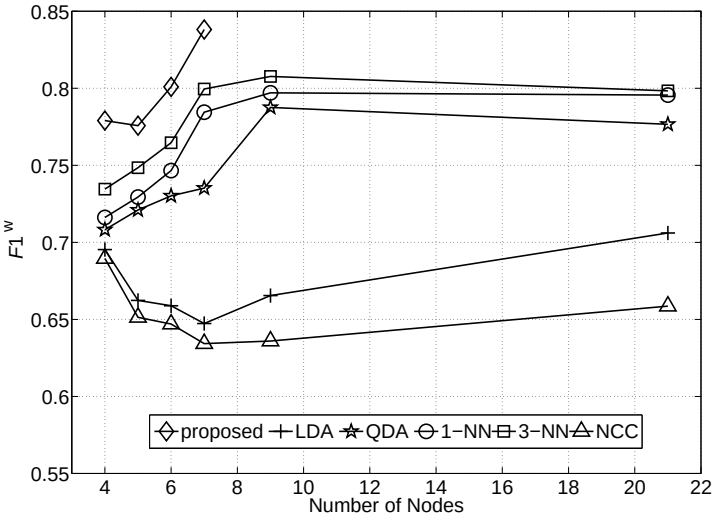
Figure 3.7: Average (over 4 subjects) classification performance (i.e., weighted f1 score) as a function of the considered configurations of nodes. The performance of the proposed algorithm is compared with that of some existing algorithms, averaging the performance of the four considered subjects. For every configuration, the considered subsets of BSN nodes (numbered as in Figure 3.1) are highlighted. The features per configuration and per algorithm are properly selected as summarized in Subsection 3.3.1.

- for configuration 35: 6 additional features (3 from gyroscope and 3 from magnetometer) are extracted at node 13;
- for configuration 36: 3 additional gyroscopic features are extracted at node 13;
- for configurations 37 and 38: 3 additional magnetometric features are extracted at node 13.

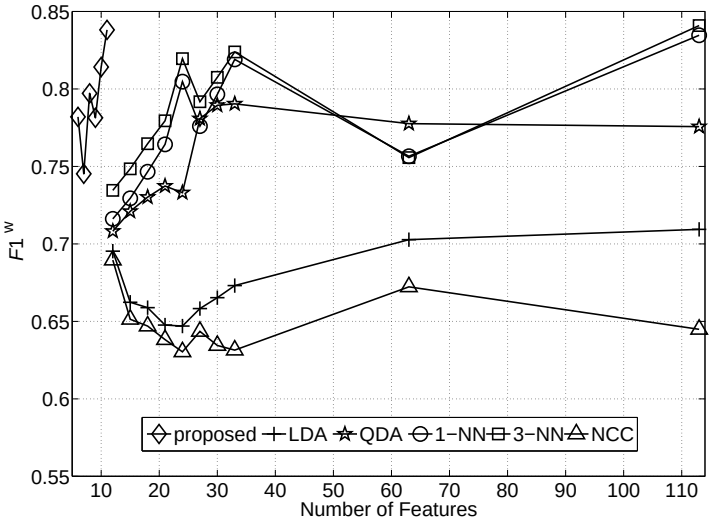
3.3.2 Classification Performance

The performance of the proposed algorithm has been evaluated for 31 different configurations of nodes in order to evaluate the optimal subset of BSN nodes (and, thus, of features) useful for the activity classification. In particular, in Figure 3.7, the performance of our algorithm (in terms of $F1^{\mathrm{w}}$) is shown as a function of the configurations of nodes, averaging over the considered four subjects. A comparison with the performance obtained with the other considered algorithms, run with the same configurations of nodes, is also shown. Note that the performance of these algorithms is also evaluated for more complex configurations of nodes (for a total of 38 configurations), which consider the use of a larger number of nodes and features. It is easy to observe that, in most of the considered configurations of nodes, our algorithm outperforms the other ones. It can also be seen that some of the benchmarking algorithms (in particular, k-NN and QDA) have similar performance, but only when considering, in their cases, more complex configurations with large numbers of nodes and features. Note also that the central "hole" in Figure 3.1 is associated with those configurations of nodes (namely, 15, 16, 17, 18, and 21 in Figure 3.7) which do not use node 1 in Figure 3.1 (the thigh node), i.e., a key node in discriminating between sit and stand activities. A similar observation can be made for configuration 31 (as denoted in Figure 3.7), which hardly discriminates between stand and walk due to the absence of both feet nodes (i.e., nodes 19 and 21 in Figure 3.1).

In order to better investigate the impact of the number of nodes on the performance of the considered algorithms, in Figure 3.8 (a) the performance of the considered algorithms is properly averaged over all configurations with the same number of



(a)



(b)

Figure 3.8: Average (over 4 subjects) classification performance (i.e., weighted f1 score) as a function of the considered (a) number of nodes and (b) number of features. For each considered algorithm, the configurations which use the same number of (a) nodes or (b) features have been averaged together.

nodes. It can be observed that, for a given number of nodes in the BSN, our algorithm, making use of configurations with at most 7 nodes, outperforms the others. Moreover, with only 7 nodes, our algorithm outperforms all existing algorithms, including the k-NN and QDA, run with a much larger number of nodes (e.g., 21).

Another aspect to take into account is the number of features used in the algorithms. Unlike the existing algorithms, where each node generates at least three features,[6] our algorithm is such that a maximum of two features (i.e., the p-feature and the dev-feature) can be extracted at each node. In Figure 3.8 (b), it is then shown how the algorithms' performance changes with respect to the considered overall number of features (over all nodes). It can be concluded that our algorithm provides the same, or even better, performance, with respect to the other algorithms, using a significantly smaller number of features. The number of used features has a significant impact on the time complexity of a classification algorithm, as it will be explained in Subsection 3.3.3, allowing a better real-time applicability of our algorithm with respect to the others. As previously observed, the k-NN and QDA are the only algorithms which can achieve, for a very large number of features (over 110), a performance similar to that of our algorithm (with only 11 features).

For the sake of completeness, we want to highlight that, if the previously cited configurations 15, 16, 17, 18, 21 in Figure 3.7 (which do not use node 1 in Figure 3.1) are not taken into account for the evaluation of the curves in Figure 3.8, the performance of our algorithm improves significantly more (in relative terms) than those of the other algorithms.

Finally, on the basis of the previous results, configuration 22 (as denoted in Figure 3.7) can be identified as the best configuration of nodes for our algorithm. In particular, it needs 5 nodes (namely, nodes {1, 2, 13, 19, 21} in Figure 3.1) and generates $F = 7$ features (5 dev-features, one per node, and 2 p-features, associated with node 1 and 2 in Figure 3.1). Using this configuration, our algorithm obtains a value

[6]The typical features computed at each node are the mean of every acceleration component within a sliding window. In addition, other six features are considered for nodes provided with gyroscopes and magnetometers. A standard deviation-related feature has been also used giving however poor performance.

Table 3.1: Time complexity of the online phase for the considered algorithms.

Algorithm	Time Complexity
proposed, NCC, LDA, and QDA	$O(F \cdot A)$
k-NN	$O(F \cdot D)$

of $F1^{\mathrm{w}}$ around 85%. The second best algorithm, i.e., the k-NN (with $k = 3$), reaches a value of $F1^{\mathrm{w}}$ around 77% using 15 features (more than twice the number in our algorithm) and more sensors (indeed, gyroscopes and magnetometers are available in nodes {13, 19, 21} in Figure 3.1). Furthermore, the best among the other algorithms (again, the k-NN with $k = 3$) obtains its highest $F1^{\mathrm{w}}$ score (namely, $F1^{\mathrm{w}} = 84\%$) using a significantly more complex configuration (namely, configuration 34 in Figure 3.7), which comprises a total of 21 nodes and $F = 113$ features.

3.3.3 Time Complexity

The time complexity of the proposed algorithm has been evaluated and compared with those of the classification algorithms previously considered for performance comparison.[7] In particular, it is possible to prove that the time complexity of the online phase of our algorithm is a linear function of the number of features F and the number of activities A. In Table 3.1, the time complexity of our algorithm is reported along with those of the other considered algorithms [36, 55]. It can be observed that the time complexity of all algorithms is a linear function of the number of features F. In addition, our algorithm, the NCC, LDA, and QDA algorithms have a complexity linearly dependent on the number of activities A, but independent of the training dataset size D. On the contrary, the time complexity of the k-NN algorithm depends linearly on D and is independent of A. Observing also that, typically, $A << D$, it can be concluded that the complexity of the online phase of our algorithm is similar to

[7] Recall that, we here consider the time complexity of the online phase due to its impact on real-time applicability of the algorithm and due to the fact that the training phase should be performed just once (offline), provided that the BSN configuration does not change over time.

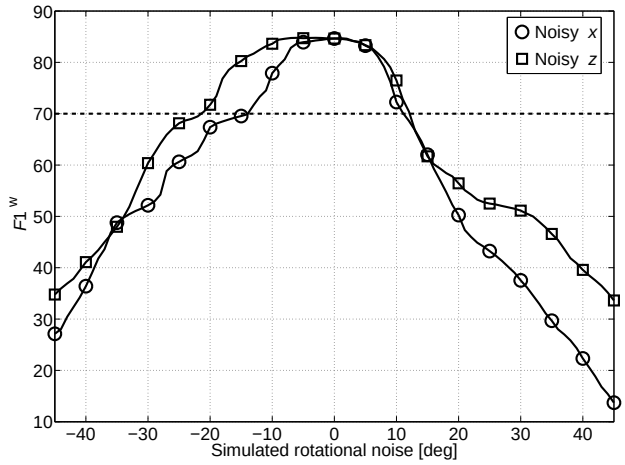
that of the NCC, LDA, and QDA algorithms and scales better than that of the k-NN algorithm. Moreover, having shown that our algorithm performance, when considering few features, is far better than that of the NCC, LDA, and QDA algorithms (and also slightly better than that of the k-NN algorithm), it can be concluded that our algorithm guarantees the best compromise between performance and time complexity.

3.3.4 Robustness to Noise of the Proposed Algorithm

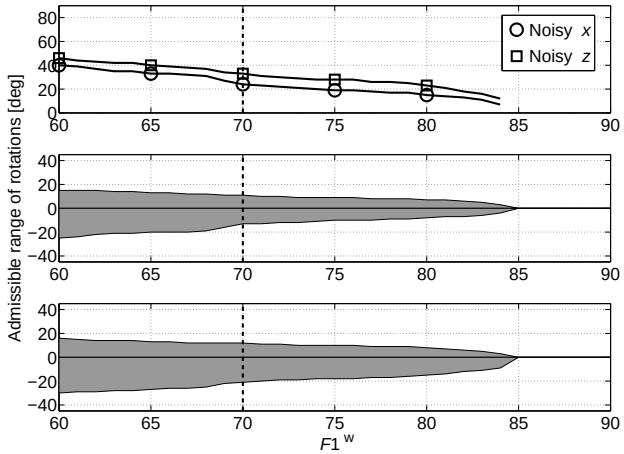
A typical problem of a real BSN scenario consists of unwanted rotations in the displacement of the BSN nodes, that can differ between the training and the online phases. It is then of interest to investigate the robustness of an algorithm to rotational noise. To this end, artificial rotational noise (dimension: [deg]) has been added to the accelerometric data, simulating possible rotations of a device around its three reference axes.

Two considerations can be preliminary made: the dev-feature is actually insensitive to rotational noise, due to its definition; the p-feature is invariant to rotational noise around the axis about which the feature is measured (i.e., the one parallel to the related body segment, which we assume to be the y axis for every device). For such reasons, the rotational noise is then being only added to p-features and the device rotations have been simulated only around its other two axes, namely (due to the previous assumption) the x and z axes. More specifically, from now on, we assume that the x axis is directed from the front to the back of the user, whereas the z axis is directed from his/her right side to his/her left side. For ease of simplicity, we only simulate rotations around one axis at a time.

In Figure 3.9 (a), the average performance of our algorithm (averaged over the 4 considered subjects) is shown as function of the intensity of the simulated rotational noise (in terms of degrees of rotation with respect to the initial orientation), for the previously estimated (at the end of Subsection 3.3.2) optimal configuration of nodes (i.e., configuration 22, as denoted in Figure 3.7). In Figure 3.9 (b), for the same configuration 22 (as denoted in Figure 3.7) and averaging over the same 4 subjects, the curves show which is the range of rotations that can be applied to the nodes in order to keep a user-defined admissible minimum performance (in terms of weighted f1



(a)



(b)

Figure 3.9: Average (over 4 subjects) classification performance (i.e., weighted f1 score) of the proposed algorithm in the presence of simulated rotational noise: (a) the weighted f1 score as a function of the intensity of the simulated rotational noise; (b) the admissible range of rotations as a function of the weighted f1 score. The previously estimated optimal configuration of nodes (i.e., configuration 22, as denoted in Figure 3.7) is considered. Indicative thresholds (dashed lines), corresponding to an admissible minimum performance of $F1^{\mathrm{w}} = 70\%$, are also shown in two subfigures.

score). The results in Figure 3.9 show that our algorithm, possibly due to the nature of the activities that we want to classify, suffers less from rotations around the z axis than those around the x axis. As an example, if one accepts as a minimum acceptable performance a $F1^{\mathrm{w}}$ equal to 70%, the system can tolerate rotations in the range of $[-10^{\mathrm{o}}, 10^{\mathrm{o}}]$.

We also remark that, due to the realistic data collection (often operated in different times with respect to the training acquisition), the testing data (used in the online phase) implicitly presents real rotational noise. Therefore, the results previously presented implicitly assume that the proposed algorithm has to combat some rotational noise.

3.4 Conclusions

In this chapter, a simple, yet effective, activity classification algorithm has been presented. Its performance has been evaluated and compared with that of existing algorithms. The data used to test the algorithms are publicly available and, thus, represent a valid and unbiased benchmark for the evaluation of the performance of different algorithms. The proposed algorithm is based on simple comparisons of properly selected features with thresholds that are automatically optimized during a preliminary training phase performed once (offline). In order to simplify the operations of the online phase of the algorithm, the training phase also allows to automatically select the optimal subset of nodes and features to be used.

The time complexity of the proposed algorithm has also been evaluated. Our results show that its complexity is on the order of that of existing algorithms, but its performance is better. On the other hand, some of the existing algorithms show a performance similar to that of ours at the cost of higher complexity. In particular, our algorithm significantly outperforms the others when using a small numbers of nodes and features. Taking also into account the robustness of the proposed algorithm against rotational noise, it can be concluded that it can be used effectively for real-time activity classification, especially when very strict constraints on the number of BSN nodes are introduced (e.g., activity classification applications for elderly or

motion-impaired people).

Chapter 4

Evaluation of Functional Motor Tasks

"Then you shall judge yourself," the king answered. "that is the most difficult thing of all.
It is much more difficult to judge oneself than to judge others.
If you succeed in judging yourself rightly, then you are indeed a man of true wisdom."

– Antoine de Saint-Exupéry (The Little Prince)

Chapters 2 and 3 presented two examples of how inertial sensors can be effectively exploited in multi-purpose human motion analysis. Provided that we have already shown that inertial sensors can be used to recognize the pose (Chapter 2) and the activities (Chapter 3) of a user, in this chapter we will instead move at a lower level of analysis and we will focus on raw inertial signal processing in order to *evaluate the motor performance of a user while performing an exercise*. Note that this is especially useful in clinical scenarios where it is likely to find patients who need to be continuously monitored while performing functional exercises (e.g., for rehabilitation purposes and/or therapy redefinition). In particular, this chapter focuses on the characterization, from a kinematic viewpoint, of the LA task, which is used to observe motion impairments in PD patients, relying on the use of a few inertial devices placed on the patient body.

4.1 Introduction

4.1.1 Motivation

PD is the second most common neurodegenerative disorder after Alzheimer's disease. According to the Global Declaration for Parkinson's Disease, 6.3 million people suffer from PD worldwide [57]. The prevalence of PD is about 0.3% of the whole population in industrialized countries, rising up to 1% over the age of 65 and to 4% over 80. The clinical picture of PD is characterized by a progressive deterioration of the motor performance, with the occurrence of slowness (bradykinesia) and poverty of voluntary movements, expressionless face, "resting" tremor, stooped posture, festinating gait, and axial instability. Asymmetry of motor symptoms is also a typical characteristic of PD. Although the symptoms can be improved by dopaminergic drugs, such as L-dopa, over time their effectiveness worsens and motor fluctuations may occur as well as dyskynesias and involuntary movements. Furthermore, variations in the severity of these symptoms are observed during dosing intervals.

The clinical picture assessed during an outpatient check-up in the medical office poorly represents the real (actual) clinical status especially in fluctuating patients. Indeed, repeated daily assessments of motor symptoms would be required and this is usually done by asking the patient to annotate the number of hours of OFF (i.e. when drugs are not effective) and ON (i.e. when they are effective) conditions, although this is not fully reliable due to perceptual bias. For this reason, in recent years a large number of studies on automatic systems to evaluate motor fluctuations of PD patients has been developed [58]. The most common approach is leveraging sensing technology to automatically evaluate the performance of specific motor tasks, such as "sit-to-stand" [59, 60], gait analysis [61, 62], and tremor [63]. The basic idea is to develop a system able to get an evaluation of the motor status of a patient as close as possible to the evaluation of neurologists when they apply semi-quantitative evaluation scales, such as the Unified Parkinson's Disease Rating Scale (UPDRS) [64, 65].

4.1.2 The Leg Agility Task

In particular, to the best of our knowledge, little attention [66, 67, 68] has been devoted, in the literature, to the evaluation of the LA task (which is an UPDRS item).

Task Description

The LA task aims at evaluating the severity of motion impairments of a PD patient, with specific focus on the lower limbs. In this exercise, the patient is asked to sit on a chair provided with rigid backrest and armrests. The patient must place both his/her feet on the floor in a comfortable position. The exercise consists in alternately raising up and stomping the feet on the ground, as high and as fast as possible. Ten repetitions per leg must be performed while sitting on the chair in order to test each leg separately. The examiner should first train the patient, showing him/her the correct execution of the exercise, stopping as soon as the patient starts. The significant parameters that have to be measured, independently for each leg, are the speed, the regularity, and the amplitude of the movement. Moreover, differences can be observed between the movements performed with the different legs. For this reason, in the following we will distinguish between Right LA (RLA) and Left LA (LLA) tasks.

UPDRS Evaluation

According to the guidelines of the Movement Disorder Society (MDS), the LA task must be evaluated observing the following parameters: amplitude, slowing, hesitations, interruptions, and freezing. In particular, in Table 4.1, an attempt at mapping these parameters with an UDPRS evaluation is presented. While the first feature (i.e., amplitude) directly corresponds to a physical measure, the quantitative evaluation of the other ones typically relies on the experience of neurologists. Therefore, inter-neurologist score variations cannot be a priori excluded.

Table 4.1: UPDRS mapping.

UPDRS	Amplitude	Slowing	Hesitations	Interruptions	Freezing
0	nearly constant	no	0	0	0
1	decrements near the end	slight	≥ 1	1,2	0
2	decrements midway	mild	-	3,4,5	0
3	decrements after first tap	moderate	-	≥ 6	≥ 1
4	always minimal or null	severe	-	always	-

4.1.3 Chapter Contribution

In this chapter, we focus on the characterization of the LA task in PD patients, devising an approach for quantitative evaluation of relevant kinematic characteristics. In particular, we first show that the LA task can be effectively characterized by analyzing the inclination and angular velocity of the thighs. After verifying, with direct comparison with an optoelectronic system, the accuracy of measurement of a wireless inertial system (with simply one inertial unit per thigh), we characterize the kinematic variables associated with the thighs' motion. We finally present preliminary experimental results, obtained with the proposed approach to the characterization of the LA task, both for a healthy subject and a PD patient, highlighting similarities and differences.

Chapter Outline

More specifically, the remainder of this chapter is structured as follows. Section 4.2 describes the considered experimental set-up, including the considered hardware, the subjects, and the experimental testbed. In Section 4.3, the proposed signal processing approach to the analysis of the LA task is described. Furthermore, the obtained experimental results are presented and commented. Finally, in Section 4.4 concluding remarks are given.

4.2 Experimental Set-up

The experiments were carried out at the San Giuseppe Hospital, Istituto Auxologico Italiano, in Piancavallo (Verbania, Italy), at a fully equipped last generation motion analysis laboratory. In particular, the kinematic analysis was carried out, in a comparative way, considering (i) an optoelectronic system and (ii) a wireless BSN-based system, based on a few nodes (equipped with inertial and magnetic sensors) placed over the body. In the following, the used optoelectronic and inertial systems will be described in details. Furthermore, the examined subjects and the considered experimental testbed will be also presented.

4.2.1 Hardware Description

The optoelectronic system (Vicon, Oxford, UK [1]) performs a real-time processing of images from 6 fixed infrared cameras (sampling at 100 Hz) to extract the reflectance of passive markers (with a diameter of 15 mm) which are positioned on specific anatomical landmarks of the subject. Prior to testing, the system was calibrated to assure accuracy and to allow the computation of each marker's three-dimensional coordinates. The mean error on the computation of the difference between measured and actual distances of two markers fixed on the extremities of a rigid bar was within 0.21 mm (with standard deviation equal to 0.1 mm). The calibrated volume for this application was 3.5 m in length (x axis of the laboratory reference system), 2 m in height (y axis of the laboratory reference system), and 2 m in width (z axis of the laboratory reference system).

