Pedestrian Dead Reckoning: a neuro-fuzzy approach with inertial measurements fusion based on Kalman filter and DWT

By
Mariana Natalia Ibarra Bonilla, M.Sc.

Thesis submitted in partial fulfillment of the requirements to obtain the degree

DOCTOR OF SCIENCE IN ELECTRONICS

At the

Instituto Nacional de Astrofísica, Óptica y Electrónica

Directors:
Dr. Juan Manuel Ramírez-Cortés
Dr. Ponciano Jorge Escamilla-Ambrosio

Tonantzintla, Puebla, 2015

Abstract

Pedestrian dead reckoning (PDR) is a navigation technique that provides and maintains the geographical position for a person travelling on foot by using self-contained sensors. In PDR techniques, each new position estimate is based on the previous estimate of the last step taking advantage of the sequential nature of pedestrian motion. In general, a PDR algorithm is composed of three parts: step detection, estimation of walking distance and tracking of sensor attitude. Most of PDR approaches consider low cost MEMS accelerometers, gyroscopes and magnetometers as the source of information. However, these sensors are affected by sensor noise and drift, which introduce errors in the displacement and relative attitude changes in the sensor´s frame of reference with respect to the human body. In order to improve the accuracy of the attitude estimation reducing the time-varying drift, this work presents the development of a Kalman Filter with Neuro-Fuzzy adaptation (KF-NFA), relying on information derived from triaxial accelerometer and gyroscope sensors contained in an inertial measurement unit (IMU). The adaptation process is performed on the filter statistical information matrix *R*, which is tuned using an Adaptive Neuro Fuzzy Inference System (ANFIS) based on the filter innovation sequence through a covariance-matching technique. Besides, in order to improve the distance traveled by the pedestrian, and consequently the localization accuracy, different types of activities are classified using a Multi-layer Perceptron (MLP) Neural Network (NN) according to extracted features based on Wavelet Decomposition. Basic activities that a pedestrian performs in his daily life, such as walking, walking fast, jogging and running, are considered. Subsequently the step-length is dynamically estimated using a multiple-input-single-output (MISO) Fuzzy Inference System (FIS). Validation testing and obtained results are presented.

Resumen

Pedestrian dead reckoning, (PDR) es una técnica de navegación que proporciona y mantiene la posición geográfica de un peatón que usa sensores auto-contenidos. En los sistemas PDR, cada nueva posición está estimada en base a una estimación anterior del último paso, así se toma ventaja de la naturaleza secuencial del movimiento del peatón. En general, un algoritmo PDR está compuesto de tres partes: detección de pasos, estimación de la distancia recorrida y la estimación de la orientación del sensor. La mayoría de los sistemas PDR consideran como fuente de información a los sensores MEMS de bajo costo, tales como acelerómetros, giroscopios y magnetómetros. Sin embargo, estos sensores son afectados por el ruido y el sesgo en las mediciones, los cuales introducen error en el desplazamiento y cambios en la orientación relativa en el cuadro de referencia del sensor con respecto al cuerpo humano. Con la finalidad de mejorar la exactitud de la orientación y reducir el sesgo en las mediciones que varía con el tiempo, este trabajo presenta el desarrollo de un Filtro Kalman con una adaptación Neuro-Difusa, que usa la información de un acelerómetro y giroscopio de tres ejes contenidos en una unidad de medición inercial. El proceso de adaptación es ejecutado sobre la matriz de información estadística del filtro, R, la cual es ajustada usando un Sistema Adaptativo de Inferencia Neuro-Difuso basado en la secuencia de innovación del filtro a través de la igualación de covarianzas. Por otra parte, para mejorar la estimación de la distancia recorrida por el peatón, y consecuentemente la exactitud de la localización, se clasifican diferentes actividades en una Red Neuronal Artificial Perceptrón Multicapa usando las características obtenidas de una descomposición Wavelet de la señal de la aceleración. Se proponen las actividades: caminar, caminar rápido, trotar y correr, y se incorpora un altímetro para obtener los datos de la altura. Posteriormente la longitud del paso se estima dinámicamente usando un Sistema de Inferencia Difuso, de múltiples entradas a una salida. Se presentan las pruebas de validación y los resultados obtenidos.

Contents

1 Introduction 1
1.1 Justification 2
1.2 Problem definition 3
1.3 Objectives 4
1.3.1 General objective 4
1.3.2 Particular objectives 4
1.3.3 Solution scope 5
1.4 Background 5
1.4.1 Pedestrian Dead Reckoning 7
1.4.1.1 Step detection 7
1.4.1.2 Step length estimation 8
1.4.1.3 Attitude estimation 12
1.5 Thesis organization 14

2 Theoretical framework 15
2.1 Kalman filtering 15
2.1.1 The Kalman filter algorithm 15
2.2 Neuro-Fuzzy systems 19
2.2.1 Fuzzy Inference Systems 20
2.2.1.1 Fuzzification 22
2.2.1.2 The inference mechanism 23
2.2.1.3 Defuzzification 28
2.2.1.4 Types of Fuzzy Inference Systems 29
2.2.2 Artificial neural networks 31
2.2.3 Adaptive Network based Fuzzy Inference System (ANFIS) 34
2.3 Discrete Wavelet Transform 36
2.4 Summary 38