The BSN is formed by Shimmer (Sensing Health with Intelligence, Modularity, Mobility, and Experimental Reusability) nodes [69]. A Shimmer node is a small and low-power wireless sensing platform that can capture and communicate a wide range of sensed data in real time. The main module is a compact wearable device (size: 53mm x 32mm x 25mm, weight: 22g) provided with: a TI MSP430 microcontroller; a Bluetooth radio (Roving Networks RN-42) and an IEEE 802.15.4 radio (TI CC2420); an integrated 2 GB microSD card slot; a 450mAh rechargeable Li-ion battery; and a triaxial accelerometer (Freescale MMA7361). Moreover, the device is

Table 4.2: Considered subjects.

Subject	Sex	Age	Weight	Height	UPDRS score for LA	Exercises
A (healthy)	female	40	56 Kg	171 cm	0	1xRLA, 1xLLA
B (PD)	male	42	85 Kg	180 cm	1 (bilaterally)	1xRLA, 1xLLA

designed so that different external sensing modules can be easily connected. In particular, the used external module is the 9DoF Kinematic Sensor expansion module, which is supplied with a triaxial gyroscope (InvenSense 500 series) and a triaxial magnetometer (Honeywell HMC5843).

4.2.2 Subjects

In this study, we evaluated the LA tasks performed by two individuals: one healthy subject (subject A) and a PD patient (subject B). Subject B has a disease duration of 4 years and does not present motor fluctuations. His Hoehn & Yahr score was 2 and the UPDRS score for LA was 1 bilaterally. The score of 1 was assigned in presence of 1 hesitation of the movement and a slight slowing during the limb motion. In Table 4.2, we summarize the information about the considered subjects, indicating also the performed exercises.

4.2.3 Experimental Testbed

As anticipated in Subsection 4.2.1, spatial and temporal parameters, along with the kinematics of the user lower limbs, have been monitored and evaluated using an inertial system and an optoelectronic system. Specifically the optoelectronic system has been used to estimate the three-dimensional positions of passive markers positioned on specific anatomical landmarks of the subject. Passive marker data were collected for all body segments (pelvis, thigh, shank, and foot, bilaterally). The Davis marker-set was chosen as the protocol of choice to acquire the motion of lower limbs and trunk based on [70, 71].

Concerning the inertial system, a Shimmer node (with sampling rate equal to 102.4 Hz) has been attached to each thigh of the monitored user with Velcro straps.

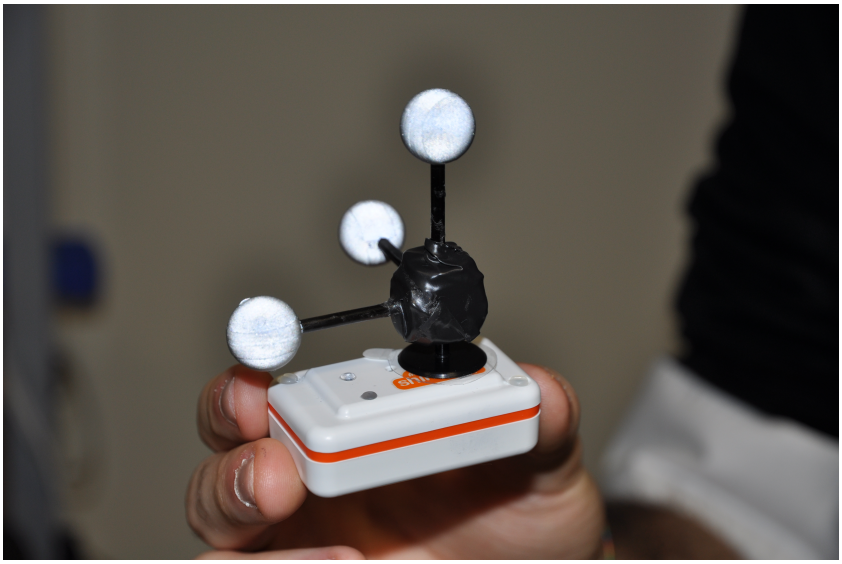


Figure 4.1: Three orthogonally positioned reflective markers fixed on a Shimmer node.

The Shimmer devices have been placed trying to align the plane defined by the x and y axes of the device with the frontal plane of the user and trying to align one of the two axes with the direction of the femur.

In addition to the markers specified in the Davis protocol, two groups of three markers were mounted on two frames fixed on two Shimmer devices (for a total of 6 additional markers). The three reflective markers are fixed on every Shimmer device through a frame of orthogonal rods of equal lengths (aligned as possible with the reference system of the Shimmer device), as shown in Figure 4.1. The estimation of the three-dimensional positions of the markers of a frame (with the optoelectronic system) allows to estimate the reference orientation of the device, which is used as ground truth of the actual Shimmer orientation. We remark that, because of imperfections in the frame design (i.e., the rods could not be perfectly orthogonal or have the same lengths), a "best-fit" orientation is estimated [72]. Furthermore, due to a possible misalignment between the Shimmer reference system and the frame, a cali-

bration step is performed once (at the beginning of each exercise) in order to estimate the fixed rotation between the two reference systems. This rotation is then applied to align the following measurements.

Finally, the inertial and optoelectronic systems (which are already independently synchronized) are synchronized together by computing the angular velocity of every optical frame and comparing it with that measured by the related Shimmer. The estimated time shift value is determined as the one which maximizes the correlation of the two signals.

For ease of clarity, in Figure 4.2 the experimental testbed used for the PD patient is shown.

4.3 Leg Agility Characterization

4.3.1 Derivation of Significant Variables

As anticipated earlier, according to the guidelines of the MDS, the LA item of the UPDRS should be evaluated by observing specific significant variables. As an example, the amplitude of the heel elevation and the speed of each repetition should be monitored, specifically focusing on their variations along the duration of the exercise. Furthermore, hesitations, interruptions, and freezing of the movements should be also evaluated. As shown in Table 4.1, general rules can be easily constructed in order to define an unambiguous mapping between observed variables and UPDRS scores.

These variables can be quite easily extracted from optical data just observing the estimated three-dimensional positions of each marker placed on the subjects' heels (one per heel) and, in particular, its "vertical" component, denoted as z_{H} (dimension: [m]), which then indicates its elevation. To this end, a segmentation of the LA data, manually performed upon frame-by-frame observation of the videos recorded for each session, is necessary. This segmentation allows to define three time labels, denoted as $t_{\mathrm{S}}(r)$, $t_{\mathrm{E}}(r)$, and $t_{\mathrm{P}}(r)$, associated, respectively, with the start, the end, and the epoch of maximal heel elevation (i.e., the elevation peak) of each repetition r ($r \in \{1,2,\ldots,10\}$) of LA task. Starting from these labels, the amplitudes of ascent and descent and their average (dimension: [m]), denoted, respectively, as $A_{\mathrm{A}}(r)$,

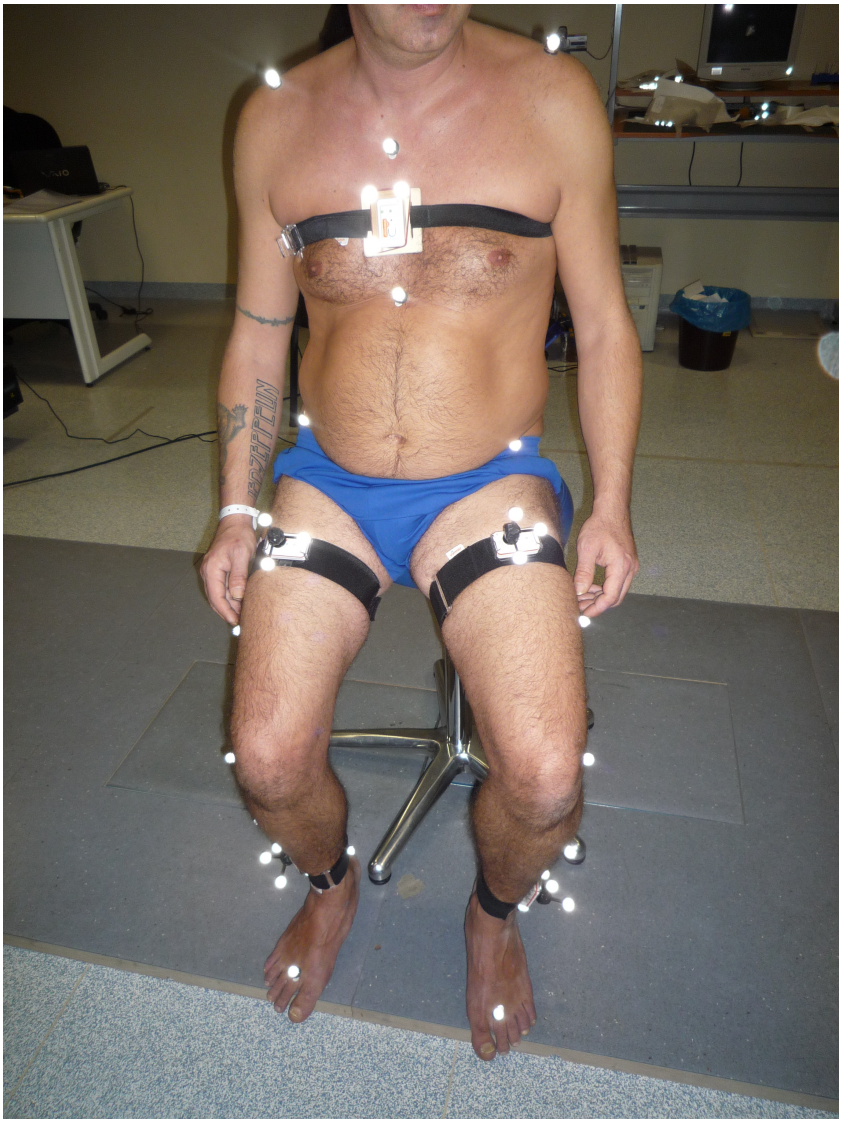


Figure 4.2: Overview of the experimental testbed applied to a monitored subject: both passive markers and Shimmer wireless nodes are considered.

$A_{\mathrm{D}}(r)$, and $A(r)$, computed for the r-th repetition, can be straightforwardly defined as follows:

$$A_{\mathrm{A}}(r) \triangleq z_{\mathrm{H}}(t_{\mathrm{P}}(r)) - z_{\mathrm{H}}(t_{\mathrm{S}}(r)) \tag{4.1}$$

$$A_{\mathrm{D}}(r) \triangleq z_{\mathrm{H}}(t_{\mathrm{P}}(r)) - z_{\mathrm{H}}(t_{\mathrm{E}}(r)) \tag{4.2}$$

$$A(r) \triangleq \frac{A_{\mathrm{A}}(r) + A_{\mathrm{D}}(r)}{2}. \tag{4.3}$$

The speed of execution $S(r)$ (dimension: [m/s]) of the r-th repetition can then be computed as

$$S(r) \triangleq \frac{A_{\mathrm{A}}(r) + A_{\mathrm{D}}(r)}{D(r)} = \frac{A_{\mathrm{A}}(r) + A_{\mathrm{D}}(r)}{t_{\mathrm{E}}(r) - t_{\mathrm{S}}(r)} \tag{4.4}$$

where $D(r)$ is the r-th repetition's duration (dimension: [s]). Hesitations, interruptions, and freezing of the movement are more difficult to define but they can be generally associated with sudden variations, fluctuations, or pauses in the z_{H} and in the linear "vertical" velocity $v_{\mathrm{H}} \triangleq \mathrm{d}z_{\mathrm{H}}/\mathrm{d}t$ (dimension: [m/s]).

First, we show that, in order to analyze the LA task, it is sufficient to consider the Shimmer nodes positioned over the thighs. In order to do this, we verify that the analysis of thighs' kinematics (measured through the inertial system) is actually equivalent to that of the heels' kinematics (measured through the optoelectronic system).

To this end, the three-dimensional orientation of a Shimmer device is estimated through an orientation filter based on a gradient descent algorithm which properly weighs the measurements of the three sensors of the Shimmer (i.e., accelerometer, gyroscope, and magnetometer) [73]. The inclination θ (dimension: [degrees]) of the thigh is then computed as the angle between the Shimmer axis (parallel to the femur direction) and the world vertical axis, reduced by 90 degrees—this is expedient to measure 0 degrees when the subject is sitting. Moreover, the angular velocity of the thighs, directly measured through the Shimmer's gyroscope, is considered. In particular, we define as ω (dimension: [degrees/s]) the component of the angular velocity measured around the Shimmer axis perpendicular to the femur direction and lying in the frontal plane of the user.

At this point, z_{H} has been compared to θ and v_{H} has been compared to ω. For both the subjects, the correlation between z_{H} and θ is between 0.98 and 0.99 and the correlation between v_{H} and ω is between 0.93 and 0.98, showing then a strong correlation between heels' optical data and thighs' inertial data—no graph is shown for lack of space. Therefore, in the following analysis, we will just consider the signals extracted from the Shimmer of the thighs, i.e., θ and ω.

Therefore, relevant kinematic variables of the LA task can be redefined, from an inertial perspective, with reference to the Shimmer nodes on the thighs. More formally, the angular amplitudes of ascent and descent and their average (dimension: [deg]), denoted as, $\Theta_{\mathrm{A}}(r)$ $\Theta_{\mathrm{D}}(r)$, and $\Theta(r)$, computed for the r-th repetition are defined as

$$\Theta_{\mathrm{A}}(r) \triangleq \theta\left(t_{\mathrm{P}}(r)\right) - \theta\left(t_{\mathrm{S}}(r)\right) \tag{4.5}$$

$$\Theta_{\mathrm{D}}(r) \triangleq \theta\left(t_{\mathrm{P}}(r)\right) - \theta\left(t_{\mathrm{E}}(r)\right) \tag{4.6}$$

$$\Theta(r) \triangleq \frac{\Theta_{\mathrm{A}}(r) + \Theta_{\mathrm{D}}(r)}{2} \tag{4.7}$$

and the angular speed of execution $\Omega(r)$ (dimension: [deg/s]) of the r-th repetition can be evaluated as

$$\Omega(r) \triangleq \frac{\Theta_{\mathrm{A}}(r) + \Theta_{\mathrm{D}}(r)}{D(r)} = \frac{\Theta_{\mathrm{A}}(r) + \Theta_{\mathrm{D}}(r)}{t_{\mathrm{E}}(r) - t_{\mathrm{S}}(r)}. \tag{4.8}$$

It is then of interest to monitor and quantitatively analyze how the variables above (namely, Θ and Ω) behave in healthy subjects and PD patients. Indeed, full understanding of the meaning of these variables could allow to develop an automatic system able to easily translate their measures into UPDRS scores, still providing a precious quantitative analysis to the neurologists.

4.3.2 Comparison between a Healthy Subject and a PD Patient

In order to highlight the accuracy in the estimation of the thigh inclination θ provided by the orientation filter acting on Shimmer data, in Figure 4.3 a direct comparison of the inclination estimated through the optoelectronic system (by means of the orthogonal frame of markers shown in Figure 4.1) and that estimated through the inertial

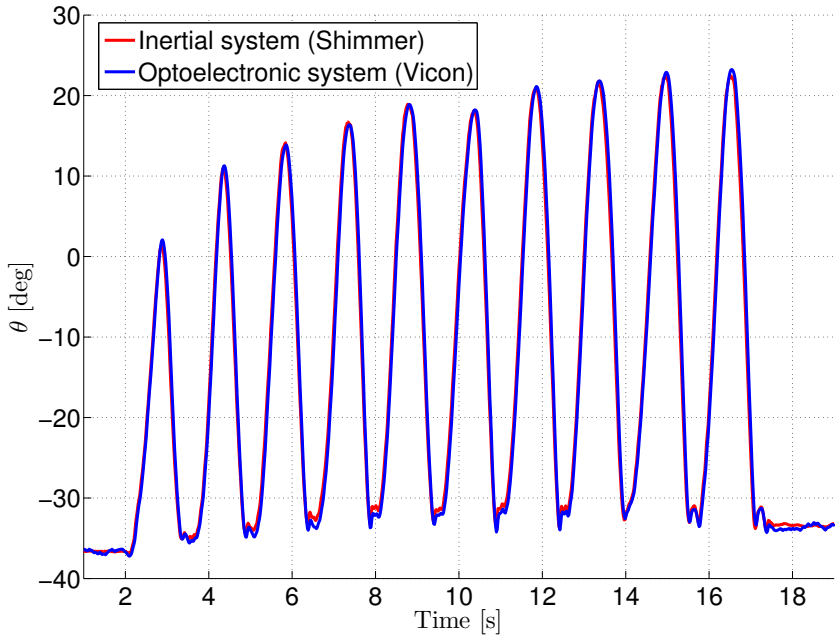
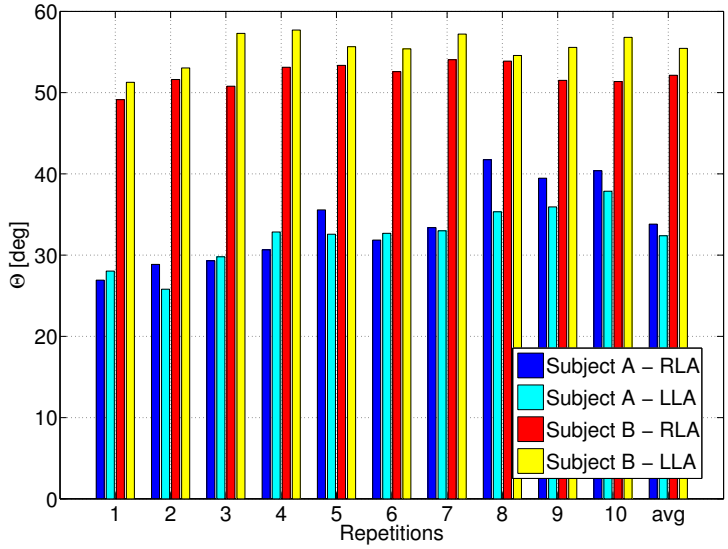


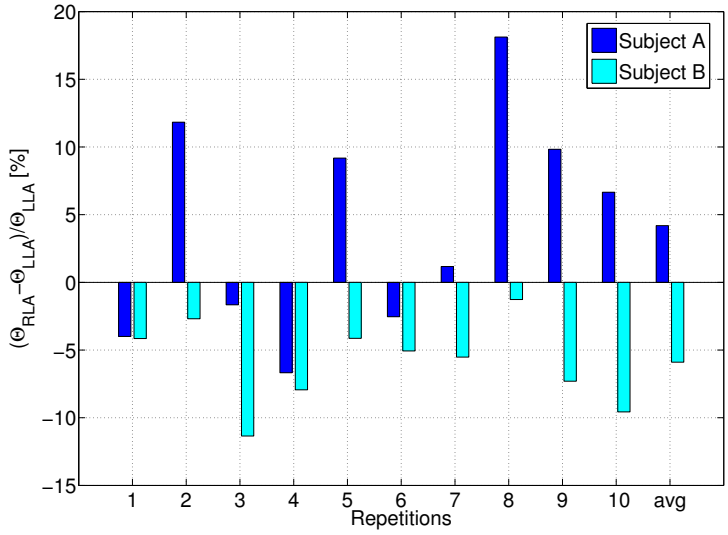
Figure 4.3: Direct comparison between the inclinations estimated through optoelectronic (Vicon) and inertial (Shimmer) systems.

sensor is shown. It is easy to see that, even if the subject movement presents a high dynamicity, the error between the inclination measured with the two systems is almost negligible. Therefore, the analysis conducted in the following will be based on inertial data.

First, we analyze the amplitude and the speed of each repetition, indicated in the MDS's UPDRS document as the main variables to observe in the LA task. In Figure 4.4 (a) the angular amplitude Θ of each repetition (from 1 to 10) is shown, comparing RLA and LLA performed by both Subject A and Subject B. Moreover, we also show the angular amplitude averaged over the ten repetitions. For both subjects, Θ does not decrease over time, even if its absolute value is higher in Subject B. On the contrary, the angular amplitude of Subject A seems to increase repetition by repetition. However, focusing on the difference in angular amplitude between RLA



(a)



(b)

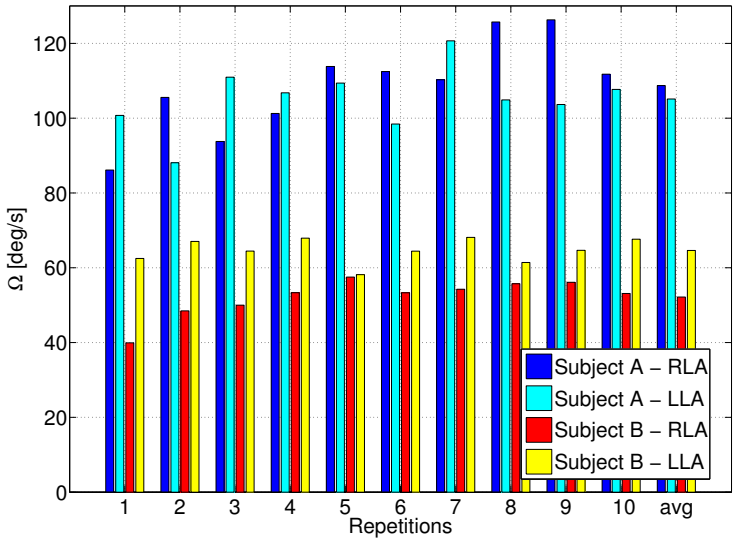
Figure 4.4: Comparison, over ten repetitions, between Subject A (healthy) and Subject B (PD), in terms of: (a) angular amplitude Θ for RLA and LLA and (b) relative angular amplitude difference (percentage) between the two legs.

and LLA, expressed in percentage in Figure 4.4 (b), it can be noticed that Subject A does not present a biased difference between RLA and LLA, whereas Subject B's RLA angular amplitude is always lower than that of LLA. On average, it can be observed that the relative difference between RLA and LLA is around 4% for Subject A and around 6% for Subject B.