3 Attitude estimation using Neuro Fuzzy Adaptive Kalman Filter 39
3.1 IMU data fusion for attitude estimation through Kalman filtering 40
3.1.1 The process and measurement model 43
3.2 Attitude estimation using Neuro-Fuzzy Adaptive Kalman Filtering 45
3.2.1 Innovation based adaptive estimation algorithm 45
3.2.2 Adaptive adjustment of the measurement noise covariance matrix *R* with *Q* known 46
3.3 The Neuro-Fuzzy Adaptive Kalman filtering test 49
3.4 Summary 52

4 Step length estimation and activity detection in a PDR system 53
4.1 Fuzzy model for step length estimation 54
4.2 Pedestrian activity classification algorithm 55
4.3 Classification using a multi-layer perceptron neural network 58
4.4 Step length estimation algorithm 59
4.4.1 Step detection 60
4.4.2 Step length estimation 63
4.5 Summary 65

5 Experimental results 66
5.1 KF-NFA attitude estimation results 66
5.2 Activity detection results 68
5.3 PDR system results 69
5.3.1 Experiment 1 70
5.3.2 Experiment 2 73
5.3.2.1 Test 1 73
5.3.2.2 Test 2 75
5.3.3 Experiment 3 78
5.3.3.1 Test 1 78
5.3.3.2 Test 2 80
5.3.4 Experiment 4 83

5.3.4.1 Test 1 83
5.3.4.2 Test 2 86
5.3.5 General results 88
5.4 Summary 90

6 Conclusions 91

Publications 94
References 95

Chapter 1

Introduction

Modern life style makes people increasingly mobile. This has stimulated the development of mobile devices fitted with communications and positioning technologies, e.g. Wi-Fi and global positioning system (GPS) [1]. As a result, the demand for systems capable to deliver information to people according to their geographical location has dramatically increased. To satisfy this demand, lately a new set of applications known as *Location-Based Services* (LBS) or (location-based applications) have been developed.

LBS integrate satellite navigation, mobile networking, and mobile computing to enable such services. A broad range of LBS applications are continuously emerging in the following market segments [2]: emergency location, personal child security, asset tracking and people finder, fleet management, telematics, driving directions, wireless gaming, location based billing, information directory services, push/pull advertising, and many others.

A fundamental component common to all LBS is the use of positioning technologies to track the movement of mobile users and to deliver information services to these users on the move at the right time and right location [3]. Therefore, the effective use of positioning technologies has a significant impact on the performance, reliability, security, and privacy of LBS systems and applications.

Initially, many research studies were focused on LBS applications for outdoor environments and the *Global Positioning System* (GPS) as the main source of localization. In outdoor environments, where there is a strong satellite signal, the GPS technology is able to provide the user's location with accuracy and continuity of service. However, a lot of the movements of persons take place indoors and in dense urban environments where buildings block satellite signals and, consequently, the GPS system is completely disrupted or it does not provide accuracy and continuity of service.

1.1 Justification

Location in both environments, indoors and outdoors, is difficult to obtain due to the lack of sensors that can work in both situations. In spite of that, location in both indoor and outdoor environments is a basic requirement for the development of LBS applications. For that reason, alternative techniques have been adapted to work on those environments, such as: stereo or monocular visual information [4], [5], ultrasonic sensors [6], [7] laser sensors [8] and Inertial Measurement Units (IMU) [9], [10], [11], [12], all of these techniques have advantages and disadvantages related to accuracy, availability of technology, feasibility of installation, etc. This work focuses on the Pedestrian Dead Reckoning (PDR) algorithms which provide an alternative solution as it can work in outdoor and indoor environments.

Pedestrian Dead Reckoning (PDR) is a navigation technique that provides and maintains the geographical position for a person travelling on foot by using self-contained sensors. The current advancements of personal mobile devices with low cost MEMS (*Microelectromechanical Systems*) sensors such as accelerometers, gyroscopes and magnetometers, have made PDR a relevant approach for the development of LBS applications, with potential use in GPS-denied environments.

In PDR algorithms, estimation of each new position is based on the previous one derived from the last step, taking advantage of the sequential nature of pedestrian motion. In general, a PDR algorithm is composed of three parts: step detection, estimation of walking

distance and tracking of sensor attitude. The accelerometer signals are used to detect steps and estimate step length, which in turn is calculated based on a walking step frequency. Attitude can be estimated by combining the measurement signals from the accelerometer, gyroscope and magnetometer included in an inertial measurement unit (IMU). Figure 1.1 presents the general block diagram of the basic PDR algorithm.