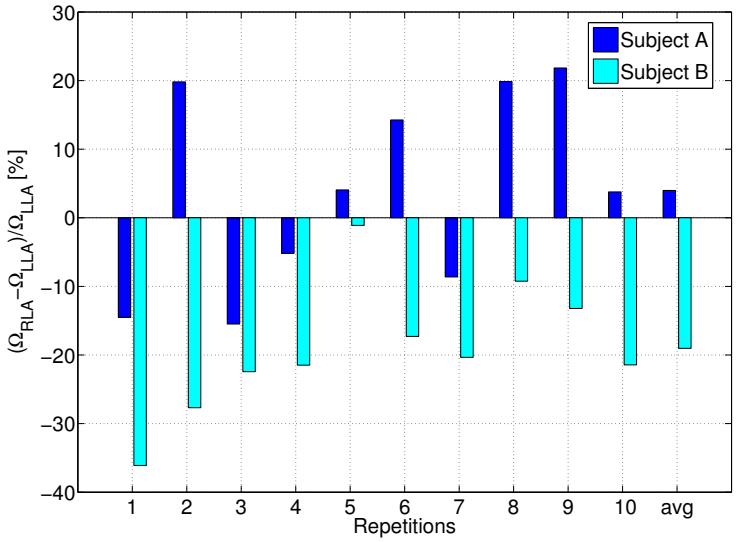
Concerning the speed of the LA task, in Figure 4.5 (a) the RLA and LLA angular speeds are evaluated and compared for both Subject A and Subject B. Unlike what observed in Figure 4.4 for the amplitude, the LA angular speed of Subject B is averagely half the LA angular speed of Subject A. Again, even if the angular speed of Subject B is higher than that of Subject A, the LA angular speed for both subjects does not decrease over time. However, as for amplitude, the difference (expressed in percentage) between the RLA and LLA angular speeds, shown in Figure 4.5 (b), reveals that Subject B's RLA angular speed is 19% lower than his LLA angular speed. On the contrary, this is generally not true for Subject A, for which the relative difference is, on average, around 4%.

According to the previous analysis, it is worth to highlight that the differences observed between the RLA and the LLA of a specific subject, even if not specifically referenced in the MDS's document as a significant variable, can instead represent a clear evidence of a non-zero UPDRS score. Therefore, a possible extension of the UPDRS can be already envisioned just by introducing this new variable in the LA analysis. Moreover, an automatic approach could possibly overcome the limitations and subjectivity of a neurologist evaluation or, at least, represent a precious support instrument.

A third variable which has been analyzed is the *regularity* of the execution rhythm of the LA task. In particular, possible hesitations in the execution of LA have been monitored evaluating the time distance between the end of a repetition ($t_{\mathrm{E}}(r)$) and the start of the following repetition ($t_{\mathrm{S}}(r+1)$), i.e,, the "pause" between consecutive leg movements, shown in Figure 4.6 (a) for RLA and LLA of both Subjects A and B—note that the pause time for Subject A's RLA is always zero, i.e., she is doing continuous movements. Another variable related to regularity is the distance between consecutive peak times (t_{P}), shown in Figure 4.6 (b) for RLA and LLA of both sub-

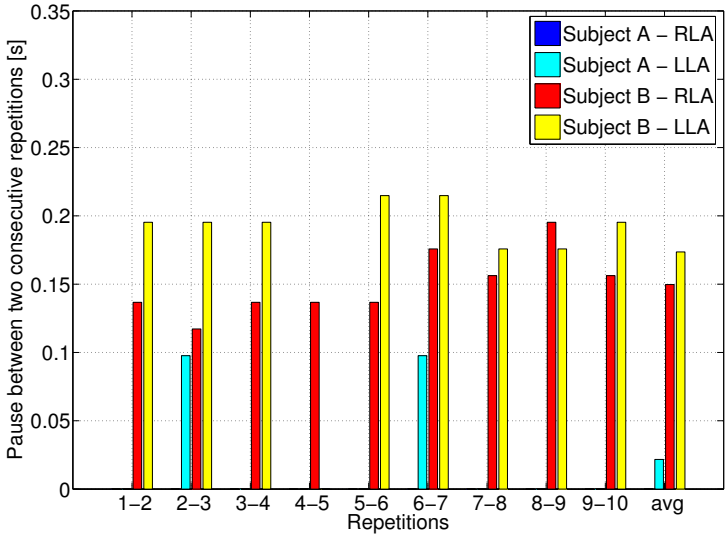


(a)

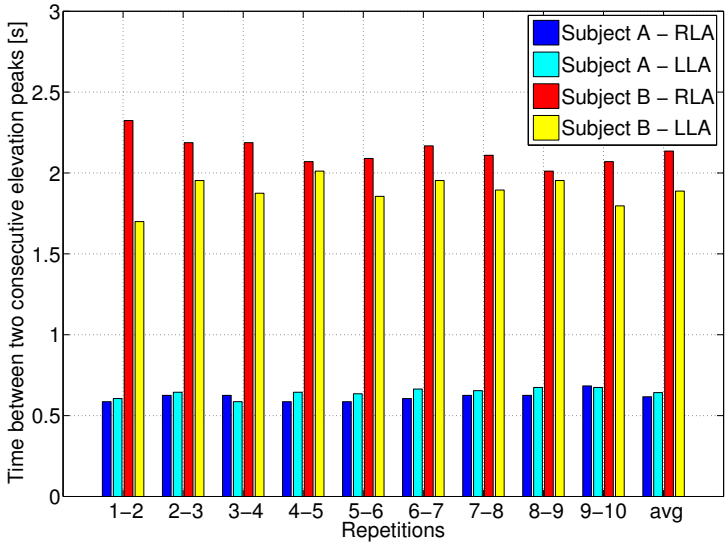


(b)

Figure 4.5: Comparison, over ten repetitions, between Subject A (healthy) and Subject B (PD), in terms of: (a) angular speed Ω for RLA and LLA and (b) relative angular speed difference (percentage) between the two legs.



(a)



(b)

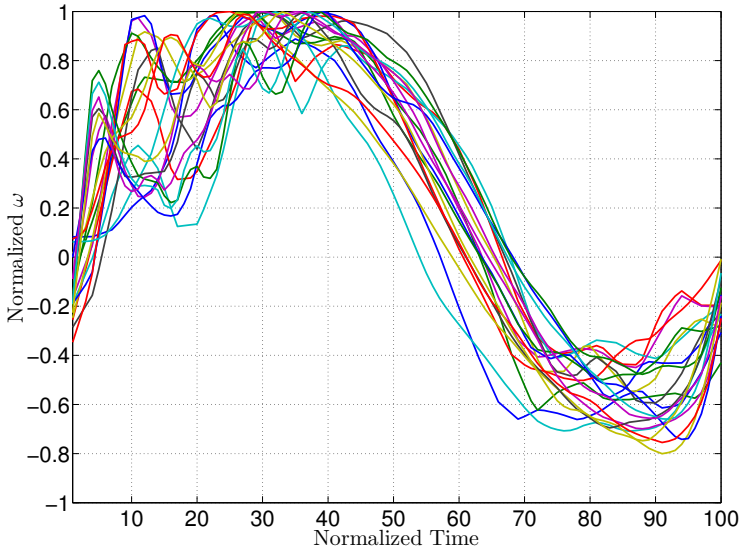
Figure 4.6: Comparison, over ten repetitions, of RLA and LLA of Subject A (healthy) and Subject B (PD), in terms of: (a) pauses (between consecutive leg movements) and (b) distance between the epochs of consecutive peaks of angular movements.

jects. Concerning the peak-to-peak time distance, it is easy to see that the rhythm of Subject A is definitely more regular than that of Subject B, which, as already observed for the amplitude and speed, shows a significant difference between RLA and LLA. The difference is even clearer by observing that, unlike what happens with Subject A, Subject B's repetitions are separated by pauses between 0.1 s and 0.25 s.

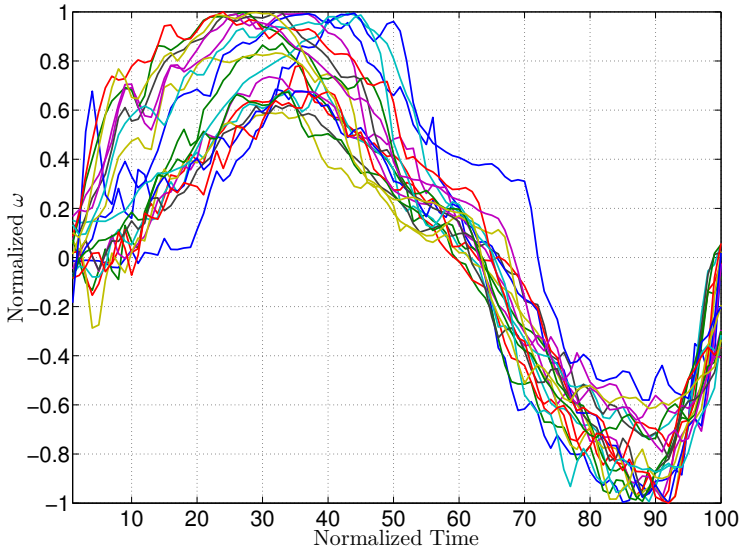
Finally, in order to better investigate and characterize the LA repetitions, a qualitative analysis has been done comparing the angular velocities ω along an entire repetition. In particular, in Figure 4.7, the segmented portions of ω corresponding to each LA repetition have been normalized in time and value (so that time goes from 1 to 100 and $-1 \leq \omega \leq +1$) and overlapped for (a) Subject A and (b) Subject B, respectively. Even if both subjects show some macro-hesitations (specifically when raising up the heels), it is easy to see that Subject B's angular velocity presents several micro-hesitations, not easily observable in recorded videos. This can be then another useful instrument in order to better investigate the LA task and predict the presence of PD.

4.4 Conclusions and Future Work

In this chapter, we have characterized, from a kinematic perspective, the LA task, used to evaluate the severity of the PD. In particular, by using wireless inertial devices (Shimmer), we have analyzed, from a kinematic perspective, the motion of the thighs (using one inertial device per thigh). By considering the angular amplitude and speed of the inertial device on the thighs, a direct comparison between a healthy subject and a PD patient has lead to interesting insights. In fact, our results show that, with respect to a healthy patient, a PD patient is likely to present: (i) a higher asymmetry of movement between right and left legs and (ii) micro-hesitations along the movement. Future developments include the evaluation of a larger number of PD patients to verify if the emerged kinematic characteristics are confirmed. Furthermore, the proposed approach will be extended to other motor tasks considered by the MDS.



(a)



(b)

Figure 4.7: Juxtaposed normalized (between ± 1) angular velocities ω of ten repetitions, normalized over time (between 1 and 100) for (a) Subject A and (b) Subject B.

Chapter 5

Motion Reconstruction

..and we can build through this destruction
as we are standing on our feet..

– Gavin DeGraw (Follow Through)

In previous chapters, motion analysis has been restricted to specific parts of the user body. This is, however, somehow limiting in applications where the full body motion of a user needs to be investigated. Although very accurate and precise motion capture systems have been already developed and are commercially available, these solutions rely on the use of a large number of sensor devices, leading to an increased cost and complexity. Thus, it would be of interest to investigate more compact solutions, based on small subsets of these devices, which still allow to estimate as precisely as possible the full body pose of the user. To this end, in this chapter, a *motion reconstruction* approach is described and experimentally investigated. The proposed system is composed of only five sensor devices, placed on the wrists, the ankles, and the pelvis of the user, and is able to estimate his/her full body pose by relying on the use of a training dataset of previously recorded human poses.

5.1 Introduction

The artificial synthesis of human motion has become in the last two decades an important research area, especially due to the high interest in areas such as computer graphics, computer animation, and virtual reality. To this end, many efforts have been spent in developing motion capture systems based on different sensing technologies (e.g., optical [1], inertial [3, 4], mechanical [5], magnetic [6], etc.). To this end, many efforts have been spent in developing motion capture systems based on different sensing technologies (e.g., optical, inertial, mechanical, magnetic, etc.). Though these systems have been demonstrated as very accurate and are currently used in many applications where nearly exact estimation of motion parameters is required, they are all still considered expensive and complex (in terms of body accessories and/or subject preparation). Nevertheless, the increasing availability of motion capture data obtained through these accurate systems has provided a solid basis upon which new motion synthesis techniques can be developed. Specifically, even if first attempts of motion synthesis were based on properly constructed biomechanical and physical models, more recent works rely on the development of data-driven approaches. More generally, according to the survey in [74], these techniques can be distinguished in *procedural* (based on empirical and biomechanical models, they offer high-level control but are not perceived as realistic [75, 76, 77, 78]), *physics-based* (based on dynamics and physical properties, they offer less control but appear more realistic [79, 80, 81, 82]), and *example-based* (where existing motion data are exploited in order to generate realistic motion sequences [83, 84]).

Concerning data-driven motion synthesis, several works have focused on the extraction of models of motion trying to reduce the complexity of the original motion and to determine its mapping to lower dimensional subspaces, using techniques such as, for instance, Principal Component Analysis (PCA) [85], Fourier analysis [86], probabilistic PCA [87], motion texture [88], Gaussian process latent variable models [89], multi-dimensional scaling [90], binary latent variables [91, 92], and Gaussian process dynamical models [93, 94].

More generally, realistic data-driven motion synthesis typically relies on tech-

niques based on motion blending and space/time warping [95, 96, 84, 97, 98, 99, 100, 101, 102, 83] and deals with concepts such as style [103, 104, 89, 105] and retargeting [96, 106, 107, 108, 102, 89].

Concerning the interactive control of synthesized motion, besides the concept of "driving" the synthesized motion through synthetic variables such as center of mass trajectories or footsteps paths [109, 110, 111, 112, 113, 114, 115, 116, 117], the (real-time) control of virtual characters using actual movements of a user, known as *computer puppetry* [118, 108], has also been widely studied and investigated. In this context, a low-dimensional sensor input is often used (e.g., extracted from optical data [119, 120], inertial data [121, 122, 123, 124] , and acoustic/inertial data [125, 126]), together with the use of online local dynamic motion models [125, 126], motion graphs [124], and search techniques such as Nearest Neighbors [119, 127, 123].

5.1.1 Chapter Contribution

In this work, a proof-of-concept investigation of accurate full body motion reconstruction with unobtrusive, cheap, and low-power body-only worn sensor devices is presented. Specifically, the proposed system is conceived to work with only 5 sensor devices, placed on the wrists, the ankles, and the pelvis of a user. Even taking into account that an accuracy loss (proportional to the amount of missing information) is unavoidable, the aim of this work is to make the reconstructed motion to still be perceived as realistic and natural. Note that one of the key novelties of this work consists of the fact that the proposed motion reconstruction system does not rely on any external infrastructure (i.e., it is self-contained), thus enabling user motion freedom.

Since the information provided by five sensor devices is incomplete, with respect to that of full body motion capture systems, the motion of the body segments which are not equipped with sensor devices has to be reconstructed on the basis of previously collected motion data (i.e., a *training dataset*). High-dimensional full body poses, contained in the training dataset, are then mapped to lower dimensional characteristic features that can also be computed, at *runtime*, using the available sensors. In particular, the following features, which imply the use of different sensor technologies, have been considered: relative orientations (through sensor fusion of

accelerometers, gyroscopes, and magnetometers); relative heights (through barometers); and relative ranges (trough Ultra-WideBand (UWB) ranging). A novel motion reconstruction framework is thus introduced and described, where a Nearest Neighbors Search (NNS) in the feature domain is used to identify suitable full body poses in the training dataset that can be elected as candidates for the reconstruction of the actual user pose. Full body kinematics of an individual has been evaluated using a commercially available inertial/magnetic full body suit (namely, Xsens MVN Awinda [4, 16]), composed of 17 inertial/magnetic sensor devices. The collected motion data have been then split into disjoint training and testing datasets in order to, respectively, train and test the proposed system.

We remark that, even if this work just presents a proof-of-concept investigation, the proposed system has the potential, from a technological viewpoint, to be amenable for practical implementation, since the involved sensor technologies have indeed the potential for low-power, low-cost, and small size. To this end, note that cost, size, power, and integration efforts of these sensors are already dramatically decreased due to recent market developments (e.g., concerning accelerometers. gyroscopes, magnetometers, and barometers, the drive from the smartphone market [128] and to some recently commercially released UWB chips (e.g. from Decawave [129]).

Chapter Outline

The rest of this chapter is structured as follows. Section 5.2 is dedicated to related work. In Section 5.3, some preliminaries on notation, expedient to the following derivation, are given. In Section 5.4, the general idea of NNS-based human motion reconstruction is introduced. In Section 5.5, the proposed motion reconstruction framework is presented and thoroughly described. In Section 5.6, the experimental set-up and the performance metrics of interest are presented. In Section 5.7, the proposed system is experimentally investigated. Finally, in Section 5.8, conclusions and future work are discussed.

5.2 Related Work

Other studies, similar to ours, have already focused on the reconstruction of human motion by exploiting a training dataset of previously collected motion data. In the following, a brief description of these works (in chronological order of appearance) is given.

- In [119], an approach to optical-based performance animation is introduced. This approach relies on the use of video cameras and a small set (namely, from 6 to 9) of retro-reflective optical markers placed on the user body. The low-dimensional control data, obtained from the optical markers position data, are then used to construct a local model of the user motion by searching into a prerecorded set of motion capture data. PCA is also used in order to properly map human motion into a lower dimensional space. Even if the presented performance is very promising, one drawback of this approach is that the use of video cameras and reflecting markers (and, thus, of an external infrastructure) limits the motion freedom of the user which is constrained to always stay in a specific recording area. Finally, markers' occlusions and light conditions may also be critical problems to take care of.

- In [123], an approach conceptually similar to the previous one is presented, relying on the use of 4 triaxial accelerometers attached to the user wrists and ankles. A training dataset of prerecorded motion capture data is still expedient and an efficient neighborhood graph is constructed and exploited to search for similar acceleration patterns in this database. The performance is again very promising and the fact that the system, composed of only 4 sensor devices, does not rely on any external infrastructures is a strong plus of this approach. However, acceleration data are often noisy (e.g., due to unwanted vibrations or slight misplacements of the devices) and, if considered alone, are not necessarily as reliable as position data and may thus lead to errors while searching for similar patterns in the training dataset. Furthermore, as already mentioned in Section 5.1, the use of only accelerometers, in order to keep the system cost low, is not that justified anymore today.

- In [125], a data-driven motion reconstruction approach based on the use of 6 sensor devices placed on the user ankles, hands, pelvis, and head is described. Both orientation (through the use of inertial and magnetic sensors) and position (through ultrasonic ranging with respect to an external infrastructure) data are collected and used as a low-dimensional control signal. The control signal is then jointly used with local dynamic models, constructed using a training dataset of motion capture data, to predict the user pose in a maximum a posteriori framework. As for [119], the weakness of this approach relies on its dependency on an external infrastructure.

- In [126], an approach very similar to that of [125] is presented, using 5 sensor devices (head device is not used here) which produce orientation and position data. However, a non-linear statistical model is here used to learn, from the training dataset, the non-linear structures of high-dimensional motion data in order to transform the low-dimensional input signal into a high-dimensional human pose.

We remark that, unlike solutions presented in [119, 125, 126] where external infrastructures are expedient to the computation of the required features, the key novelty of our motion reconstruction system relies in its self-containedness. On the other hand, even if in [123] only accelerometers are used (and the system is thus self-contained too), long acceleration patterns (rather than instantaneous values) have to be compared in order to minimize the ambiguity in the motion reconstruction. To this end, note that two poses may be very similar, in terms of body segments positions and orientations, and still be associated with very different acceleration values, depending on the actual execution of the whole motion patterns to which they belong. Instead, the input features allowed by our framework, which are meaningfully richer than acceleration values alone, guarantee the system to work properly even if the whole motion patterns are not available in the training dataset.