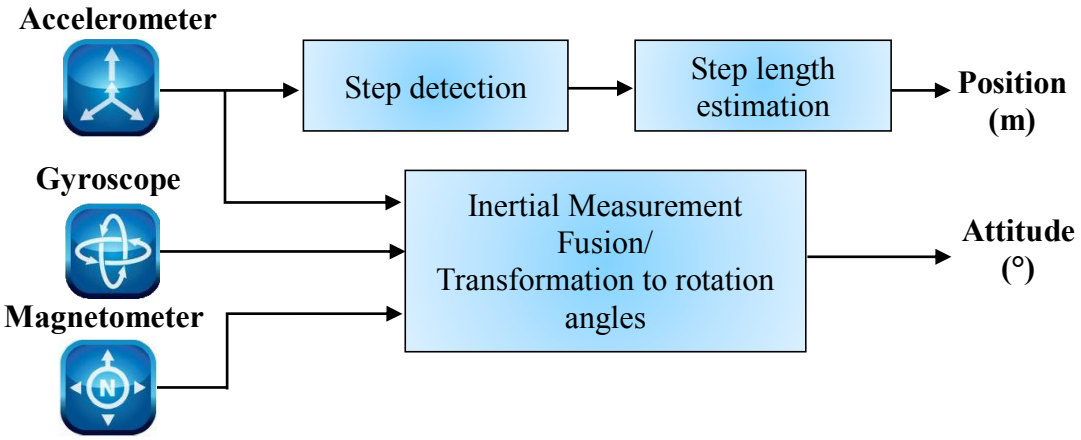


Figure 1.1. Pedestrian Dead Reckoning.

1.2 Problem definition

Most of PDR approaches consider low cost MEMS accelerometers, gyroscopes and magnetometers as the source of information for the PDR algorithm. However, this type of sensors are affected by sensor noise and drift, which introduce errors in the displacement and relative attitude changes in the sensor´s frame of reference with respect to the human body [12].

On the other hand, PDR systems utilizing a magnetometer are not the best choice for indoor/urban navigation because magnetometers are subject to strong magnetic disturbances such as power lines, computers and different metal/steel objects. Therefore, if accurate navigation is required, the system has to rely on other means to determine attitude, such as gyroscopes and accelerometers, according to the angle or position representation.

According with this, due to the drift error of inexpensive gyroscopes and accelerometers and the vulnerability of magnetometers to several types of errors, it is still a challenge to obtain an accurate attitude and achieve a satisfactory positioning accuracy. In order to improve the accuracy of the attitude estimation by reducing the time-varying drift, the distance traveled calculation and consequently the localization accuracy, the present thesis work proposes a PDR algorithm which incorporates additional techniques as inertial measurement fusion and pedestrian activity classification.

PDR algorithms have no restrictions on the location of sensor placed on the body, for that reason, some works incorporate sensors mounted on the pedestrian's shoes to reduce the drift error of gyroscopes, making use of a step's stance phase for the zero velocity update (ZUPT) [12], [13], [14]. In this work, it has been proposed the use of waist-mounted accelerometers and gyroscopes. This decision has been made considering future applications, where the proposed algorithm could be implemented on a mobile platform, such as a smartphone or tablet.

1.3 Objectives

1.3.1 General objective

The main objective of this thesis is to design and develop adaptive techniques for estimating position and attitude in a Pedestrian Dead Reckoning system, in GPS-denied indoor environments, based on inertial measurements fusion and pedestrian activity classification.

1.3.2 Particular objectives

a) To develop a Neuro-Fuzzy tuning based on adaptive Kalman Filter for estimating and reducing the sensors drift and consequently to improve the attitude calculation. The adaptation process is performed on the filter statistical information matrix *R*.

b) To incorporate activity classification on Fuzzy Inference System (FIS) for step length estimation in order to improve the accuracy of the distance traveled calculation, and improving the localization accuracy.
c) To perform MATLAB® simulations of the above techniques for position and orientation estimation.
d) To develop a Pedestrian Dead Reckoning system incorporating the above adaptive techniques using inertial measurements obtained from a three-axis accelerometer and a three-axis gyroscope contained into an Inertial Measurement Unit (IMU) attached to the waist of the wearer.

1.3.3 Solution scope

The solution will be developed in MATLAB® code. The smartphone or tablet implementation is considered as future work.

1.4 Background

Dead reckoning is a relative navigation technique, which calculates the distance travelled by a moving platform starting from a known position and successively adding displacements. Since the past position estimates are projected through time to obtain new estimates in dead reckoning, position errors accumulate over time. Because of this cumulative error propagation, dead-reckoning estimates are unreliable if calculated over long periods of time. Hence, dead reckoning is seldom used alone in practice and is often combined with other types of position sensing to improve position accuracy [15].

Dead reckoning is employed in mobile robotics through the use of odometry and/or inertial navigation systems (INS). Inertial navigation is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of an object relative to a known starting point, orientation and

velocity [16]. Fundamentally, gyroscopes provide angular rate information and accelerometers provide velocity rate information.

INS [16] can be used for both indoor and outdoor positioning and navigation. Although the information provided by inertial sensors is reliable over long periods of time, it must be integrated to provide position, orientation and velocity estimates. Thus, even very small errors in the information cause unbounded growth in the error of the integrated measurements. As a consequence, an INS by itself is characterized by position errors that grow with time and distance, usually referred to as the *drift error* or *bias error* [15].

An inertial measurement unit (IMU) typically contains three orthogonal rate-gyroscopes and three orthogonal accelerometers, measuring angular velocity and linear acceleration respectively. Recent advances in the construction of Microelectromechanical Systems (MEMS) devices have made it possible to manufacture small and light inertial navigation systems. If the IMU is directly mounted on the moving object, the system is called a *strap-down* INS [16].

A basic block diagram of a strap-down INS is given in figure1.2. To keep track of orientation the raw signals from the gyroscopes are integrated. Then, using the estimated orientation, accelerometer outputs should be transformed to the Earth coordinate frame. Thus, to track position is necessary to execute double integration of the acceleration values in Earth coordinate frame.

Because of the integration operations involved in the position calculation, any error in the sensor outputs accumulates in the position output, causing a rapid *drift* in both the gyroscope and accelerometer outputs. Thus, the reliability of position estimates decreases with time. For this reason, inertial sensors are usually used in conjunction with other sensing systems that provide absolute external reference information [15].

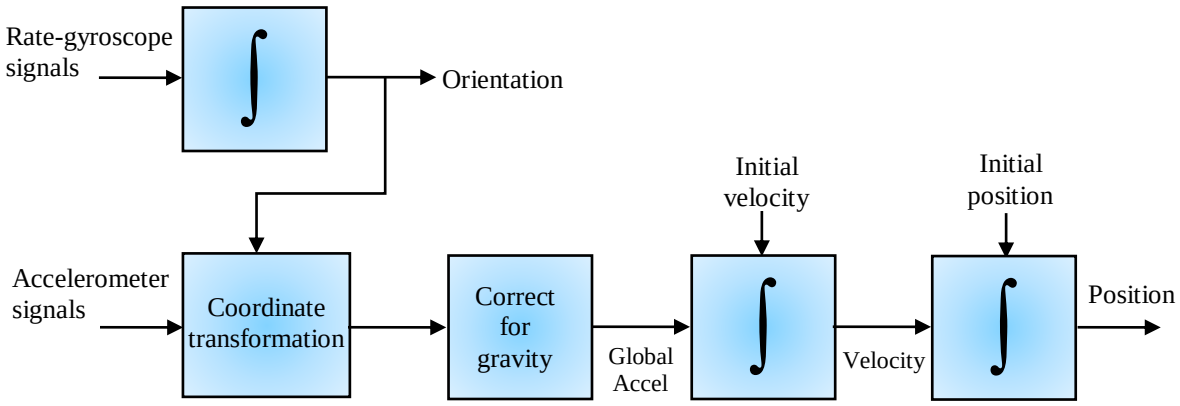


Figure 1.2. Strap-down inertial navigation algorithm [16].

1.4.1 Pedestrian Dead Reckoning

One application of INS is found in Pedestrian Dead Reckoning (PDR). PDR systems are generally used in GPS-denied environments such as inside buildings, tunnels, underground or dense forests and around tall buildings in urban areas where GPS data are not accurate or always available. PDR mechanization exploits the kinematics of human walking. As it was mentioned in the previous chapter, a PDR algorithm is composed of three parts: step detection, estimation of walking distance and tracking of sensor attitude. The following sections provide a brief description of the methods used to perform these tasks reported in the literature.

1.4.1.1 Step detection

There are two basic methods to detect pedestrian steps based on the measurements obtained from three orthogonal accelerometers: peak detection [17] and zero-crossing detection [10]. These methods compare the acceleration values with the predefined thresholds and take the minimum step period into account.

The *peak detection* implies the detection of the peaks of the vertical acceleration because the vertical acceleration is generated by vertical impact when the foot hits the ground. However, peak detection is not appropriate due to that peaks of acceleration also occur in irregular motions, such as turning when avoiding obstacles. It is hard to distinguish between these peaks and those measured during regular walking, thus step misdetections are prone to occur [18].

The *zero-crossing detection* counts signal crossing zero level to determine the occurrence of a step. Researchers usually have used time interval thresholding to reject false step detection. The problem comes when time interval between footfalls varied for some subjects, so it is quite difficult to detect step accurately using zero-crossing method without a calibration process. Figure 1.3 shows the difference between peak and zero-crossing detection.