5.3 Preliminaries on Notation

As stated in Section 5.1, the proposed system aims at *reconstructing the full body motion of a user* who is wearing a small number of sensor devices. Generally, *full body motion* means determining both the *pose* of the user (i.e., the orientations of his/her body segments and the positions of his/her joints with respect to an origin on his/her body) and his/her *absolute position* in a global reference system (i.e., the positions of his/her joints with respect to an origin on, e.g., the Earth). However, in this work, we will just focus on the reconstruction of the *pose* of the user.[1] In particular, in the following, we always assume that the origin of our reference system is the user pelvis and that all body segment positions are expressed with respect to that on-body reference.

5.3.1 Describing the User Pose

If we assume that a proper biomechanical model [34] is known (i.e., the body structure and the body segments' lengths are known), a generic user pose $\boldsymbol{p}^{\mathrm{O}}$ can be expressed, by using body segments' (four-dimensional) orientations, as follows:

$$\boldsymbol{p}^{\mathrm{O}} = \begin{bmatrix} \boldsymbol{q}^1 & \boldsymbol{q}^2 & \dots & \boldsymbol{q}^S \end{bmatrix} \tag{5.1}$$

where S is the number of all rigid body segments in the considered biomechanical model and

$$\boldsymbol{q}^i = \begin{bmatrix} \boldsymbol{q}^i_s & \boldsymbol{q}^i_x & \boldsymbol{q}^i_y & \boldsymbol{q}^i_z \end{bmatrix} \qquad i \in \{1,2,\dots,S\} \tag{5.2}$$

is the quaternion describing the *absolute* (i.e., with respect to the Earth reference system) orientation of the i-th body segment.[2] Note that, according to this notation, the

[1] Note that the problem of determining the absolute position of the user (i.e. motion tracking) can be often solved by assuming that the lowest point of his/her body is in contact with the ground (i.e., contact detection) and, therefore, has zero height. Therefore, by simply "pivoting" the whole body around this point, the user motion can be naively *tracked* in the global reference system.

[2] Quaternions (i.e., four-dimensional unit vectors [38]) are considered, instead of Euler angles (i.e., yaw, pitch, and roll) and rotation matrices, to describe body segment orientations because of their compactness and implicit robustness to three-dimensional rotation ambiguities (e.g., gimbal lock).

full body pose of a user can be expressed using a $(4 \cdot S)$-dimensional vector. If one now wants to derive body segments' (three-dimensional) positions $\boldsymbol{p}^{\mathrm{P}}$ corresponding to $\boldsymbol{p}^{\mathrm{O}}$, a forward kinematics function $g^{\mathrm{FK}}\colon \mathbb{R}^{4S} \rightarrow \mathbb{R}^{3S}$ [34], which reflects the considered biomechanical model, can be used as follows:

$$\begin{aligned} \boldsymbol{p}^{\mathrm{P}} &= g^{\mathrm{FK}}\left(\boldsymbol{p}^{\mathrm{O}}\right) && (5.3)\\ &= \begin{bmatrix} \boldsymbol{c}^1 & \boldsymbol{c}^2 & \dots & \boldsymbol{c}^S \end{bmatrix} && (5.4) \end{aligned}$$

with

$$\boldsymbol{c}^i = \begin{bmatrix} \boldsymbol{c}^i_x & \boldsymbol{c}^i_y & \boldsymbol{c}^i_z \end{bmatrix} \qquad i \in \{1,2,\dots,S\} \tag{5.5}$$

where $\boldsymbol{c}^i_x$, $\boldsymbol{c}^i_y$, and $\boldsymbol{c}^i_z$ are, respectively, the x, y, and z coordinates of the i-th body segment. We remark that, for the purpose of this work, we are not interested in the user global position and, therefore, pelvis coordinates are always set to be equal to $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ and all the other body segments' positions are expressed with respect to it.

For reasons that will be clearer later on, the user pose can also be expressed, without loss of generality, in terms of (i) *body shape* and (ii) *absolute orientation of the body shape*. To this end, one body segment must be chosen to be the *reference body segment*. More formally, indicating as $i_1 \in \{1,2,\dots,S\}$ the index of the reference body segment in Equation (5.1), we may denote as $\boldsymbol{q}^{\sim} \triangleq \boldsymbol{q}^{i_1}$ the *absolute orientation* of the reference body segment (i.e., with respect to the Earth reference system). Furthermore, denoting the *relative orientations* of the remaining $S-1$ body segments, with respect to that of the reference one, as $\{\boldsymbol{q}^{1,\sim}, \boldsymbol{q}^{2,\sim}, \dots, \boldsymbol{q}^{S-1,\sim}\}$, the pose of the user, previously expressed as in Equation (5.1), can be equivalently expressed as follows:

$$\boldsymbol{p}^{\mathrm{O},\sim\sim} = \begin{bmatrix} \boldsymbol{q}^{\sim} & \boldsymbol{q}^{1,\sim} & \boldsymbol{q}^{2,\sim} & \dots & \boldsymbol{q}^{S-1,\sim} \end{bmatrix}. \tag{5.6}$$

In order to further highlight the orientation of the reference body segment, by denoting as

$$\boldsymbol{p}^{\mathrm{O},\sim} \triangleq \begin{bmatrix} \boldsymbol{q}^{1,\sim} & \boldsymbol{q}^{2,\sim} & \dots & \boldsymbol{q}^{S-1,\sim} \end{bmatrix}, \tag{5.7}$$

the pose in Equation (5.6) can be also written as

$$\begin{aligned} \boldsymbol{p}^{\mathrm{O},\sim\sim} &= \left\{ \left[\boldsymbol{q}^{\sim} \right], \left[\boldsymbol{p}^{\mathrm{O},\sim} \right] \right\} \\ &= \left\{ \left[\boldsymbol{q}^{\sim} \right], \left[\boldsymbol{q}^{1,\sim} \quad \boldsymbol{q}^{2,\sim} \quad \dots \quad \boldsymbol{q}^{S-1,\sim} \right] \right\}. \end{aligned} \tag{5.8}$$

Although Equations (5.1) and (5.8) are conceptually equivalent and can be alternatively used to define the user pose, two contributions emerge clearly in Equation (5.8): $\boldsymbol{p}^{\mathrm{O},\sim}$ (which only contains relative orientations) gives information about the *shape of the body*, whereas $\boldsymbol{q}^{\sim}$ gives information about the *absolute orientation of that shape* with respect to the Earth reference system. However, since both of them describe the same pose,

$$\boldsymbol{p}^{\mathrm{P}} = g^{\mathrm{FK}}\left(\boldsymbol{p}^{\mathrm{O}}\right) = g^{\mathrm{FK}}\left(\boldsymbol{p}^{\mathrm{O},\sim\sim}\right). \tag{5.9}$$

5.3.2 Describing the Training Dataset of Poses

As anticipated in Section 5.1, in order to estimate the full body pose of the user, the proposed motion reconstruction system exploits a training dataset of previously recorded human poses to bridge the information gap caused by the small number of available sensor devices.

Straightforwardly, *a training dataset of poses* $\overline{\boldsymbol{P}}^{\mathrm{O}}$ can be expressed as the concatenation of N distinct poses as follows:

$$\overline{\boldsymbol{P}}^{\mathrm{O}} = \begin{bmatrix} \overline{\boldsymbol{p}}_1^{\mathrm{O}} \\ \overline{\boldsymbol{p}}_2^{\mathrm{O}} \\ \vdots \\ \overline{\boldsymbol{p}}_N^{\mathrm{O}} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{q}}_1^1 & \overline{\boldsymbol{q}}_1^2 & \cdots & \overline{\boldsymbol{q}}_1^S \\ \overline{\boldsymbol{q}}_2^1 & \overline{\boldsymbol{q}}_2^2 & \cdots & \overline{\boldsymbol{q}}_2^S \\ \vdots & \vdots & & \vdots \\ \overline{\boldsymbol{q}}_N^1 & \overline{\boldsymbol{q}}_N^2 & \cdots & \overline{\boldsymbol{q}}_N^S \end{bmatrix} \tag{5.10}$$

where $\overline{\boldsymbol{q}}_n^i$ represents the absolute orientation of the i-th body segment ($i \in \{1,2,\dots,S\}$) of the n-th pose ($n \in \{1,2,\dots,N\}$) of the training dataset $\overline{\boldsymbol{P}}^{\mathrm{O}}$. Note that, from a notational viewpoint, the "overline" notation will be used from now on to refer to *training* variables. Note also that, for the purposes of the motion reconstruction, all the poses contained in $\overline{\boldsymbol{P}}^{\mathrm{O}}$ do not necessarily belong to a single motion sequence (i.e. the concatenation of the N poses does not represent a continuous movement). Instead, several

motion sequences (that can be related to different users and different data acquisition sessions) may be properly concatenated to compose an overall training dataset.[3]

Similarly, a *training dataset of body shapes* $\overline{\boldsymbol{P}}^{\mathrm{O},\sim}$ can be expressed as the concatenation of N distinct body shapes as follows:

$$\overline{\boldsymbol{P}}^{\mathrm{O},\sim} = \begin{bmatrix} \overline{\boldsymbol{p}}_1^{\mathrm{O}\sim} \\ \overline{\boldsymbol{p}}_2^{\mathrm{O}\sim} \\ \vdots \\ \overline{\boldsymbol{p}}_N^{\mathrm{O}\sim} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{q}}_1^{1,\sim} & \overline{\boldsymbol{q}}_1^{2,\sim} & \cdots & \overline{\boldsymbol{q}}_1^{S-1,\sim} \\ \overline{\boldsymbol{q}}_2^{1,\sim} & \overline{\boldsymbol{q}}_2^{2,\sim} & \cdots & \overline{\boldsymbol{q}}_2^{S-1,\sim} \\ \vdots & \vdots & & \vdots \\ \overline{\boldsymbol{q}}_N^{1,\sim} & \overline{\boldsymbol{q}}_N^{2,\sim} & \cdots & \overline{\boldsymbol{q}}_N^{S-1,\sim} \end{bmatrix} \tag{5.11}$$

where $\overline{\boldsymbol{q}}_n^{i,\sim}$ represents the relative orientation (with respect to that of the reference body segment) of the i-th body segment ($i \in \{1,2,\ldots,S\}$) of the n-th pose ($n \in \{1,2,\ldots,N\}$) of the training dataset $\overline{\boldsymbol{P}}^{\mathrm{O},\sim}$.

5.4 NNS-based Human Motion Reconstruction: the General Idea

In this section, the general idea behind a NNS-based human reconstruction system is described. An NNS-based human motion reconstruction system generally relies on an NNS (performed at *runtime*) among a training dataset of features which has a direct correspondence with a training dataset of poses (where both training datasets are built during a preliminary *training* phase). In this section, the general idea behind a NNS-based human reconstruction system is intuitively described. In Section 5.5, an improved version of NNS-based human motion reconstruction will be presented and further details, more specific to the proposed framework, will be given.

5.4.1 Training

Let us assume that, during a preliminary training phase, a training dataset of poses $\overline{\boldsymbol{P}}^{\mathrm{O}}$, containing N body poses, is built as shown in Equation (5.10). Besides the training

[3] For the same reason, a training dataset can also be continuously extended and, thus, improved every time new motion capture data are collected.

dataset of poses, a training dataset of features $\overline{\boldsymbol{F}}$ may be constructed where each of its rows contains a feature vector $\overline{\boldsymbol{f}}$, which is representative of (and can be computed from) the corresponding pose in $\overline{\boldsymbol{P}}^{\mathrm{O}}$. More formally, $\overline{\boldsymbol{F}}$ can be written as follows:

$$\overline{\boldsymbol{F}} = \begin{bmatrix} \overline{\boldsymbol{f}}_1 & \overline{\boldsymbol{f}}_2 & \dots & \overline{\boldsymbol{f}}_T \end{bmatrix}^{\mathrm{T}} \tag{5.12}$$

where: $\overline{\boldsymbol{f}}_n$ ($n \in \{1,2,\dots,N\}$) is the feature vector corresponding to the n-th pose in $\overline{\boldsymbol{P}}^{\mathrm{O}}$ (i.e., $\overline{\boldsymbol{p}}^{\mathrm{O}}_n$); and $[\cdot]^{\mathrm{T}}$ denotes vector transposition.

More details on the considered types of features will be given in Subsection 5.5.2. Just note, however, that a feature vector should aim at uniquely characterizing the corresponding pose with respect to the other ones and, since after the training phase (i.e., at runtime) only few sensor devices will be available and full body pose data will not, it must be also possible to compute (estimate) it directly using the available sensor devices (specifically, through the sensors they are equipped with). Finally, note that, generally, $\overline{\boldsymbol{F}}$ represents a mapping of (high-dimensional) $\overline{\boldsymbol{P}}^{\mathrm{O}}$ onto a lower-dimensional space.

5.4.2 Runtime

At runtime, assuming to have properly constructed $\overline{\boldsymbol{F}}$ and $\overline{\boldsymbol{P}}^{\mathrm{O}}$, every time a new pose $\boldsymbol{p}^{\mathrm{O}}$ of the user has to be reconstructed, a feature vector $\boldsymbol{f}$ containing the same features considered in the training phase should be computed using the available sensor devices and compared with those contained in $\overline{\boldsymbol{F}}$. If the features have been properly chosen, it is expected that the pose which corresponds to the (training) feature vector with minimal distance from $\boldsymbol{f}$ is the most (or, at least, very) similar to $\boldsymbol{p}^{\mathrm{O}}$. More formally, according to the NNS, the reconstructed pose $\widehat{\boldsymbol{p}}^{\mathrm{O}}$ is estimated as follows:

$$\widehat{\boldsymbol{p}}^{\mathrm{O}} = \overline{\boldsymbol{p}}^{\mathrm{O}}_{\widehat{n}} \tag{5.13}$$

where

$$\widehat{n} = \underset{n \in \{1,2,\dots,N\}}{\arg\min}\ d\left(\boldsymbol{f}, \overline{\boldsymbol{f}}_n\right) \tag{5.14}$$

where $d(\cdot,\cdot)$ is a properly defined distance metric—more details will be given in Subsection 5.5.3.

Note that $\overline{\boldsymbol{P}}^{\mathrm{O}}$ may not necessarily contain a pose exactly similar to $\boldsymbol{p}^{\mathrm{O}}$. Instead, $\boldsymbol{p}^{\mathrm{O}}$ may be "close" to many training poses and, possibly, can be expressed as a weighted interpolation between closest training poses. Therefore, a better way to proceed is to search (still in the feature space) for the k "nearest neighbors" of $\boldsymbol{f}$ in $\overline{\boldsymbol{F}}$ (i.e., the k training feature vectors whose distances from $\boldsymbol{f}$ are smallest). Eventually, denoting the ordered set of indexes of the k nearest neighbors of $\boldsymbol{f}$ as $\mathscr{K}$, the estimated pose $\widehat{\boldsymbol{p}}^{\mathrm{O}}$ can then be expressed as follows:

$$\widehat{\boldsymbol{p}}^{\mathrm{O}} = \sum_{n=1}^{k} w_{\mathscr{K}(n)} \cdot \overline{\boldsymbol{p}}^{\mathrm{O}}_{\mathscr{K}(n)} \tag{5.15}$$

where $w_{\mathscr{K}(n)}$ is the interpolation weight of the n-th nearest neighbor and $\sum_{n=1}^{k} w_{\mathscr{K}(n)} = 1$. A common policy in defining the weights is to weigh more the neighbors in correspondence to which $d(\boldsymbol{f}, \overline{\boldsymbol{f}}_n)$ is smallest and viceversa (e.g., using the inverse of the distance). For ease of clarity, a block diagram representation of an NNS-based motion reconstruction algorithm is shown in Figure 5.1.

5.5 NNS-based Human Motion Reconstruction: the Proposed Framework

So far, the general idea behind a NNS-based human motion reconstruction has been described. The proposed framework substantiates the general idea introduced in Section 5.4 mainly in terms of the following two aspects.

- We use a training dataset of shapes, rather than poses. This improves the *flexibility* of the motion reconstruction algorithm, especially when a poor training has been performed and few poses are available in the training dataset. Indeed, two poses can still have very similar shapes, while having very different absolute orientation of the body. For instance, turning around (while keeping the same pose) will change the absolute orientation but will keep the shape unchanged. Note, however, that the considered features should thus properly characterize shapes (and not poses) and should be handled consistently at runtime.

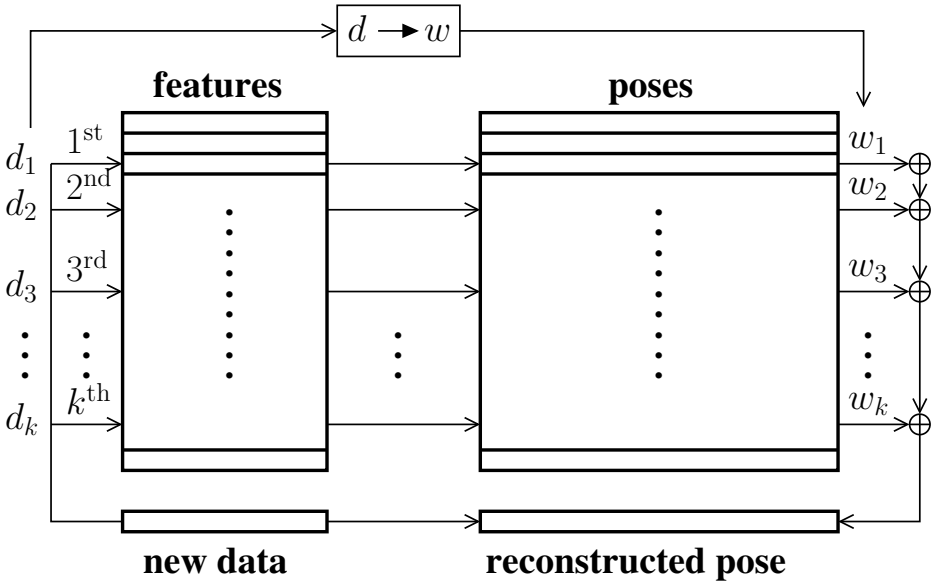


Figure 5.1: Block diagram representation of an NNS-based motion reconstruction algorithm.

- More than a type of feature is considered (namely, three types). This improves the *robustness* of the motion reconstruction algorithm. Since the information related to a single type of features is typically partial and incomplete, the use of more types of features allows to better constrain the unambiguous motion reconstruction process. Note, however, that different features require different distance metrics and more efforts to integrate them in a common framework.

In the following, the proposed framework will be thus thoroughly described, trying to focus on the solutions used to address the previous issues.

5.5.1 Placement of Sensor Devices

In the proposed motion reconstruction framework, a total of $B = 5$ sensor devices have been considered, placed on the pelvis, on the two wrists, and on the two ankles, respectively. For ease of clarity, in Figure 5.2 the placement of sensor devices on the

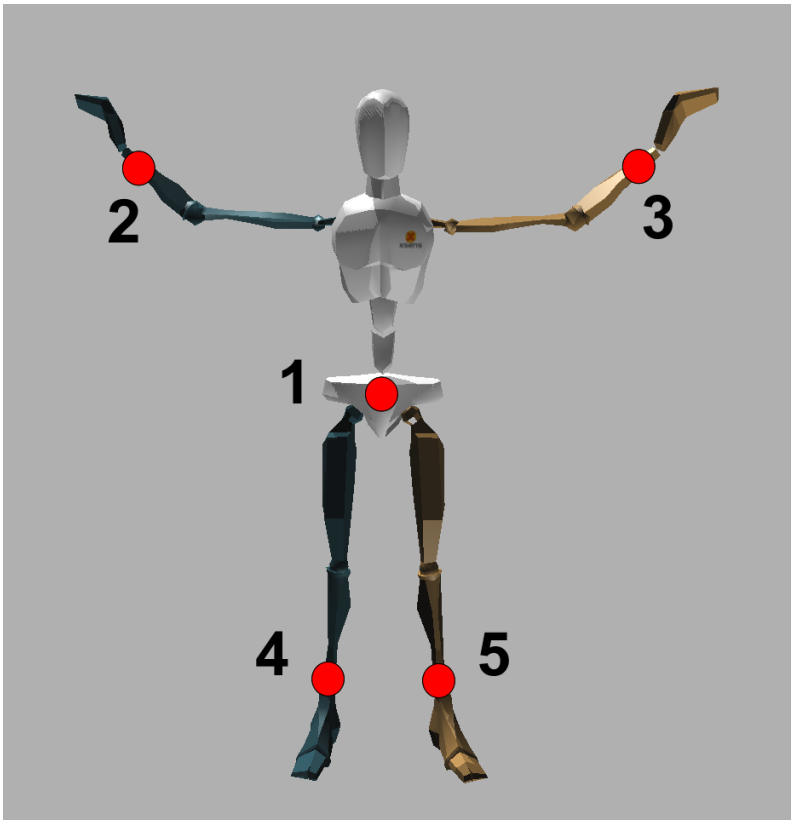


Figure 5.2: Placement of $B = 5$ sensor devices on the user body. An avatar composed of $S = 23$ rigid body segments has been used to virtually represent the user body.

avatar of a user body is shown ($S = 23$ body segments are considered). In particular, the pelvis is chosen to be the *reference body segment* (therefore, the relative orientations of the remaining $S-1$ body segments will be expressed with respect to it) and, similarly, the sensor device placed on the pelvis will be called the *reference sensor device*—more details about this choice will be later given in Section 5.7. Note that this sensor device configuration has been heuristically chosen in order (i) to favor the comfort of the user and (ii) to extract the most significant kinematic information from the user body (it is quite intuitive to assume that kinematic information extracted

from distal body segments, such as the hands and the feet, will give more valuable information for the reconstruction of the full body pose of the user than that extracted from proximal body segments, such as the upper arms, the thighs, or even the chest). Nevertheless, the following derivation is presented in general terms and different sensor device configurations and different choices of the reference body segment can be considered, without loss of generality.