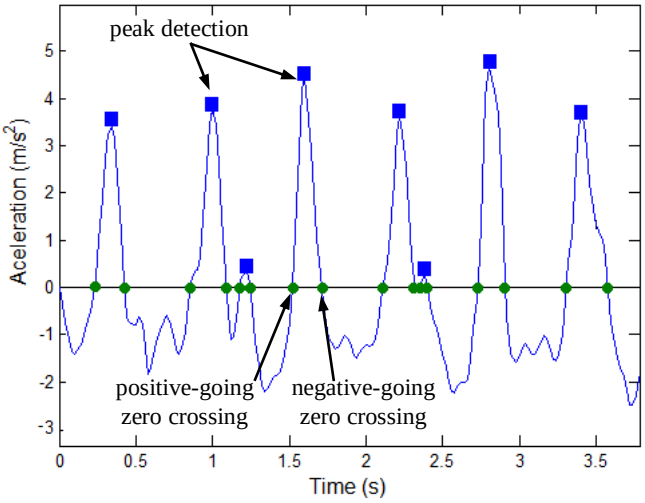


Figure 1.3. Peak and zero crossing as step detection method.

1.4.1.2 Step length estimation

In order to calculate the distance travelled it is fundamental to estimate the step length. Subsequently, to estimate the step length (or step size) the step frequency needs to be calculated and the relationship between step length and step frequency needs to be defined. The most used method to determine the frequency content of a signal is the Fast Fourier

Transform (FFT). Levi and Judd [9] used the FFT to determine the step frequency in the PDR system they proposed. However, to obtain an accurate step frequency solution based on the FFT a large number of acceleration samples are needed.

It is known that regular human walking motion produces periodic variations in vertical acceleration. Therefore, when humans move faster, for stability reasons they stretch out their legs further. This means that as a pedestrian walks faster, both the step length and step frequency increases [19]. Hence, to increase the precision of the distance travelled, the step length parameter must be determined continuously during the walk. Levi and Judd [9] argued that the step length could be estimated based on a linear relationship between the measured step frequency and step length.

Another method called *zero velocity update* (ZUPT) is used to estimate the distance travelled by a pedestrian, without any external reference sensor. The ZUPT method exploits the fact that during normal walking cycles, a foot touches the ground almost periodically and stays on the ground for a short time (usually about 0.1–0.3 s), which is called the *zero velocity interval*. In ZUPT algorithm, this zero velocity interval is detected and thus the velocity error is reset to zero [12]. The principal disadvantage in this method is that the IMU, in most cases must be mounted on the foot, as shown in figure 1.4, which is not applicable to waist-mounted devices.

In the other hand, the step length is a time-varying process which is strongly correlated to the velocity and the step frequency of the pedestrian [17]. For that reason some approaches, as the used in the present work, have integrated pedestrian activity classification in order to improve the accuracy of the distance traveled calculation [15], [17]. The accelerometers are widely used in human activity classification due to that they respond to both, acceleration due to gravity and acceleration due to body movement [21].

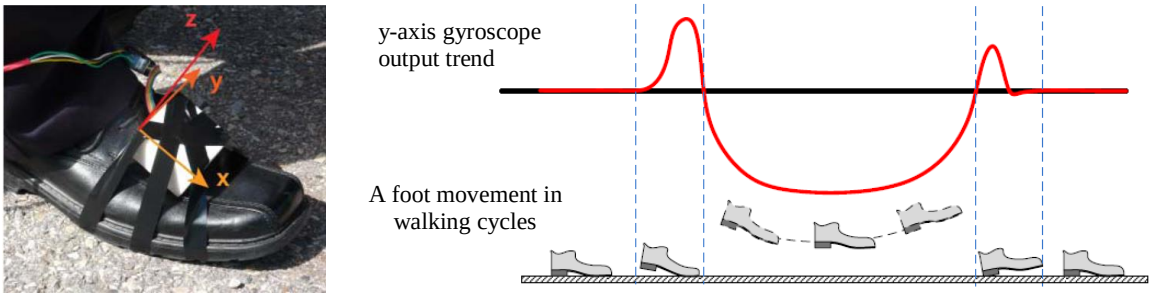


Figure 1.4. Foot-mounted inertial measurement unit [20] for detecting zero velocity interval [12].

Effective algorithms are required to interpret the accelerometer data in the context of different activities. The most common approaches to automatic classification of human activity are based on machine learning techniques, especially Hidden Markov Models (HMMs) [22], [23]. The use of Hidden Markov Models (HMMs) is attractive, although they are known for being methods potentially plagued by severe difficulties of parameter estimation. These techniques typically operate via two-stage process: first, features are obtained from sliding windows of accelerometer data and then a classifier is used to identify the activity [24].

A range of different approaches has been used to obtain features from accelerometer data, some works obtain features directly from the time-varying acceleration signal [25], [26] and others from frequency analysis [17]. More recently, wavelet analysis has been used to derive time-frequency features [24], [27], [28]. Table 1.1 presents a brief summary of the recent different techniques to perform step detection, step length estimation and pedestrian activity classification reported in the literature.

In search for integrating the classification algorithm into a PDR system, this work proposes a dynamical method for estimating the step length using a Fuzzy Inference System (FIS) which use the pedestrian activity as one of its input.

Table 1.1. A brief summary of methods used to step detection, step length estimation and pedestrian activity classification using triaxial accelerometer.

Ref	Task	Contribution	Purpose
[15]	Activity classification	This work present experimental results in 2D and 3D obtained from classifying three activities: walking, standing and turning activities. In order to recognize those activities are used sensors (3-axial accelerometer, 3-axial gyroscope and a 3-axial magnetomer) located on the right and left legs. In the 2D case it is used a rule-based classifier and for the 3D case is analyzed stairs activity using a K-nearest neighbors (k-nn) classifier.	PDR system
[20]	Step length estimation	The steps are detected using the relationship between arm's and foot's motion, from the swing of the subject's arm the step events can be detected. They use two IMUs located on the foot and the hand.	Mobile phone applications
[22], [29]	Activity classification	The classification method is based on the hidden Markov model (HMM) where the hidden states are in fact different activity classes or gait phases: walking, running, standing, going up stairs and going down stairs.	Gait Analysis
[26]	Activity classification	This work presents three classifications techniques: decision trees, logistic regression and multilayer neural networks. The classified activities are: walking, jogging, climbing stairs, sitting, and standing.	Automatic customization of the mobile devices
[30]	Step detection/ step length estimation	A combination of zero-crossing, peak detection and time restriction is used to identify the steps. The step length is estimated using a linear model between the step length and the step frequency.	PDR system
[31]	Step length estimation	The step length estimation is conducted with an empirical model, which uses personal constants as well as the step frequency.	PDR system
[32]	Step detection	The steps are identified when the acceleration norm crosses *g* (gravitational acceleration) since it is followed by a rise rate and a peak height that exceed a threshold.	PDR system

[33]	Step length estimation	The step length is estimated using the double integral of acceleration and is used zero velocity update (ZUPT) to calibrate the sensor data.	PDR system
[34]	Step detection/ step length estimation	The detection of successive steps is based on a peak and threshold analysis with the energy of the acceleration signal. The step length is calculated from the maximum and minimum measured acceleration in z-direction.	PDR system
[35]	Activity classification	HMM as classification method. The activities are: walking, running, standing, stair climbing, sitting, lying and cycling.	Healthcare monitoring devices
[36]	Activity classification	This work presents activity and environment classification using a foot mounted device with IMUs and a GPS receiver, and is proposed an algorithm that classifies the following activities: stationary, crawling, walking, running, biking, moving in vehicle, level up or down elevator and up or down stairs. Multiple probability density functions that map each feature (i.e. metric) to an activity are provided, and a naive Bayesian probabilistic model is used to determine the probability of each activity.	Foot mounted navigation systems

1.4.1.3 Attitude estimation

Attitude can be estimated by combining the measurement signals from the accelerometer, gyroscope and magnetometer included in an inertial measurement unit (IMU). Attitude estimation methods in pedestrian navigation are classified as quaternion method [37], [38], direction cosine matrix method [39], [40], or Euler angles method [41], [42].

The Euler angles method has the merit of being a more meaningful attitude expression than either the quaternion method or the direction cosine matrix method, and the user can recognize the attitude directly [43]. Kalman-based sensors fusion has been extensively used to pursue attitude estimation through inertial measurement units in navigation systems [44], [45], [46]. A problem with Kalman filter formulation is that it requires a priori knowledge of the process and measurement noise statistics. Furthermore, inadequate initial statistics of the

filter will reduce the precision of the estimated states or will introduce biases to the estimates. Adaptive Kalman Filter (AKF) can be used to deal with such a problem [47], [48], [49].

Previous studies have investigated the fuzzy and neuro-fuzzy adaptation of Kalman filtering [49], [50], [51], and they have used covariance-matching based approaches to perform the process of adapting the statistics of the filter in mobile applications. In [49] and [50] a fuzzy adaptive Kalman Filter algorithm is applied to an Attitude and Heading Reference System (AHRS). Usually, an AHRS consists of MEMS gyroscopes, accelerometers and magnetometers providing three axes signals. The authors use magnetometers to provide a measure of the attitude of the mobile device. By detecting the magnetic field information, they can compensate the errors of attitude caused by drifts of MEMS gyros and accelerometers.

However, the use of magnetometers is not appropriate in PDR systems, where the indoor/urban navigation is essential and the magnetometers can be easily disturbed. In [51] the authors report a two-input ANFIS approach similar to our proposal, with the difference that they do not use inertial measurements, and therefore a bias correction is not required.

The main problem when using inertial sensors to obtain orientation information of a moving platform is the accumulation of errors due to the bias and off-set inherently present in those sensors. As stated previously, several adaptive Kalman filter formulations can be found in the literature to deal with the orientation estimation problem. However, none of them seems to estimate the bias present in the inertial measurements.

Therefore, the main purpose of this work is to develop an adaptive technique for the estimation and reduction of the inertial sensor bias. This in turn will reduce the error in the estimation of the orientation applied for pedestrian navigation.

1.5 Thesis organization

The organization of this document is as follows: Chapter 2 presents an information compilation regarding the topics related to the system developed in this work. Chapters 3 and 4 describe the proposed methodology. Chapter 5 present the experimental results. Finally, conclusions and future work are discussed in Chapter 6.