5.5.2 Feature Description

Three types of features have been considered for the proposed framework: relative orientations, relative ranges, and relative heights. In the following, these features are properly defined, specifying how they can be computed from the orientation and/or position data of a full body pose.

- *Relative orientations*: the relative orientation of each considered sensor device (excluding the reference one) expressed with respect to that of the reference one. More formally, the feature vector of relative orientations is defined as follows:

$$\boldsymbol{f}^{\mathrm{RO}} \triangleq \begin{bmatrix} \boldsymbol{q}^{j_1,\sim} & \boldsymbol{q}^{j_2,\sim} & \dots & \boldsymbol{q}^{j_{B-1},\sim} \end{bmatrix} \tag{5.16}$$

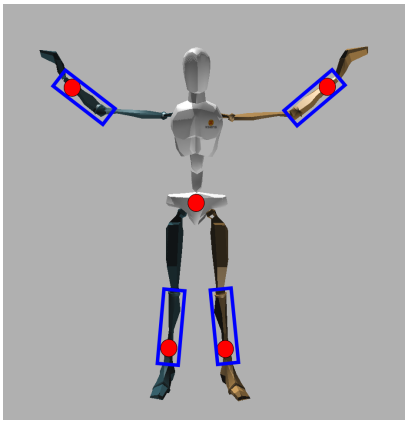
  where $\left\{\boldsymbol{q}^{j_b,\sim}\right\}_{b=1}^{B-1} \subset \left\{\boldsymbol{q}^{s,\sim}\right\}_{s=1}^{S-1} = \boldsymbol{p}^{\mathrm{O},\sim}$, i.e., the feature vector contains the relative orientations of $B-1$ (out of a total of $S-1$) body segments. Since $\boldsymbol{q}^{j_b,\sim}$, $b \in \{1,2,\dots,B-1\}$, is a quaternion, $\boldsymbol{f}^{\mathrm{RO}}$ is a $(4\cdot(B-1))$-dimensional vector. For the sake of clarity, in Figure 5.3 (a) the $B-1=4$ body segments for which the relative orientations are computed in our experimental analysis are highlighted.

- *Relative ranges*: the physical distance between each possible pair of the considered sensor devices. More formally, the feature vector of relative ranges is defined as follows:

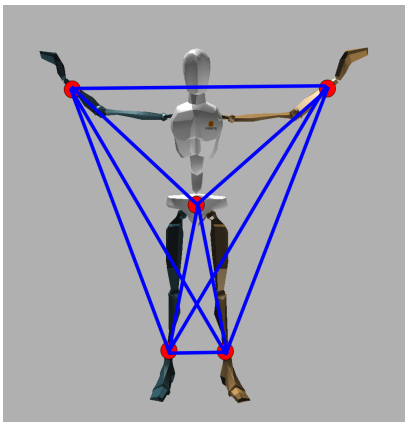
$$\boldsymbol{f}^{\mathrm{RR}} \triangleq \begin{bmatrix} r^1 & r^2 & \dots & r^{\binom{B}{2}} \end{bmatrix} \tag{5.17}$$

  where $\binom{B}{2}$ is the number of (non-ordered) pairs and r^b is the Euclidean distance between the b-th pair of devices ($b \in \left\{1,2,\dots,\binom{B}{2}\right\}$). The generic distance

(a)



(b)



(c)

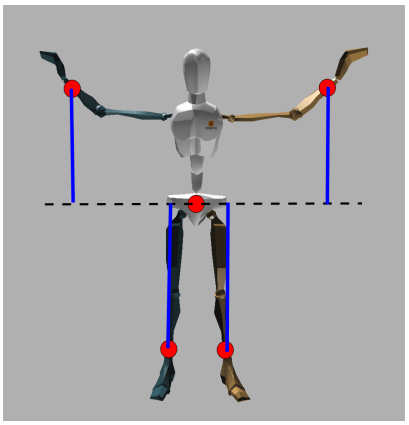


Figure 5.3: Graphically intuitive representation of the considered features, assuming that the 5 sensor devices shown in Figure 5.2 are available: (a) relative orientations, (b) relative ranges, and (c) relative heights.

between a pair of sensor devices, say i and j, can be then computed as follows:

$$r = \sqrt{\left(c_x^i - c_x^j\right)^2 + \left(c_y^i - c_y^j\right)^2 + \left(c_z^i - c_z^j\right)^2} \tag{5.18}$$

where $\left(c_x^i, c_y^i, c_z^i\right)$ and $\left(c_x^j, c_y^j, c_z^j\right)$ are the three-dimensional physical coordinates of, respectively, the i-th and the j-th sensor devices. Note that $\boldsymbol{f}^{\mathrm{RR}}$ is a $\binom{B}{2}$-dimensional vector. For ease of clarity, in Figure 5.3 (b) the $\binom{B}{2} = \binom{5}{2} = 10$ relative ranges considered in our experimental analysis are shown.

- *Relative heights*: the (signed) distance of each considered sensor device (excluding the reference one) with respect to the horizontal plane (i.e., the plane perpendicular to the gravity vector direction) defined as the one where the reference sensor device lies. More formally, the feature vector of relative heights can be defined as follows:

$$\boldsymbol{f}^{\mathrm{RH}} \triangleq \begin{bmatrix} h^{j_1,\sim} & h^{j_2,\sim} & \dots & h^{j_{B-1},\sim} \end{bmatrix} \tag{5.19}$$

where the relative height of the j_b-th sensor device, $b \in \{1,2,\dots,B-1\}$, is computed as follows:

$$h^{j_b,\sim} = c_z^{j_b} - c_z^{\sim} \tag{5.20}$$

where $c_z^{j_b}$ is the z coordinate of the j_b-th sensor device and $c_z^{\sim}$ is the z coordinate of the reference sensor device. Note that $\boldsymbol{f}^{\mathrm{RH}}$ is a $(B-1)$-dimensional vector. For ease of clarity, in Figure 5.3 (c) the relative heights of the $B-1=4$ devices considered in our experimental analysis are shown.

We remark that, while the relative orientations and ranges depend only on the *shape* of the body but not on its actual absolute orientation, different absolute orientations of the body lead to different values of relative heights (since they are computed along the vertical direction). Therefore, since the proposed motion reconstruction framework is based on a training dataset of shapes (rather than poses), relative orientations and ranges can be computed in the training phase, whereas relative heights are only computed at runtime, when the actual absolute orientation of the body (i.e.,

that of the reference body segment) is known. This will be better understood in Subsection 5.5.3.

Finally, unlike in the training phase, where the full body poses are known and the previously described features can be easily computed (exploiting the pose dataset), at runtime it is not necessarily true, since five devices are only available and the full body pose cannot be unambiguously estimated. Therefore, at runtime, different types of sensing technologies are necessary to compute the features. In particular, (i) relative orientations can be computed using inertial/magnetic sensors (namely, accelerometers, gyroscopes, and magnetometers), (ii) relative ranges can be computed using UWB ranging, and (iii) relative heights can be computed using barometers.

5.5.3 The Algorithm

Training

The proposed algorithm assumes that a training phase has been properly performed where two training datasets have been built: a dataset of features and a dataset of shapes. In particular, the dataset of shapes $\overline{\boldsymbol{P}}^{\mathrm{O},\sim}$ is defined as in Equation (5.11), whereas the dataset of features is defined as follows:

$$\overline{\boldsymbol{F}} = \begin{bmatrix} \overline{\boldsymbol{F}}^{\mathrm{RO}} & \overline{\boldsymbol{F}}^{\mathrm{RR}} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{f}}_1^{\mathrm{RO}} & \overline{\boldsymbol{f}}_1^{\mathrm{RR}} \\ \overline{\boldsymbol{f}}_2^{\mathrm{RO}} & \overline{\boldsymbol{f}}_2^{\mathrm{RR}} \\ \vdots & \vdots \\ \overline{\boldsymbol{f}}_N^{\mathrm{RO}} & \overline{\boldsymbol{f}}_N^{\mathrm{RR}} \end{bmatrix}. \tag{5.21}$$

Note that, as already mentioned in Subsection 5.5.2, since our feature dataset is used to characterize a dataset of shapes and relative heights depend instead also on the actual absolute orientation of the body, only relative orientations and ranges are already included in the training dataset of features. Relative heights will be instead computed at runtime, on the basis of the updated absolute orientation of the body, i.e., that newly measured from the reference body segment (namely, $\boldsymbol{q}^{\sim}$).

Runtime

At runtime, when a new pose has to be reconstructed using only the B available sensor devices, the following assumptions are then expedient.

- The absolute orientation $\boldsymbol{q}^{\sim}$ of the reference sensor device (and, thus, of the reference body segment), expressed with respect to the Earth reference system, is known.
- A (new) feature vector $\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{RO}} & \boldsymbol{f}^{\mathrm{RR}} & \boldsymbol{f}^{\mathrm{RH}} \end{bmatrix}$ is computed using the available sensor devices. As indicated in Subsection 5.5.2, this can be done by using different sensor technologies (such as inertial sensors, UWB, and barometers).

At this point, the NNS must be performed (as described in Subsection 5.4), comparing the newly measured $\boldsymbol{f}$ with the feature vectors in $\overline{\boldsymbol{F}}$, in order to select the nearest neighbors in the training dataset of features and, correspondingly, in that of shapes. To this end, in Equation (5.14), we have shown that a proper metric has to be used to measure feature vectors' distances. However, since three different types of features are now considered, a different definition of distance must be considered for each feature type.

Runtime—Step 1: Computation of Distances

Concerning *relative orientations*, a distance vector $\boldsymbol{d}^{\mathrm{RO}}$ can be defined so that its n-th element ($n \in \{1,2,\ldots,N\}$) can be expressed as follows:

$$d_n^{\mathrm{RO}} \triangleq d^{\mathrm{RO}}\left(\boldsymbol{f}^{\mathrm{RO}}, \overline{\boldsymbol{f}}_n^{\mathrm{RO}}\right) \triangleq \sum_{i=1}^{B-1} \frac{\alpha\left(\boldsymbol{q}^{j_i,\sim}, \overline{\boldsymbol{q}}_n^{j_i,\sim}\right)}{B-1} \tag{5.22}$$

where: $\boldsymbol{f}^{\mathrm{RO}}$ (runtime) and $\overline{\boldsymbol{f}}_n^{\mathrm{RO}}$ (training) are the feature vectors containing the relative orientations and are defined as in Equation (5.16); $d^{\mathrm{RO}}\colon \mathbb{R}^{4\cdot(B-1)} \times \mathbb{R}^{4\cdot(B-1)} \to \mathbb{R}$; and $\alpha\colon \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}$ computes the smallest angle between two quaternions (i.e., the smallest angle between two orientations). In particular, α can be computed as follows [38]:

$$\alpha\left(\boldsymbol{q}^{j_i,\sim}, \overline{\boldsymbol{q}}_n^{j_i,\sim}\right) = 2 \cdot \arccos\left(\left[\left(\boldsymbol{q}^{j_i,\sim}\right)^{-1} \otimes \left(\overline{\boldsymbol{q}}_n^{j_i,\sim}\right)\right]_1\right) \tag{5.23}$$

where: $\otimes$ is the quaternion multiplication function; $(\cdot)^{-1}$ is the quaternion inverse function (equivalent to the conjugate if the quaternion is a unit vector); and $[\cdot]_1$ extracts the vector first component.

Concerning *relative ranges*, a distance vector $\boldsymbol{d}^{\mathrm{RR}}$ can be defined so that its n-th element can be expressed as follows:

$$d_n^{\mathrm{RR}} \triangleq d^{\mathrm{RR}}\left(\boldsymbol{f}^{\mathrm{RR}}, \overline{\boldsymbol{f}}_n^{\mathrm{RR}}\right) \triangleq \sqrt{\sum_{i=1}^{\binom{B}{2}} \left(r^i - \overline{r}_n^i\right)^2} \tag{5.24}$$

where $\boldsymbol{f}^{\mathrm{RR}}$ (runtime) and $\overline{\boldsymbol{f}}_n^{\mathrm{RR}}$ (training) are the feature vectors containing the relative ranges and are defined as in Equations (5.17) and (5.18). Note that $d^{\mathrm{RR}}\colon \mathbb{R}^{\binom{B}{2}} \times \mathbb{R}^{\binom{B}{2}} \to \mathbb{R}$ is actually the Euclidean distance between $\binom{B}{2}$-dimensional vectors.

Finally, concerning *relative heights*, a distance vector $\boldsymbol{d}^{\mathrm{RH}}$ can be defined so that its n-th element can be expressed as follows:

$$d_n^{\mathrm{RH}} \triangleq d^{\mathrm{RH}}\left(\boldsymbol{f}^{\mathrm{RH}}, \overline{\boldsymbol{f}}_n^{\mathrm{RH}}\right) \triangleq \sqrt{\sum_{i=1}^{B-1} \left(h^{j_i,\sim} - \overline{h}_n^{j_i,\sim}\right)^2} \tag{5.25}$$

where $\boldsymbol{f}^{\mathrm{RH}}$ (runtime) and $\overline{\boldsymbol{f}}_n^{\mathrm{RH}}$ (training) are the feature vectors containing the relative heights and are defined as in Equations (5.19) and (5.20) and, as $d^{\mathrm{RR}}(\cdot,\cdot)$ in Equation (5.24), $d^{\mathrm{RH}}\colon \mathbb{R}^{B-1} \times \mathbb{R}^{B-1} \to \mathbb{R}$ is the Euclidean distance, between $(B-1)$-dimensional vectors. Given the n-th shape contained in the training dataset (i.e., $\boldsymbol{p}_n^{\mathrm{O},\sim}$) and the newly measured reference body segment orientation (i.e., $\boldsymbol{q}^{\sim}$), the (training) relative height $\overline{h}_n^{j_i,\sim}$ of Equation (5.25) can be obtained as follows:

$$\overline{h}_n^{j_i} = \overline{c}_z^{j_i} - \overline{c}_z^{\sim} = \left[\boldsymbol{p}_n^{\mathrm{P},\sim\sim}\right]_z^{j_i} - \left[\boldsymbol{p}_n^{\mathrm{P},\sim\sim}\right]_z^{\sim} \tag{5.26}$$

with

$$\boldsymbol{p}_n^{\mathrm{P},\sim\sim} = g^{\mathrm{FK}}\left(g^{\mathrm{R}}\left(\boldsymbol{q}^{\sim}, \boldsymbol{p}_n^{\mathrm{O},\sim}\right)\right) \tag{5.27}$$

where: $\overline{c}_z^{j_i}$ and $\overline{c}_z^{\sim}$ are the z coordinates of, respectively, the j_i-th and the reference sensor devices; $[\cdot]_z^{j_i}$ and $[\cdot]_z^{\sim}$ extract the z coordinates of, respectively, the j_i-th and the reference body segments; $g^{\mathrm{R}}\left(\boldsymbol{q}^{\sim}, \boldsymbol{p}_n^{\mathrm{O},\sim}\right)\colon \mathbb{R}^4 \times \mathbb{R}^{4\cdot(S-1)} \to \mathbb{R}^{4\cdot(S-1)}$ is a function

which rotates every orientation contained in $\boldsymbol{p}_n^{\mathrm{O},\sim}$ according to $\boldsymbol{q}^{\sim}$ (i.e., a novel training pose is created by rotating the training shape according to the newly measured $\boldsymbol{q}^{\sim}$); and g^{FK} is a forward kinematics function defined as in Equations (5.9) and (5.4).

Runtime—Step 2: Computation of Weights

Since in the following interpolation process, according to Equation (5.15), we are interested in weights (rather than distances) and, more specifically, in a unique vector of weights, the previously described distance vectors may be joined together and used to obtain corresponding vectors of weights. However, the previously described distance vectors (i.e., $\boldsymbol{d}^{\mathrm{RO}}$, $\boldsymbol{d}^{\mathrm{RR}}$, and $\boldsymbol{d}^{\mathrm{RH}}$) are very heterogeneous and range from 0 to undefined maxima, whose exact values strongly depend on the considered feature. To this end, for each $\boldsymbol{d}^{\mathrm{X}}$, $\mathrm{X} \in \{\mathrm{RO},\mathrm{RR},\mathrm{RH}\}$, a corresponding vector of weights $\boldsymbol{w}^{\mathrm{X}}$ (indicated depending on the related distance vector as $\boldsymbol{w}^{\mathrm{RO}}$, $\boldsymbol{w}^{\mathrm{RR}}$, or $\boldsymbol{w}^{\mathrm{RH}}$) can be defined so that its n-th element ($n \in \{1,2,\ldots,N\}$) can be expressed as follows:

$$w_n^{\mathrm{X}} = \frac{\max\left(\boldsymbol{d}^{\mathrm{X}}\right) - d_n^{\mathrm{X}}}{\sum_{\ell=1}^{N}\left(\max\left(\boldsymbol{d}^{\mathrm{X}}\right) - d_\ell^{\mathrm{X}}\right)}. \tag{5.28}$$

Note that, regardless of the considered feature, $\boldsymbol{w}^{\mathrm{X}}$ is now normalized (namely, $w_n^{\mathrm{X}} \in [0,1]$, $\forall n$) and, as opposed to the original distance vectors, assigns larger weights to elements with larger distance values and, viceversa, smaller weights to elements with smaller distance values.

At this point, the previously computed weight vectors may be consistently combined together, exploiting a logarithmic transformation, to form an overall "new" distance vector $\boldsymbol{d}$ whose n-th element is defined as follows:

$$d_n \triangleq -\gamma^{\mathrm{RO}} \cdot \log w_n^{\mathrm{RO}} - \gamma^{\mathrm{RR}} \cdot \log w_n^{\mathrm{RR}} - \gamma^{\mathrm{RH}} \cdot \log w_n^{\mathrm{RH}} \tag{5.29}$$

where γ^{RO}, γ^{RR}, and γ^{RH} can assume (continuous) values between 0 and 1 and are used to tune the contribution of, respectively, relative orientations, relative ranges, and relative heights to the newly defined distance vector $\boldsymbol{d}$. Note that $\boldsymbol{d}$, which belongs now to $[0,+\infty]$, is again a "distance" vector and its elements with lowest values are the nearest neighbors.

Now that a unique distance vector $\boldsymbol{d}$ is available, all the elements of the training datasets can be sorted according to it. In particular, the nearest neighbor is the one with the minimum value of distance and so on for increasing values of distance. Even if one may think that this newly computed distance vector could be again used, by applying Equation (5.28), in order to directly get weights with which weighing the contribution of each (nearest) pose in the interpolation step, it is important to notice that, due to the fact that d will likely contain very high distance values (corresponding to very unlikely poses), very similar weights will be assigned to the k nearest poses (if $k \ll N$), making almost useless the weighing process. To this end, if the ordered set of indexes of the first k nearest neighbors is denoted as $\mathscr{K}$, a new (more compact) version of $\boldsymbol{d}$ (which will be denoted as $\boldsymbol{d}'$), composed only of the first k nearest neighbors, can be instead considered at this stage, with the i-th element chosen as follows:

$$d'_i = d_{\mathscr{K}(i)} \qquad i \in \{1,2,\ldots,k\} \tag{5.30}$$

where $\mathscr{K}(i)$ denotes the i-th element of set $\mathscr{K}$. Equation (5.28) can be now used, replacing N with k, to compute from $\boldsymbol{d}'$ a (new) weight vector $\boldsymbol{w}'$ containing the weights that have to be given to the first k nearest poses.

Runtime—Step 3: Weighted Interpolation

The reconstructed shape $\widehat{\boldsymbol{p}}^{\mathrm{O}}$ can be thus estimated as follows:

$$\widehat{\boldsymbol{p}}^{\mathrm{O},\sim} = \sum_{i=1}^{k} w'_i \cdot \overline{\boldsymbol{p}}^{\mathrm{O},\sim}_{\mathscr{K}(i)}. \tag{5.31}$$

Note however that, since the interpolation step (Equation (5.31)) does not necessarily lead to unit quaternions,[4] after applying Equation (5.31) a re-normalization step is necessary, where each quaternion in $\widehat{\boldsymbol{p}}^{\mathrm{O},\sim}$ is divided by its norm in order to become a unit vector. For ease of notational simplicity, we do not explicitly indicate this normalization step. At this point, the reconstructed pose $\widehat{\boldsymbol{p}}^{\mathrm{O}}$ can be derived by rotating the reconstructed shape of $\boldsymbol{q}^{\sim}$ as follows:

$$\widehat{\boldsymbol{p}}^{\mathrm{O}} = g^{\mathrm{R}}\left(\boldsymbol{q}^{\sim}, \widehat{\boldsymbol{p}}^{\mathrm{O},\sim}\right). \tag{5.32}$$

[4]Note that a quaternion must be a unit vector to represent a three-dimensional rotation.