Chapter 2

Theoretical framework

In order to describe the PDR system, this chapter presents an introduction about the topics related to the system developed in this thesis work.

2.1 Kalman filtering

The Kalman Filter algorithm was first published in 1960 by R. E. Kalman. In his famous paper, Kalman described a recursive solution to the discrete data linear filtering problem [52]. Since then, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. Kalman-based sensors fusion has been extensively used to pursue attitude estimation through inertial measurement units in navigation systems [44], [45], [46].

2.1.1 The Kalman filter algorithm

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown [53].

The Kalman filter addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation:

$$x_{k+1} = A_k x_k + Bu_k + w_k \tag{2.1}$$

with a measurement $z \in \Re^m$ that is

$$z_k = H_k x_k + v_k \tag{2.2}$$

where [54]:

- x_k = (n×1) state vector at time k.
- A_k = (n×n) state transition matrix.
- B_k = (n×l) matrix that relates the control input $u_k \in \Re^l$ to the state vector x_k.
- u_k = (l×1) vector of the input forcing function.
- w_k = (n×1) process noise vector.
- z_k = (n×1) measurement vector at time k.
- H_k = (m×n) matrix giving the ideal connection between the measurement and the state vector at the time k.
- v_k = (n×1) vector of additive measurement noise.

The random variables, w_k and v_k, represent the process and measurement noise, respectively. They are assumed to be uncorrelated zero-mean Gaussian white noise sequences with covariances:

$$E\left\{w_k\, w_i^T\right\} = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \tag{2.3}$$

$$E\left\{v_k\, v_i^T\right\} = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \tag{2.4}$$

$$E\left\{w_k\, v_i^T\right\} = 0 \qquad \text{for all } k \text{ and } i \tag{2.5}$$

where $E\{\cdot\}$ is the statistical expectation operator, Q_k is the process noise covariance matrix, and R_k is the measurement noise covariance matrix.

In accord with Welch and Bishop [53], the Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: *time update* equations and *measurement update* equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measurement update equations are responsible for the feedback, i.e. for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate.

The time update equations can also be thought of as *predictor* equations, while the measurement update equations can be thought of as *corrector* equations. Therefore, the specific Kalman filter equations are organized into two groups.

1) Time update (or prediction) equations

$$\hat{x}_{k+1}^{-} = A_k \hat{x}_k + Bu_k \tag{2.6}$$

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

The time update equations project the current state and error covariance estimates, from time step k to step k+1, to obtain *a priori* estimates, denoted by a super minus ($^-$), for the next time step.

2) Measurement update (or correction) equations

$$K_k = P_k^{-} H^T \left(H P_k^{-} H^T + R \right)^{-1} \tag{2.8}$$

$$\hat{x}_k = \hat{x}_k^{-} + K_k \left(z_k - H \hat{x}_k^{-} \right) \tag{2.9}$$

$$P_k = (I - K_k H) P_k^- \tag{2.10}$$

These equations incorporate a new measurement into the *a priori* estimates to obtain the improved *a posteriori* estimates. The first task during the measurement update is to compute the Kalman gain, K_k. The next step is to actually measure the process to obtain, and then to generate an *a posteriori* state estimate by incorporating the measurement as in (2.9). The final step is to obtain an *a posteriori* error covariance estimate via (2.10).

After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This recursive nature is one of the very appealing features of the Kalman filter. A graphical representation of the Kalman filter algorithm is presented in figure 2.1.