Finally, the estimated component orientations in $\widehat{\boldsymbol{p}}^{\mathrm{O}}$ corresponding to the body segments equipped with the sensor devices are further corrected by replacing them with those directly measured and estimated by the available sensor devices.

5.6 Experimental Set-up

5.6.1 Experimental Testbed

In order to test the system performance, a commercially available inertial motion capture suit, i.e., Xsens MVN Awinda [4] (i.e., the wireless version of Xsens MVN [3, 16]), has been used. Xsens MVN Awinda is a full body Lycra suit equipped with 17 sensor devices. Each sensor device is provided with triaxial accelerometer, gyroscope, magnetometer, and a barometer. Each device can communicate wirelessly with an external base station using a proprietary radio protocol. The sensor devices are attached to the suit in order to be rigidly fixed to the user body segments when the suit is worn and are placed in correspondence to the following 17 body segments: the pelvis (1); the thighs (2); the lower legs (2); the feet (2); the chest (1); the shoulders (2); the upper arms (2); the forearms (2); the hands (2); and the head (1). Even though only 17 sensor devices are used, Xsens MVN Awinda is able to estimate, with errors lower than 1^{o}, the orientations of 23 body segments (6 of which, i.e., the toes, the neck, and three segments of the spine, are just virtually reconstructed according to a proper biomechanical model). Furthermore, the positions of all the body segments are also given. The motion capture data are recorded at 30 Hz.

Due to its great accuracy and ease of use, Xsens MVN Awinda has been used to collect all the data involved in the experimental analysis. More specifically, the estimated full body pose of the user has been used as the ground truth for our performance analysis. The "ideal" performance of a motion reconstruction system composed of a limited number of sensor devices (such as the system proposed in this chapter) is evaluated taking into account only a proper subset (i.e., corresponding to the selected devices) of all the available data.

Since this chapter presents a proof-of-concept investigation of the proposed human motion reconstruction system, only one user has been considered. Furthermore,

relative orientations have been directly extracted from the suit, whereas ideally simulated values of relative ranges and relative heights (computed from the position data provided by the suit) have been used. Note, however, that the use of simulated relative ranges and heights is just expedient for such preliminary investigation. In a practical system implementation (as already discussed in Subsection 5.5.2) those features would come indeed from inertial/magnetic sensors, UWB nodes, and barometers.

A dataset of poses has been built, using Xsens MVN Awinda, comprising different types of motion tasks, including both common movements, sport-related movements, and unusual movements. The complete dataset is composed of about 30 minutes of motion capture data, recorded in different sessions. Note that, since the motion capture data are recorded at 30 Hz, the complete dataset is composed of about $N = 50000 \div 60000$ poses.

To be more precise, the proposed algorithm has been tested on the following *motion tasks* (sometimes more than one repetition per task): walk, run, movements of upper limbs, ADL, movements of one leg, jumps over one leg, crouch, crawl, basketball, golf, baseball, tennis, fencing, jumping jacks. For the sake of generalization of performance analysis, the algorithm has been tested using a leave-one-out approach where each considered motion task has been, in turn, manually extracted from the complete dataset and tested, leading to the generation of different testing (i.e., the elements of the dataset associated with the considered motion task) and training (i.e., the remaining elements) datasets for each considered motion task. Note, however, that the training and testing datasets are always distinct and this process is actually performed in order to obtain unbiased system performance. In addition, in order to have a more general indication of the system performance, the independently-evaluated performances for all motion tasks have been also properly averaged together, obtaining a single performance evaluation.

For the purpose of this preliminary experimental analysis, the impact of the following variables on the system performance has been investigated: the value of k (i.e., the number of considered nearest neighbors in the NNS and following weighted interpolation); the considered features (and, correspondingly, the considered sensor technologies); and the number and placement of the considered sensor devices.

Concerning the choice of k, values which are powers of two have been considered.[5] Since the normalization in Equation (5.28) gives a zero weight to the farthest neighbors, the following values of k have been considered: $\mathscr{K} = \{1,3,5,9,17,33,65\}$. Note that the case of $k = 1$ corresponds to the case where only the nearest neighbor is taken into account and the weighted interpolation is thus not performed.

Concerning the considered *features*, in order to test different combinations of features, the weights γ^{RO},γ^{RR}, and γ^{RH} used in Equation (5.29) have been each time set to 1 or 0, depending whether the corresponding feature is considered (1) or not (0). Relative orientations are always considered. In addition, combinations of the other two features are also investigated. In particular, the following combinations of features have been considered: (i) relative orientations ($\gamma^{\mathrm{RO}} = 1$,$\gamma^{\mathrm{RR}} = 0$, and $\gamma^{\mathrm{RH}} = 0$); (ii) relative orientations and relative heights ($\gamma^{\mathrm{RO}} = 1$,$\gamma^{\mathrm{RR}} = 0$, and $\gamma^{\mathrm{RH}} = 1$); (iii) relative orientations and relative ranges ($\gamma^{\mathrm{RO}} = 1$,$\gamma^{\mathrm{RR}} = 1$, and $\gamma^{\mathrm{RH}} = 0$); and (iv) relative orientations, ranges, and heights ($\gamma^{\mathrm{RO}} = 1$,$\gamma^{\mathrm{RR}} = 1$, and $\gamma^{\mathrm{RH}} = 1$).

Finally, concerning the *number and placement of the sensor devices*, besides the initial configuration shown in Figure 5.2, other 14 configurations of sensor devices have been investigated. In particular, the pelvis sensor device is always considered together with all the combinations of the other four devices. Therefore, configurations of 5, 4, 3, and 2 sensor devices are taken into account.

5.6.2 Performance Metrics

A suitable performance metric for a motion reconstruction algorithm should aim at quantifying the realism and naturalness of the reconstructed motion, in addition to the similarity between the actual motion and the reconstructed one. Since human perception is highly subjective among different individuals, determining a fair way to evaluate the system performance is not an easy task [130].

One possible way to quantify the similarity of two poses is to compute the average Euclidean distance between joint positions (which we denote as *position error*). More

[5] We remark, however, that there is no formal reason behind this choice and values which are not powers of two could be selected.

formally, the *position error* is defined as follows:

$$\begin{aligned} e^{\mathrm{P}} &\triangleq \frac{\sum_{i=1}^{S} d^{\mathrm{E}}\left(\left[\widehat{\boldsymbol{p}}^{\mathrm{P}}\right]^{i}, \left[\boldsymbol{p}^{\mathrm{P}}\right]^{i}\right)}{S} \\ &= \frac{\sum_{i=1}^{S} d^{\mathrm{E}}\left(\left[g^{\mathrm{FK}}\left(\widehat{\boldsymbol{p}}^{\mathrm{O}}\right)\right]^{i}, \left[g^{\mathrm{FK}}\left(\boldsymbol{p}^{\mathrm{O}}\right)\right]^{i}\right)}{S} \\ &= \frac{\sum_{i=1}^{S} \sqrt{\left(\widehat{c}_x^i - c_x^i\right)^2 + \left(\widehat{c}_y^i - c_y^i\right)^2 + \left(\widehat{c}_z^i - c_z^i\right)^2}}{S} \end{aligned} \tag{5.33}$$

where: $\widehat{\boldsymbol{p}}^{\mathrm{O}}$ and $\boldsymbol{p}^{\mathrm{O}}$ are, respectively, the reconstructed and the actual poses of the user (expressed with body segment orientations); $\widehat{\boldsymbol{p}}^{\mathrm{P}}$ and $\boldsymbol{p}^{\mathrm{P}}$ are, respectively, the reconstructed and the actual poses of the user (expressed with body segment positions); S is the number of body segments; $[\cdot]^i$ extracts the x, y, and z coordinates of the i-th body segment ($i \in \{1,2,\ldots,S\}$); $d^{\mathrm{E}}(\cdot)$ computes the Euclidean distance; g^{FK} is the forward kinematics function defined in Equations (5.9) and (5.4); $\widehat{c}_x^i$, $\widehat{c}_y^i$, and $\widehat{c}_z^i$ are, respectively, the x, y, and z coordinates of the i-th body segment for the reconstructed pose; and c_x^i, c_y^i, and c_z^i are, respectively, the x, y, and z coordinates of the i-th body segment for the actual pose.

On the other hand, if one wants to measure the similarity of two poses in terms of the average error in the body segment orientations, an *orientation error* can be computed as follows:

$$\begin{aligned} e^{\mathrm{O}} &\triangleq \frac{\sum_{i=1}^{S} \alpha\left(\left[\widehat{\boldsymbol{p}}^{\mathrm{O}}\right]^{i}, \left[\boldsymbol{p}^{\mathrm{O}}\right]^{i}\right)}{S} \\ &= \frac{\sum_{i=1}^{S} \alpha\left(\widehat{\boldsymbol{q}}^{i}, \boldsymbol{q}^{i}\right)}{S} \end{aligned} \tag{5.34}$$

where: $\widehat{\boldsymbol{q}}^i$ and $\boldsymbol{q}^i$ are the orientation of the i-th body segment (expressed as a quaternion) for, respectively, the reconstructed and the actual poses; $[\cdot]^i$ extracts the orientation (expressed as a quaternion) of the i-th body segment; and $\alpha(\cdot)$ is the smallest angle between two quaternions (i.e., the smallest angle between two orientations) and can be computed as shown in Equation (5.23).

Both Equations (5.33) and (5.34), even if suitable to be used as performance metrics, are not sufficient to thoroughly evaluate the system performance. Indeed, the following observations can be carried out: (i) Equation (5.33) conveys better the idea of human perception of similarity between two poses (because it only observes the physical distance of the body joints), but does not allow to evaluate errors in the body segment orientations; (ii) Equation (5.34) is more precise at evaluating the actual differences between two poses but, since the three-dimensional physical coordinates of the human body are not involved in the error evaluation, it is farther from reflecting the human perception of similarity. Furthermore, since they involve different measurement units, the two previous metrics cannot be averaged together and, thus, a single measurement of the system performance cannot be derived.

To this end, in this chapter we propose a third novel way to measure the system performance, which is particularly suitable to be used in the context of an NNS-based approach. Specifically, the proposed algorithm actually reconstructs the user pose by choosing among all the poses contained in the training dataset. More precisely, the body shape is chosen and is properly rotated by the actual orientation of the reference body segment (i.e., the pelvis in our case). Therefore, it is reasonable to express the system performance in terms of its efficacy in choosing the best pose among the training poses. Indeed, even if this does not represent an absolute way of measuring system performance, it is also true that the construction of an exhaustive training dataset does not necessarily pertain to the algorithm, whose goal is instead to choose the most similar pose among those available. To this end, for each new pose that has to be reconstructed, we evaluate the errors that would be made by choosing the best and the worst poses in the training dataset (where the errors can be measured in terms of both position and orientation errors) and we relate these errors with that achieved with the actually reconstructed pose. More formally, given the actual reference body segment orientation $\boldsymbol{q}^{\sim}$, the minimum and the maximum achievable position errors can be computed, respectively, as follows:

$$m^{\mathrm{P}} = \min_{n\in\{1,\ldots,N\}} e^{\mathrm{P}}\left(\widehat{\boldsymbol{p}}^{P}, g^{\mathrm{FK}}\left(g^{\mathrm{R}}\left(\boldsymbol{q}^{\sim}, \boldsymbol{p}_n^{\mathrm{O},\sim}\right)\right)\right) \tag{5.35}$$

$$M^{\mathrm{P}} = \max_{n\in\{1,\ldots,N\}} e^{\mathrm{P}}\left(\widehat{\boldsymbol{p}}^{P}, g^{\mathrm{FK}}\left(g^{\mathrm{R}}\left(\boldsymbol{q}^{\sim}, \boldsymbol{p}_n^{\mathrm{O},\sim}\right)\right)\right) \tag{5.36}$$

where: $e^{\mathrm{P}}(\cdot,\cdot)$ computes the position error according to Equation (5.33); $\boldsymbol{p}_n^{\mathrm{O},\sim}$ is the n-th shape ($n \in \{1,2,\ldots,N\}$) contained in the training dataset; and $g^{\mathrm{R}}\left(\boldsymbol{q}^{\sim},\boldsymbol{p}_n^{\mathrm{O},\sim}\right)$ is a function which rotates every orientation contained in $\boldsymbol{p}_n^{\mathrm{O},\sim}$ according to $\boldsymbol{q}^{\sim}$. Similarly, the minimum and the maximum achievable orientation errors can be computed, respectively, as follows:

$$m^{\mathrm{O}} = \min_{n\in\{1,\ldots,N\}} e^{\mathrm{O}}\left(\widehat{\boldsymbol{p}}^{O}, g^{\mathrm{R}}\left(\boldsymbol{q}^{\sim},\boldsymbol{p}_n^{\mathrm{O},\sim}\right)\right) \tag{5.37}$$

$$M^{\mathrm{O}} = \max_{n\in\{1,\ldots,N\}} e^{\mathrm{O}}\left(\widehat{\boldsymbol{p}}^{O}, g^{\mathrm{R}}\left(\boldsymbol{q}^{\sim},\boldsymbol{p}_n^{\mathrm{O},\sim}\right)\right) \tag{5.38}$$

where $e^{\mathrm{O}}(\cdot,\cdot)$ computes the orientation error according to Equation (5.34).

Given m^{P}, M^{P}, m^{O}, and M^{O}, relative position and orientation errors can be defined, respectively, as follows:

$$e^{\%,\mathrm{P}} \triangleq \frac{M^{\mathrm{P}} - e^{\mathrm{P}}}{M^{\mathrm{P}} - m^{\mathrm{P}}} \times 100 \tag{5.39}$$

$$e^{\%,\mathrm{O}} \triangleq \frac{M^{\mathrm{O}} - e^{\mathrm{O}}}{M^{\mathrm{O}} - m^{\mathrm{O}}} \times 100. \tag{5.40}$$

At this stage, since both $e^{\%,\mathrm{P}}$ and $e^{\%,\mathrm{O}}$ express relative errors (i.e., percentages), a unique error measurement, denoted as *relative accuracy*, can be derived as their arithmetic average:

$$e^{\%} = \frac{e^{\%,\mathrm{P}} + e^{\%,\mathrm{O}}}{2}. \tag{5.41}$$

The relative accuracy, as described earlier, has some interesting properties that make it appealing to be used in systems similar to the proposed one.

- The relative accuracy measures a *relative performance* with respect to the available training dataset. Since the training dataset plays a key role in data-driven approaches, it is more accurate to evaluate a relative (rather than absolute) performance, as it can strongly depend on the available training dataset. In this way, the efficacy of the algorithm is effectively evaluated. To this end, it is important to highlight that $e^{\%} = 100\%$ does not necessarily indicate the absence of errors. Instead, it means that the algorithm led to the selection of the best pose among those contained in the available training dataset.

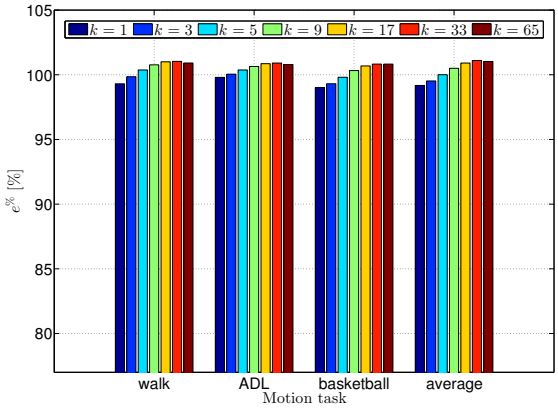
- The relative accuracy involves *both position and orientation errors*. Since the relative position and orientation errors are both expressed as percentages, the arithmetic average in Equation (5.41) can be consistently applied. This means that the relative accuracy conveys information about both position and orientation errors.

- The relative accuracy is able to evaluate the *extrapolation ability of the reconstruction algorithm*. If only the nearest neighbor is chosen to reconstruct the user pose, then $e^{\%}$ belongs to $[0, 100]$. However, in the proposed approach, a weighted interpolation of the k nearest neighbors is performed leading to the reconstruction of a possible new pose (i.e., not present in the training dataset), which may happen to be even better (or worse) than the best (or the worst) pose contained in the training dataset. In that case, $e^{\%}$ is no longer limited between 0% (worst pose in the training dataset) and 100% (best pose in the training dataset) and can thus be representative of the extrapolation ability of the reconstruction algorithm. In fact, the more often the relative accuracy goes over 100%, the more successful is the algorithm in extrapolating an accurate estimated pose from the poses available in the training dataset.

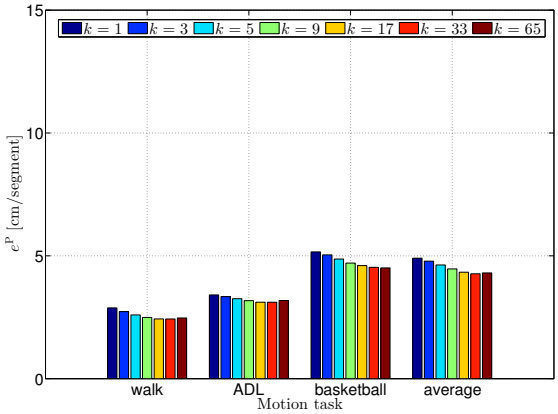
5.7 Results and Discussion

The choice of k is first investigated. As anticipated in Subsection 5.6.1, k corresponds to the number of considered nearest neighbors (i.e., training poses) which are interpolated together in order to estimate the actual pose of the user. However, note that $k = 1$ corresponds to the case where only the nearest neighbor is chosen to reconstruct the user pose. In Figure 5.4, the impact of different values of k on the system performance is investigated. In particular: the relative accuracy is shown in Figure 5.4 (a); the position error is shown in Figure 5.4 (b); and the orientation error is shown in Figure 5.4 (c). In these figures, each group of bars corresponds to a different motion task. However, since it is impractical to show all the evaluated tasks, just three illustrative tasks (i.e., walk, ADL, and basketball) are shown. In addition, the average results over all the evaluated motion tasks are also shown. In all experiments, the initial con-

(a)



(b)



(c)

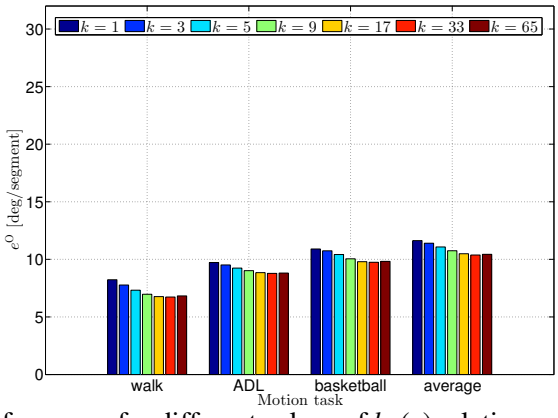


Figure 5.4: System performance for different values of k: (a) relative accuracy, (b) position error, and (c) orientation error. The performances of three illustrative tasks (namely, walk, ADL, and basketball) are shown, together with the average (over all the motion tasks) performance.

figuration of sensor devices (i.e., the one composed of 5 devices placed on ankles, wrists, and pelvis of the user) is always considered. Moreover, all the features described in Subsection 5.5.2 (i.e., relative orientations, ranges, and heights) are always taken into account in the NNS. By increasing k, slight performance improvements are observed for all motion tasks and performance metrics. However, from $k = 33$ on, the performance starts to decrease. No significant differences in relative performance are observed for the different motion tasks. On the other hand, the best performance (i.e., with $k = 33$) is achieved for the walk task, in correspondence to which $e^{\%} = 101.03\%$, $e^{\mathrm{P}} = 2.43$ cm/segment, and $e^{\mathrm{O}} = 6.73$ deg/segment. More generally, when averaging over all the motion tasks, $e^{\%} = 101.1\%$, $e^{\mathrm{P}} = 4.27$ cm/segment, and $e^{\mathrm{O}} = 10.38$ deg/segment. Since $e^{\%} > 100\%$ it can be concluded that our motion reconstruction algorithm is very effective at exploiting the potential of the available training dataset.

We now investigate the contribution of each feature to the motion reconstruction performance.[6] In Figure 5.5, the system performance is then shown as a function of the considered features. In particular: the relative accuracy is shown in Figure 5.5 (a); the position error is shown in Figure 5.5 (b); and the orientation error is shown in Figure 5.5 (c). As before, just three illustrative tasks (i.e., walk, ADL, and basketball) are shown together with the results averaged over all the evaluated motion tasks. In all experiments, the initial configuration of sensor devices (i.e., the one composed of 5 devices placed on ankles, wrists, and pelvis of the user) is always considered and k is set to 33. Even though, as expected, the case where only the relative orientations are considered has the lowest performance, slight performance improvements are observed when more features are taken into account. In particular, relative heights do not seem to produce any performance improvement when relative ranges are already used. This is, however, somehow expected since both relative ranges and relative heights values are ideally simulated and, therefore, relative heights intuitively do not

[6] We remark that, when discussing about different features, we are implicitly referring to different technologies. In particular, as already mentioned in Subsection 5.5.2: relative orientations can be extracted from inertial/magnetic sensors; relative heights from barometers; and relative ranges from UWB or acoustic ranging.

(a)

(b)

(c)

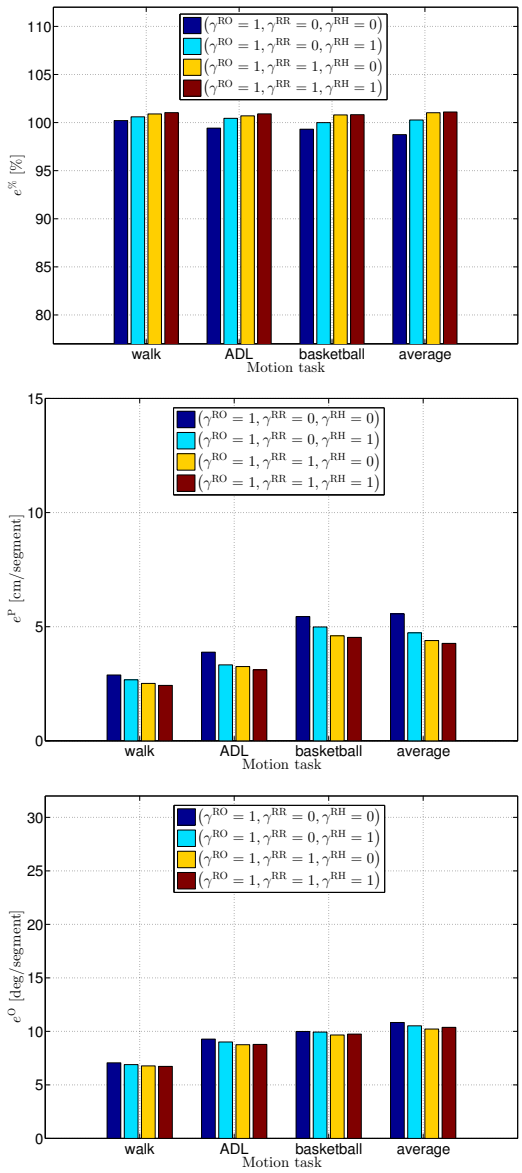
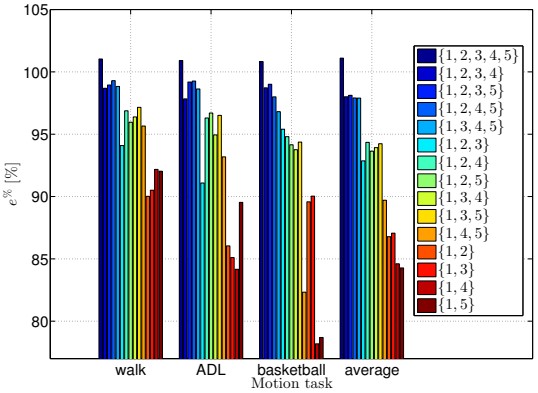


Figure 5.5: System performance while considering different types of features (namely, relative orientations, relative ranges, and relative heights): (a) relative accuracy, (b) position error, and (c) orientation error. The performances of three illustrative tasks (namely, walk, ADL, and basketball) are shown, together with the average (over all the motion tasks) performance. Note that, if feature X (X $\in \{\mathrm{RO}, \mathrm{RR}, \mathrm{RH}\}$) is considered, γ^{X} is set to 1. Adversely, γ^{X} is set to 0.

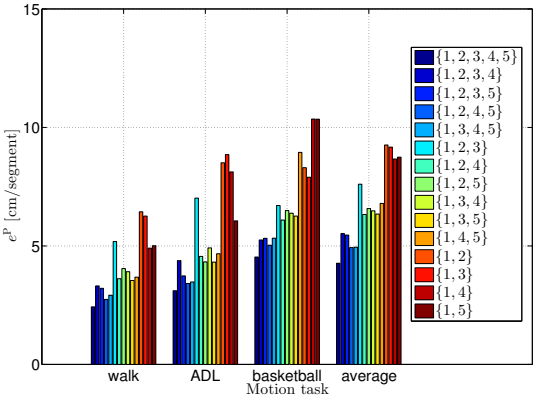
give significantly more information than relative ranges. Nevertheless, we expect that, when real (and, thus, noisier) values will instead be used, the use of both features will be expedient to increase the robustness of the motion reconstruction, possibly leading to some performance improvements. Finally, note that the largest difference in performance is observed for the basketball task, rather than the walk or ADL tasks, since more unusual movements (and, thus, less likely to be observed in different parts of the training dataset) are involved. In that case, the use of relative orientations alone may lead to possible ambiguities between different poses and, thus, to a wrong reconstruction of the user pose.

Finally, the impact of the number and the placement of sensor devices has been then investigated. To this end, in Figure 5.6 the system performances for the three illustrative tasks (i.e., walk, ADL, and basketball) and the average performance are shown for the initial configuration of $B = 5$ devices and for reduced configurations down to $B = 2$ devices. As for the previous performance figures, the relative accuracy, the position error, and the orientation error are shown, respectively, in Figure 5.6 (a), in Figure 5.6 (b), and in Figure 5.6 (c). Note that the pelvis device (i.e., the reference sensor device) is always considered. In addition, all possible combinations of the remaining $B-1$ devices are considered. Furthermore, in all experiments all the features (i.e., the relative orientations, ranges, and heights) are considered and k is set to 33. Besides the expected performance gaps between configurations with different number of devices, some interesting insights are obtained. To this end, Figure 5.7, which represents a zoom view of the average results in Figure 5.6 (a), specifically focuses on the relative accuracy of the proposed system averaged over all the considered motion tasks. From Figure 5.7, it can be noticed that the four configurations with $B = 4$ devices (where one device is each time removed from the initial configuration) have similar performance. This means that, if one device has to be removed, this choice is not critical. This is probably due to the fact that at least one device is placed on the legs and the arms. On the other end, focusing on the four configurations with $B = 2$ devices (where, in addition to the pelvis device, only one device at each time is considered), it can be observed that it is equally likely to choose a device placed on right limbs or a device placed on left limbs. However, wrists' devices guarantee better

(a)



(b)



(c)

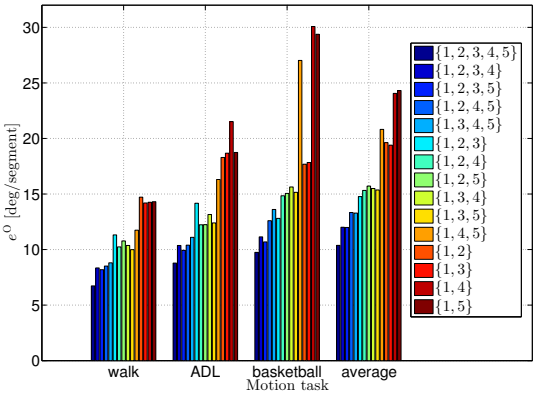


Figure 5.6: System performance for different numbers and placement strategies of sensor devices: (a) relative accuracy, (b) position error, and (c) orientation error. The performances of three illustrative tasks (namely, walk, ADL, and basketball) are shown, together with the average (over all the motion tasks) performance. The devices are numbered as shown in Figure 5.2.

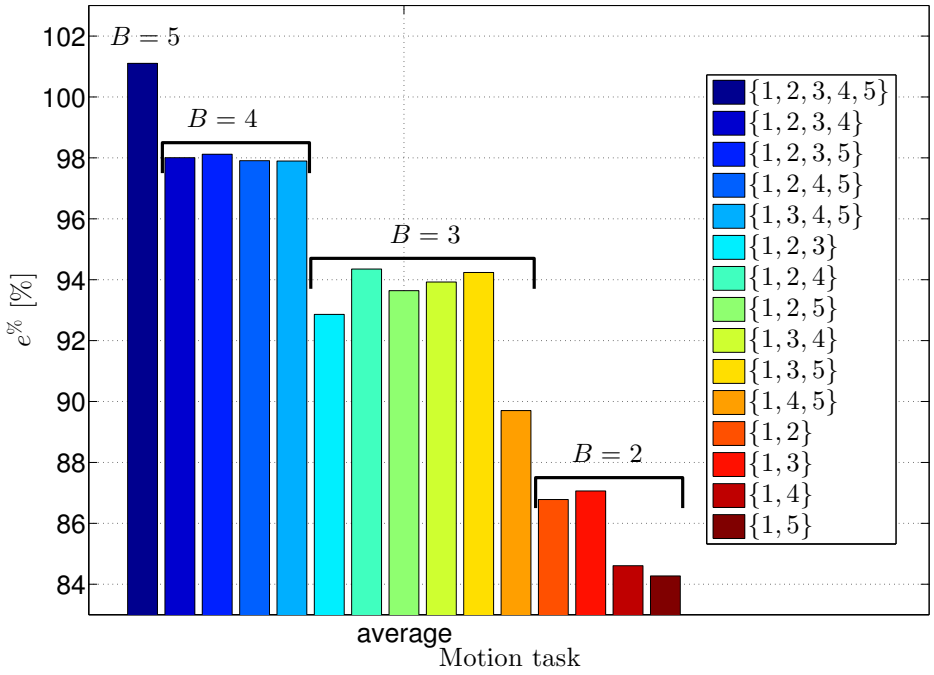


Figure 5.7: Relative accuracy for different numbers and placement strategies of sensor devices: zoom view of the average results in Figure 5.6 (a). The devices are numbered as shown in Figure 5.2.

performance (over 86%) than ankles' devices (below 85%). Finally, focusing on the remaining six configurations with $B = 3$ devices (where, each time, different pairs of sensor devices are considered), it can be observed that the configurations where the two devices are both placed on the wrists or on the ankles are the ones with lowest performance, whereas the best performance is achieved with configurations where one device is on one of the wrists and the other is on one of the ankles (both left and right sides are equally likely).

Finally, we remark that the choice of the reference device pertains to the user and does not compromise the correct behavior of the system. However, since the reference device orientation is used to rotate the reconstructed body shape, it is recommended to

choose a device able to robustly estimate its orientation. To this end, devices placed on the ankles are likely to measure noisy data due to the vibrations caused by the close contact with the ground and, similarly, devices placed on the wrists are likely to experience noisy data due to the quick, and sometimes unusual, movements of the arms. Furthermore, the selected location of the reference sensor device should be such as to avoid or minimize skin motion artifacts. For the previous reasons, we elected the pelvis device to be our reference sensor device.

In order to visually perceive the performance of the proposed motion reconstruction system, in Figure 5.8 some video frames of representative motion tasks are shown, together with ground truth poses, as extracted from the MVN Awinda suit (i.e., using 17 sensor devices), and the same poses reconstructed using the proposed algorithm (i.e., using only 5 sensor devices). For comparison purposes, in Figure 5.9 the reconstructed and the ground truth poses are also juxtaposed to better appreciate the accuracy of the proposed system.

5.8 Conclusions and Future Work

In this work, a novel NNS-based full body human motion reconstruction framework based on the use of few body-only worn sensor devices has been presented and its performance has been experimentally evaluated. The design goal of the proposed system is to estimate the full body pose of a user wearing a small number of sensor devices. Specifically, five devices are considered, placed, respectively, on the pelvis (1), the wrists (2), and the ankles (2) of the user. One of the key novelties of the proposed system relies in its self-containedness. Indeed, unlike the majority of the effective approaches appeared so far in the literature, the proposed system does not depend on any external infrastructure, yet guaranteeing very accurate motion reconstruction. Since incomplete kinematic information is obtained by using few sensor devices, the construction of a *training* dataset, composed of previously recorded motion capture data, is expedient to unambiguously extrapolate, at *runtime*, the full body pose of the user. In order to evaluate the system performance, different types of features (corresponding to different types of sensors or technologies) and different numbers

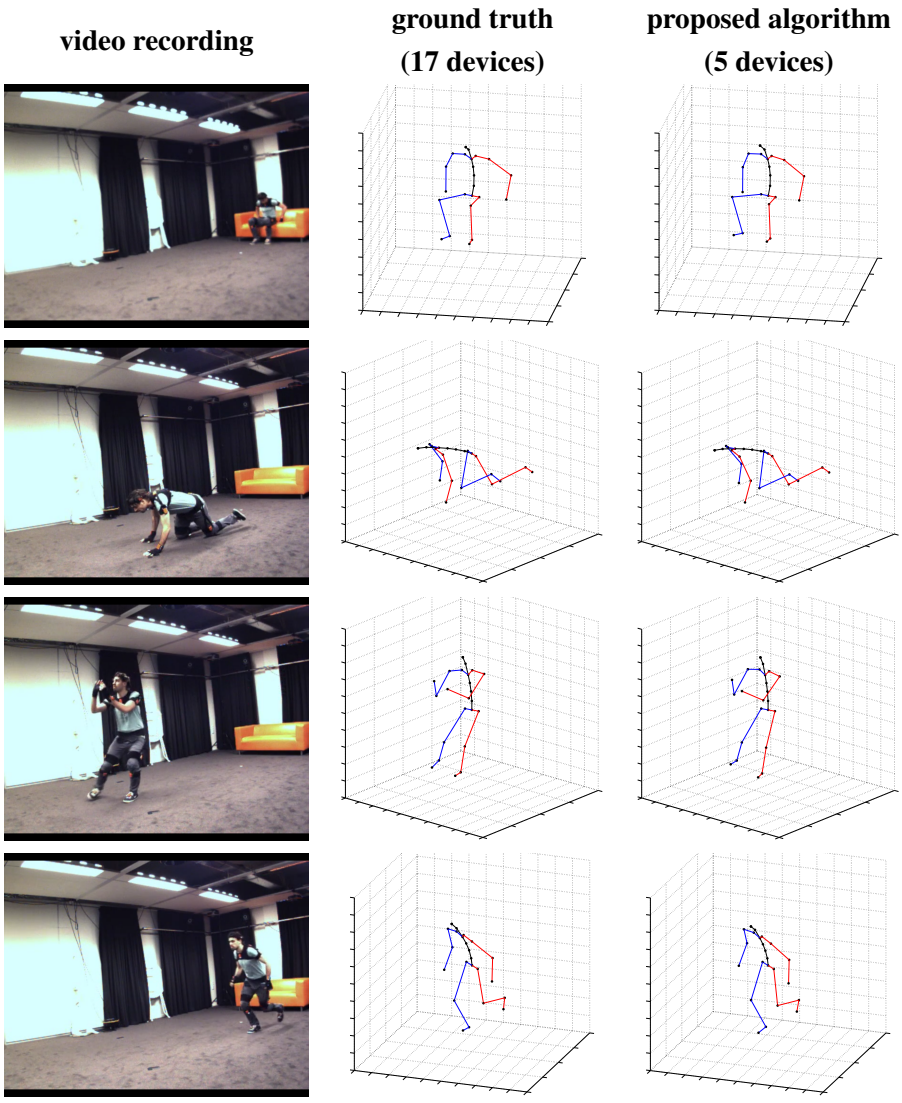


Figure 5.8: Visual performance of the proposed algorithm. In the three columns, (a) some video frames of representative motion tasks are shown, together with (b) ground truth poses, as extracted from the MVN Awinda suit (i.e., using 17 sensor devices), and (c) the same poses reconstructed using the proposed algorithm (i.e., using only 5 sensor devices). A different motion task is shown at each row (namely, from top to down, sit, crawl, basketball, run).

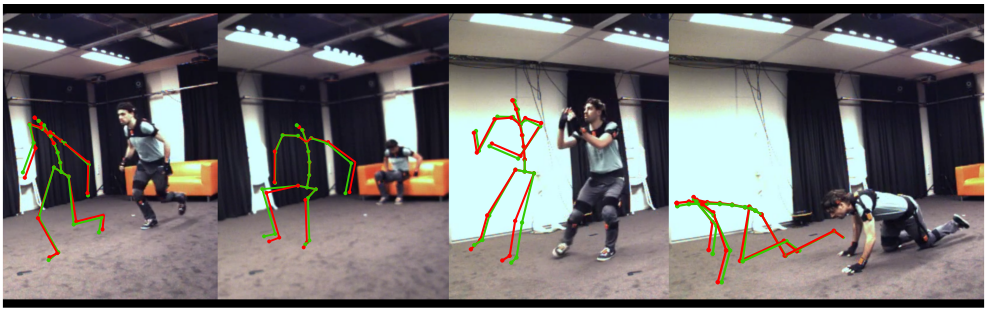


Figure 5.9: The visual performance of the proposed algorithm is shown by juxtaposing the pose reconstructed using the proposed algorithm (green stick figure) and the ground truth pose estimated using Xsens MVN Awinda (red stick figure). Different motion tasks (corresponding to those in Figure 5.8) are shown.

and placement strategies of the sensor devices have been taken into account. The proposed algorithm has shown very promising performance and good extrapolation skills. Specifically, when considering the heuristically chosen best parameters, the average position error is around 4 cm/segment and the average orientation error is around 10 deg/segment, whereas the average relative accuracy is equal to 101.1%, i.e., the proposed algorithm even allows to effectively extrapolate poses which are not in the training dataset.

This chapter represents a proof-of-concept (preliminary) work in the direction of low-complexity accurate motion reconstruction and many aspects should be better investigated. In particular, the system should be tested with more than one user. However, from preliminary experiments, we do not expect any significant performance degradation, especially if all the three types of features (namely, relative orientations, relative ranges, and relative heights) are still considered in the NNS. Another aspect that should be better investigated concerns the use of real data for relative ranges (e.g., using UWB technology) and relative heights (e.g., using barometers). Finally, since the considered training dataset is very limited, it would be interesting to quantify the impact of a larger and more exhaustive training dataset on the system performance.

Due to the very promising performance and the large room for improvement,

we believe that the proposed system might potentially represent a low-cost and low-complexity self-contained solution to standard motion capture systems.

Bibliography

[1] Optoelectronic motion capture system from Vicon. Available from: http://www.vicon.com/ [cited 17-01-2014].

[2] Kinect: markerless optical motion capture system from Microsoft. Available from: http://www.microsoft.com/en-us/kinectforwindows/ [cited 17-01-2014].

[3] MVN: inertial motion capture system from Xsens. Available from: http://www.xsens.com/en/general/mvn [cited 17-01-2014].

[4] MVN Awinda: wireless inertial motion capture system from Xsens. Available from: http://www.xsens.com/en/mvn-awinda [cited 17-01-2014].

[5] Gypsy: mechanical motion capture system from Meta Motion. Available from: http://www.metamotion.com/gypsy/gypsy-motion-capture-system.htm [cited 17-01-2014].

[6] MotionStar: magnetic motion capture system from Ascension Technology Corporation. Available from: http://www.ascension-tech.com/realtime/RTmotionStarTethered.php [cited 17-01-2014].

[7] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Forster, G. Troster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. d. R. Millan. Collecting complex activity datasets in highly rich networked

sensor environments. In *2010 7th International Conference on Networked Sensing Systems (INSS)*, pages 233–240, Kassel, Germany, June 2010.

[8] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Troster, J. d. R. Millan, and D. Roggen. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, November 2013.

[9] N. Yazdi, F. Ayazi, and K. Najafi. Micromachined inertial sensors. *Proceedings of the IEEE*, 86(8):1640–1659, August 1998.

[10] R. B. Martin. A genealogy of biomechanics. In *Proc. of the 23rd Annual Conf. of the American Society of Biomechanics*, Pittsburgh, PA, USA, October 1999.

[11] R. Baker. The history of gait analysis before the advent of modern computers. *Gait & Posture*, 26(3):331–342, September 2007.

[12] M. C. Nussbaum. *Aristotle's De Motu Animalium*. Princeton University Press, Princeton, NJ, USA, 1978.

[13] G. A. Borelli. *De Motu Animalium*. Ex Typographia A. Bernabò, Rome, Italy, 1680.

[14] W. Weber and E. Weber. *Die Mechanik der Menschlichen Gehwerkzeuge (Mechanics of Human Gait)*. Dieterichschen Buchhandlung, Gittingen, Germany, 1836.

[15] C. W. Braune and O. Fischer. *Der Gang des Menschen (The Human Gait)*. B. G. Teubner, Leipzig, Germany, 1899.

[16] D. Roetenberg, H. J. Luinge, and P. J. Slycke. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors. *Xsens Technologies B. V.*, April 2009.

[17] E. R. Bachmann. *Inertial and Magnetic Tracking of Limb Segment Orientation for Inserting Humans into Synthetic Environments*. PhD

thesis, Naval Postgraduate School, Monterey, CA, USA, 2000. Available from: `http://gamepipe.usc.edu/$\sim$zyda/Theses/EricBachmann2.pdf`.

[18] D. Roetenberg. *Inertial and Magnetic Sensing of Human Motion*. PhD thesis, University of Twente, Enschede, the Netherlands, 2006. Available from: `http://doc.utwente.nl/56176/1/thesis_Roetenberg.pdf`.

[19] M. Brodie, A. Walmsley, and W. Page. Fusion motion capture: a prototype system using inertial measurement units and GPS for the biomechanical analysis of ski racing. *Sports Technology*, 1(1):17–28, June 2008.

[20] Y. Tao, H. Hu, and H. Zhou. Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation. *The International Journal of Robotics Research*, 26(6):607–624, June 2007.

[21] D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović. Practical motion capture in everyday surroundings. *ACM Trans. on Graphics*, 26(3):35.1–35.9, July 2007.

[22] G. M. Djuknic and R. E. Richton. Geolocation and assisted GPS. *IEEE Computer*, 34(2):123–125, February 2001.

[23] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proc. IEEE Conf. on Computer Commun.* (INFOCOM), volume 2, pages 775–784, Tel Aviv, Israel, March 2000.

[24] K. Kaemarungsi. *Design of indoor positioning systems based on location fingerprinting technique*. PhD thesis, University of Pittsburgh, Pittsburgh, 2005. Available from: `http://etd.library.pitt.edu/ETD/available/etd-02232005-235903/unrestricted/dissertation28Feb05.pdf`.

[25] N. Swangmuang and P. Krishnamurthy. Location fingerprint analyses toward efficient indoor positioning. In *IEEE Int. Conf. Pervasive Comp. and Commun.* (PERCOM), pages 100–109, Hong Kong, China, March 2008.

[26] M. Giuberti, M. Martalò, and G. Ferrari. Fingerprinting-based wireless 3D localization for motion capture applications. In *Proc. 1st ACM MobiHoc Workshop on Pervasive Wireless Healthcare (*MobileHealth), pages 6.1–6.8, Paris, France, May 2011.

[27] M. Martalò, M. Giuberti, and G. Ferrari. Experimental investigation of wireless sensor networks for fingerprinting-based posture recognition. In *Riunione annuale 2011 del Gruppo nazionale Telecomunicazioni e Teoria dell'Informazione (*GTTI), Messina and Taormina, Italy, June 2011.

[28] S. O. H. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Technical report, Department of Mechanical Engineering, University of Bristol, Bristol, UK, April 2010. Available from: `http://www.x-io.co.uk/res/doc/madgwick_internal_report.pdf`.

[29] H. J. Luinge, P. H. Veltink, and C. T. M. Baten. Ambulatory measurement of arm orientation. *Journal of Biomechanics*, 40(1):78–85, January 2007.

[30] D. Roetenberg, P. J. Slycke, and P. H. Veltink. Ambulatory position and orientation tracking fusing magnetic and inertial sensing. *IEEE Trans. on Biomedical Eng.*, 54(5):883–890, May 2007.

[31] C. Chang, B. Lange, M. Zhang, S. Koenig, P. Requejo, N. Somboon, A. Sawchuk, and A. A. Rizzo. Towards pervasive physical rehabilitation using Microsoft Kinect. In *Proc. of the 6th Int. Conf. on Pervasive Computing Technologies for Healthcare* (PervasiveHealth), pages 159–162, San Diego, CA, USA, May 2012.

[32] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, Novemeber 2006.

[33] L. Chen, H. Wei, and J. Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 34(15):1995–2006, November 2013.

[34] D. A. Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, New York, NY, USA, 1990.

[35] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanen. A probabilistic approach to WLAN user location estimation. *Int. Journal Wireless Information Networks*, 9(3):155–164, July 2002.

[36] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification and Scene Analysis. 2nd edition*. Wiley-Interscience, New York, NY, USA, 2000.

[37] D. Shepard. A two-dimensional interpolation function for irregularly spaced data. In *ACM National Conf.*, 1968.

[38] J. B. Kuipers. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, Princeton, NJ, USA, 1999.

[39] X. Yun, E. R. Bachmann, and R. B. McGhee. A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements. *IEEE Trans. on Instrumentation and Measurement*, 57(3):638–650, March 2008.

[40] SunSPOT World. Available from: `http://www.sunspotworld.com` [cited 17-01-2014].

[41] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[42] Š. Obdržálek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel. Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population. In *Proc. of the 34th Annual Int. Conf. of the IEEE Eng.*

in Medicine and Biology Society (EMBS), pages 1188–1193, San Diego, CA, USA, August 2012.

[43] F. Gustafsson. *Statistical sensor fusion*. Studentlitteratur, Lund, Sweden, 2010.

[44] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers. A review of wearable sensors and systems with application in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 9(21):1–17, April 2012.

[45] L. Bao and S. Intille. Activity recognition from user annotated acceleration data. In *Pervasive Computing*, pages 1–17, April 2004.

[46] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 159–163, Grenoble, France, 2005.

[47] R. Aylward and J. Paradiso. A compact, high-speed, wearable sensor network for biomotion capture and interactive media. In *Proceedings of the 6th international conference on Information processing in sensor networks (IPSN)*, pages 380–389, Cambridge, MA, USA, April 2007.

[48] J. A. Ward, P. Lukowicz, G. Tröster, and T. E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, October 2006.

[49] R. Jafari, W. Li, R. Bajcsy, S. Glaser, and S. Sastry. Physical activity monitoring for assisted living at home. In *Proceedings of International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pages 213–219, Aachen, Germany, March 2007.

[50] S. Pirttikangas, K. Fujinami, and T. Nakajima. Feature selection and activity recognition from wearable sensors. In *International Symposium on Ubiquitous Computing Systems*, pages 516–527, Seoul, Korea, October 2006.

[51] E. Guenterberg, S. Ostadabbas, H. Ghasemzadeh, and R. Jafari. An automatic segmentation technique in body sensor networks based on signal energy. In *Fourth International Conference on Body Area Networks (BodyNets)*, pages 1–7, Los Angeles, CA, USA, April 2009.

[52] D. M. Sherrill, M. L. Moy, J. J. Reilly, and P. Bonato. Using hierarchical clustering methods to classify motor activities of copd patients from wearable sensor data. *Journal of NeuroEngineering and Rehabilitation*, 2(1):1–14, June 2005.

[53] A. Yang, S. Iyengar, S. S. Sastry, R. Bajcsy, P. Kuryloski, and R. Jafari. Distributed segmentation and classification of human actions using a wearable motion sensor network. In *Proceedings of the Computer Vision and Pattern Recognition Workshops on Human Communicative Behavior Analysis (CVPRW)*, pages 1–8, Anchorage, AK, USA, June 2008.

[54] Opportunity Dataset. Available from: `http://archive.ics.uci.edu/ml/datasets/OPPORTUNITY+Activity+Recognition` [cited 17-01-2014].

[55] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

[56] M. Giuberti and G. Ferrari. BSN-based activity classification: a low complexity windowing-&-classification approach. *Advances in Science and Technology*, 85:53–58, 2013.

[57] European Parkinson's Disease Association. Available from: `http://www.epda.eu.com/en/parkinsons/in-depth/parkinsonsdisease/` [cited 17-01-2014].

[58] B. Chen, S. Patel, T. Buckley, R. Rednic, D. McClure, L. Shih, D. Tarsy, M. Welsh, and P. Bonato. A web-based system for home monitoring of patients

with ParkinsonâĂŹs Disease using wearable sensors. *IEEE Transactions on Biomedical Engineering*, 58(3):831–836, March 2011.

[59] S. Nuzik, R. Lamb, A. Vansant, and S. Hirt. Sit-to-stand movement pattern. a kinematic study. *Physical Therapy*, 66(11):1708–1713, 1986.

[60] D. Giansanti, G. Maccioni, F. Benvenuti, and V. Macellari. Inertial measurement units furnish accurate trunk trajectory reconstruction of the sit-to-stand manoeuvre in healthy subjects. *Medical and Biological Eng. and Comput.*, 45(10):969–976, October 2007.

[61] A. Salarian, H. Russmann, F. J. G. Vingerhoets, C. Dehollain, Y. Blanc, P. R. Burkhard, and K. Aminian. Gait assessment in Parkinson's Disease: Toward an ambulatory system for long-term monitoring. *IEEE Trans. on Biomedical Eng.*, 51(8):1434–1443, August 2004.

[62] A. Salarian, C. Zampieri, F. B. Horak, P. Carlson-Kuhta, J. G. Nutt, and K. Aminian. Analyzing 180º turns using an inertial system reveals early signs of progression of ParkinsonâĂŹs Disease. In *Proc. of the 31st Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society* (EMBS), pages 224–227, Minneapolis, MN, USA, September 2009.

[63] A. Salarian, H. Russmann, C. Wider, P. R. Burkhard, F. J. G. Vingerhoets, and K. Aminian. Quantification of tremor and bradykinesia in ParkinsonâĂŹs Disease using a novel ambulatory monitoring system. *IEEE Trans. on Biomedical Eng.*, 54(2):313–322, February 2007.

[64] S. Fahn and R. L. Elton. *Recent Developments in Parkinson's Disease*, volume 2, chapter Unified Parkinson's Disease Rating Scale, pages 153–163. Macmillan Health Care Information, Florham Park, NJ, USA, 1987.

[65] C. G. Goetz, S. Fahn, P. Martinez-Martin, W. Poewe, C. Sampaio, G. T. Stebbins, M. B. Stern, B. C. Tilley, R. Dodel, B. Dubois, R. Holloway, J. Jankovic, J. Kulisevsky, A. E. Lang, A. Lees, S. Leurgans, P. A. LeWitt, D. Nyenhuis, C. W. Olanow, O. Rascol, A. Schrag, J. A. Teresi, J. J. Van Hilten,

and N. LaPelle. Movement disorder society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Process, format, and clinimetric testing plan. *Movement Disorders*, 22(1):41–47, January 2007.

[66] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato. Monitoring motor fluctuations in patients with ParkinsonâĂŹs Disease using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 13(6):864–873, November 2009.

[67] S. Das, L. Trutoiu, A. Murai, D. Alcindor, M. Oh, F. De la Torre, and J. Hodgins. Quantitative measurement of motor symptoms in Parkinson's disease: A study with full-body motion capture data. In *Proc. of the 2011 Annual Int. Conf. of the Engineering in Medicine and Biology Society* (EMBC), pages 6789–6792, 2011.

[68] D. A. Heldman, D. E. Filipkowski, D. E. Riley, C. M. Whitney, B. L. Walter, S. A. Gunzler, J. P.Giuffrida, and T. O. Mera. Automated motion sensor quantification of gait and lower extremity bradykinesia. In *Proc. of the 34th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society* (EMBS), pages 1956–1959, San Diego, CA, USA, August 2012.

[69] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca. SHIMMER - a wireless sensor platform for noninvasive biomedical research. *IEEE Sensors Journal*, 10(9):1527–1534, September 2010.

[70] R. B. Davis, S. Ounpuu, D. J. Tyburski, and J. R. Gage. A gait analysis data collection and reduction technique. *Human Movement Science*, 10:575–587, 1991.

[71] A. Ferrari, M. G. Benedetti, E. Pavan, C. Frigo, D. Bettinelli, P. Rabuffetti, P. Crenna, and A. Leardini. Quantitative comparison of five current protocols in gait analysis. *Gait Posture*, 28:207–216, 2008.

[72] I. Y. Bar-Itzhack. New method for extracting the quaternion from a rotation matrix. *Journal of Guidance, Control, and Dynamics*, 23(6):1085–1087, November-December 2000.

[73] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics* (ICORR), pages 1–7, Zurich, Switzerland, June 2011.

[74] M. Multon, L. France, M. P. Cani-Gascuel, and D. Debunne. Computer animation of human walking: a survey. *Journal of Visualization and Computer Animation*, 10(1):39–54, 1999.

[75] A. Brunderlin and T. W. Calvert. Goal-directed, dynamic animation of human walking. *SIGGRAPH Comput. Graph.*, 23(3):233–242, July 1989.

[76] R. Boulic and D. Thalmann. Combined direct and inverse kinematic control for articulated figure motion editing. *Computer Graphics Forum*, 11(4):189–202, 1992.

[77] A. Brunderlin and T. W. Calvert. Interactive animation of personalized human locomotion. *Graphics Interface*, pages 17–23, 1993.

[78] A. Brunderlin and T. W. Calvert. Knowledge-driven, interactive animation of human running. *Graphics Interface*, 96:213–221, 1996.

[79] M. H. Raibert and J. K. Hodgins. Animation of dynamic legged locomotion. *SIGGRAPH Comput. Graph.*, 25(4):349–358, July 1991.

[80] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *Proc. of the 22nd annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 71–78, 1995.

[81] H. Ko and N. I. Badler. Animating human locomotion with inverse dynamics. *IEEE Comput. Graph. Appl.*, 16(2):50–59, March 1996.

[82] J. Laszlo, M. van de Panne, and E. Fiume. Limit cycle control and its application to the animation of balancing and walking. In *Proc. of the 23rd annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 155–162, 1996.

[83] C. F. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5):32–40, September 1998.

[84] P. J. Sloan, C. F. Rose, and M. F. Cohen. Shape by example. In *Proc. of the 2001 ACM Symposium on Interactive 3D Graphics*, pages 135–143, 2001.

[85] N. F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):371–387, 2002.

[86] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *Proc. of the 22nd annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 91–96, 1995.

[87] R. Bowden. Learning statistical models of human motion. In *IEEE Workshop on Human Modeling, Analysis and Synthesis* (CVPR), 2000.

[88] Y. Li, T. Wang, and H. Shum. Motion texture: a two-level statistical model for character motion synthesis. *ACM Trans. Graph.*, 21(3):465–472, July 2002.

[89] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, August 2004.

[90] H. J. Shin and J. Lee. Motion synthesis and editing in low-dimensional spaces. *Comput. Animat. Virtual Worlds*, 17(3–4):219–227, July 2006.

[91] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems* (NIPS), pages 1345–1352, 2006.

[92] G. W. Taylor, L. Sigal, D. J. Fleet, and G. E. Hinton. Dynamical binary latent variable models for 3D human pose tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR), pages 631–638, 2010.

[93] R. Urtrasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR), pages 238–245, 2006.

[94] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (TPAMI), 30(2):283–298, February 2008.

[95] A. Bruderlin and L. Williams. Motion signal processing. In *Proc. of the 22nd annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 97–104, 1995.

[96] A. Witkin and Z. Popović. Motion warping. In *Proc. of the 22nd annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 105–108, 1995.

[97] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *Proc. of the 29th annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 473–482, San Antonio, TX, USA, 2002.

[98] O. Arikan and D. A. Forsyth. Interactive motion generation from examples. *ACM Trans. Graph.*, 21(3):483–490, July 2002.

[99] S. I. Park, H. J. Shin, and S. Y. Shin. On-line locomotion generation based on motion blending. In *Proc. of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (SCA), pages 105–111, San Antonio, TX, USA, 2002.

[100] L. Kovar and M. Gleicher. Flexible automatic motion blending with registration curves. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (SCA), pages 214–224, San Diego, CA, USA, 2003.

[101] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.*, 23(3):559–568, August 2004.

[102] S. I. Park, H. J. Shin, T. H. Kim, and S. Y. Shin. On-line motion blending for real-time locomotion generation. *Comput. Animat. Virtual Worlds*, 15(3–4):125–138, July 2004.

[103] M. Brand and A. Hertzmann. Style machines. In *Proc. of the 27th annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 183–192, 2000.

[104] R. Urtrasun, P. Glardon, R. Boulic, D. Thalmann, and P. Fua. Style-based motion synthesis. *Computer Graphics Forum*, 23(4):799–812, 2004.

[105] E. Hsu, K. Pulli, and J. Popović. Style translation for human motion. *ACM Trans. Graph.*, 24(3):1082–1089, July 2005.

[106] M. Gleicher. Retargetting motion to new characters. In *Proc. of the 25th annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 33–42, 1998.

[107] J. Lee and S. Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proc. of the 26th annual conference on Computer graphics and interactive techniques* (SIGGRAPH), pages 39–48, 1999.

[108] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, 20(2):67–94, April 2001.

[109] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.*, 21(3):491–500, July 2002.

[110] M. van de Panne. From footprints to animation. *Computer Graphics Forum*, 16(4):211–223, 1997.

[111] S. Chung and J. K. Hahn. Animation of human walking in virtual environments. In *Proc. of the Computer Animation* (CA), pages 4–15, 1999.

[112] M. G. Choi, J. Lee, and S. Y. Shin. Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Trans. Graph.*, 22(2):182–203, April 2003.

[113] S. Coros, P. Beaudoin, K. K. Yin, and M. van de Pann. Synthesis of constrained walking skills. *ACM Trans. Graph.*, 27(5):113:1–113:9, December 2008.

[114] C. Wu, J. Medina, and V. B. Zordan. Simple steps for simply stepping. In *Proc. of the 4th Int. Symposium on Advances in Visual Computing* (ISVC), pages 97–106, Las Vegas, NV, USA, 2008.

[115] A. Egges and B. J. H. van Basten. One step at a time: animating virtual characters based on foot placement. *Vis. Comput.*, 26(6–8):497–503, June 2010.

[116] B. J. H. van Basten, P. W. A. M. Peeters, and A. Egges. The step space: example-based footprint-driven motion synthesis. *Comput. Animat. Virtual Worlds*, 21(3–4):433–441, May 2010.

[117] B. J. H. van Basten, S. A. Stüvel, and A. Egges. A hybrid interpolation scheme for footprint-driven walking synthesis. In *Proc. of Graphics Interface* (GI), pages 9–16, 2011.

[118] D. J. Sturman. Computer puppetry. *IEEE Computer Graphics and Applications*, 18(1):38–45, 1998.

[119] J. Chai and J. K. Hodgins. Performance animation from low-dimensional control signals. *ACM Trans. Graph.*, 24(3):686–696, July 2005.

[120] G. Liu, J. Zhang, W. Wa, and L. McMillan. Human motion estimation from a reduced marker set. In *Proc. of the 2006 Symposium on Interactive 3D Graphics and games* (I3D), pages 35–42, Redwood City, CA, USA, 2006.

[121] R. Slyper and J. K. Hodgins. Action capture with accelerometers. In *Proc. of the 2008 ACM SIGGRAPH/Eurographics symposium on Computer animation* (SCA), pages 193–199, Dublin, Ireland, 2008.

[122] T. Shiratori and J. K. Hodgins. Accelerometer-based user interfaces for the control of a physically simulated character. *ACM Trans. Graph.*, 27(5):123:1–123:9, December 2008.

[123] J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H. Seidel, and B. Eberhardt. Motion reconstruction using sparse accelerometer data. *ACM Trans. Graph.*, 30(3):18:1–18:12, May 2011.

[124] C. Guo, X. Liang, and J. Liu. Continuous full-body motion control of virtual human using sparse Wiimotes. *International Journal of Advanced Robotic Systems*, 9, 2012.

[125] H. Liu, X. Wei, J. Chai, I. Ha, and T. Rhee. Realtime human motion control with a small number of inertial sensors. In *Proc. of Symposium on Interactive 3D Graphics and games* (I3D), pages 133–140, San Francisco, CA, USA, 2011.

[126] J. Kim, Y. Seol, and J. Lee. Realtime performance animation using sparse 3D motion sensors. In *International Conference on Motion in Games* (MIG), pages 31–42, 2012.

[127] B. Krüger, J. Tautges, A. Weber, and A. Zinke. Fast local and global similarity searches in large motion capture databases. In *Proc. of the 2010 ACM SIGGRAPH/Eurographics symposium on Computer animation* (SCA), pages 1–10, Madrid, Spain, 2010.

[128] Invensense. Available from: `http://www.invensense.com/` [cited 17-01-2014].

[129] Decawave. Available from: `http://www.decawave.com/` [cited 17-01-2014].

[130] M. Tournier, X. Wu, N. Courty, E. Arnaud, and L. Revéret. Motion compression using principal geodesics analysis. *Computer Graphics Forum*, 28(2):355–364, 2009.

Acknowledgments

It's done..
Eventually..

And, although it's still not over, it's good sometimes to stop and stare at what you left behind rather than at what is coming next. And what I see behind me makes me feel a lucky person, for everything I've achieved so far. And it's not about work achievements —of which I can't say that I'm not proud—It's about life. I keep on discovering that happiness stands in little things.. little good things..

I have a few very good friends (you know who you are!) and I have a beautiful and smart girl by my side. And that's all that counts! Really! And I want them to know that I'm building my future around them!

Fra, I love you! ♡ —and you know it!— You're growing, but I'm growing with you. We're growing. And it's so beautiful to see that we're going in the same direction. Ever more! We're crossing our paths as we're crossing our souls.

At this point, it would be however unfair to forget to thank people who've been beside me in the last years. Gian, thank you for believing in me from the very start of this experience and thank you for making me feel as a friend rather than an employee. A big "thank you" goes to all the guys and girls that I've met at WASNLab and during my studies, who made this time here always pleasant and motivating. Thanks to all

the people I've met in the VRehab project and to those I've met in the Netherlands (Henk, really sorry for stealing part of your thesis title! ☺), who really contributed to my professional growth, still staying "at my level."

Finally, thanks to my parents and my whole family, who really supported (and still support) me without asking for anything in return. I hope they'll always be proud of me.

It's done..
Eventually..