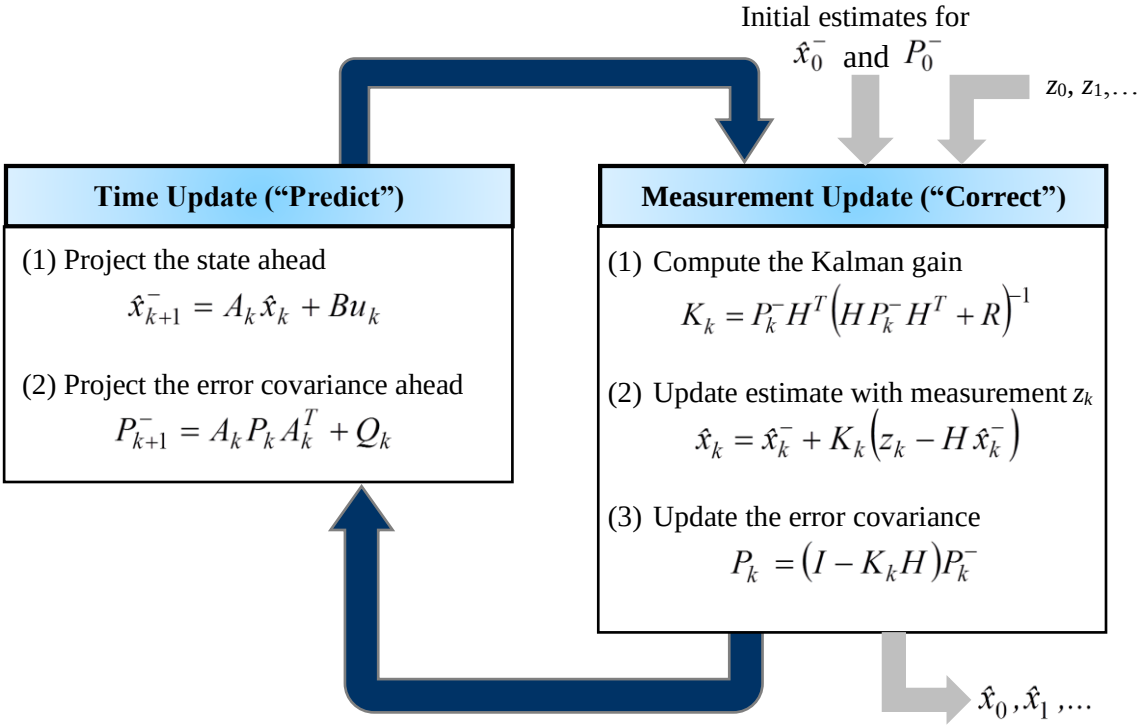


Figure 2.1. Graphical representation of the Kalman filter algorithm [53].

In the measurement update equations, $\hat{x}_k$ is an estimate of the system state vector x_k and P_k is the covariance matrix corresponding to the state estimation error defined by:

$$P_k = E\left\{\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^T\right\} \tag{2.11}$$

In equation (2.9) the term $H\hat{x}_k^-$ is referred to as the one-stage predicted measurement, which is the best prediction of what the measurement at time *k* will be before it is actually taken. The difference between the actual measurement z_k and its one-stage prediction is called the measurement residual or innovation sequence, r_k or Inn_k [54], defined as:

$$r_k = Inn_k = \left(z_k - H\hat{x}_k^-\right) \tag{2.12}$$

The innovation represents the additional information available to the filter in consequence to the new observation z_k. For an optimal filter the innovation sequence is a sequence of independent Gaussian random variables. The weighted innovation, $K_k\left(z_k - H\hat{x}_k^-\right)$, acts as a correction to the predicted estimate $\hat{x}_k^-$ to form the estimation $\hat{x}_k$

2.2 Neuro-Fuzzy systems

There are two concepts that are inherent to the human reasoning: imprecision and uncertainty. Because of that, our perception of the real world is pervaded by concepts which do not have sharply defined boundaries, for example: *many*, *tall*, *much larger than*, *young*, etc. These concepts are true only to some degree and they are false to some degree as well, therefore, this way of thinking is not captured in traditional logic and traditional computing. This fact has been perceived by several thinkers that in the past have tried to develop a mathematical structure capable of capturing this characteristic of the human way of thinking [54], [55].

The Fuzzy Logic theory was introduced in 1965, by Lotfi Zadeh, and is a mathematical tool for dealing with uncertainty. Fuzzy logic provides an inference morphology that enables approximate human reasoning capabilities to be applied to knowledge-based systems. The theory of fuzzy logic provides a mathematical strength to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning.

Artificial Neural Networks can be considered as simplified mathematical models of brain-like systems and they function as parallel distributed computing networks. However, in contrast to conventional computers, which are programmed to perform specific tasks, most neural networks must be taught, or trained. They can learn new associations, new functional dependencies and new patterns [56].

Neuro-fuzzy systems integrate two complementary approaches: fuzzy logic and neural networks. Neural networks are capable of recognizing patterns and adapting themselves to cope with changing environments; if there is data available, or if it can be learned from a simulation or real task, then a neural network can be used. Fuzzy inference systems incorporate human knowledge and perform inference and decision making; if there is knowledge that can be expressed in rules, then a fuzzy system can be built [54].

2.2.1 Fuzzy Inference Systems

The main purpose of fuzzy logic is to allow the use of vague concepts to characterize the variables of a system and its interrelations using words or propositions expressed in a natural or artificial language. This is possible because, in fuzzy sets theory, an object is no longer restricted to be totally a member or not a member of a set. Instead, an element can have a grade of membership intermediate between full membership and non-membership, in the whole range [0-1]. Therefore, using fuzzy logic, systems can be designed to be able to capture, in the form of heuristic rules, the ability that all human beings possess to model a system or process using natural language [54], [55].

A *fuzzy inference system* (FIS), also known as fuzzy rule-based system, is a computing framework based on the concepts of fuzzy logic. A FIS provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. The basic structure of a FIS, with five functional blocks, is shown in figure 2.2. The function of each block is as follows [57]:

- a *rule base* containing a number of fuzzy IF–THEN rules;
- a *database* which defines the membership functions of the fuzzy sets used in the fuzzy rules;
- a *decision-making unit* which performs the inference operations on the rules;
- a *fuzzification interface* which transforms the crisp inputs into degrees of match with linguistic values; and
- a *defuzzification interface* which transforms the fuzzy results of the inference into a crisp output.

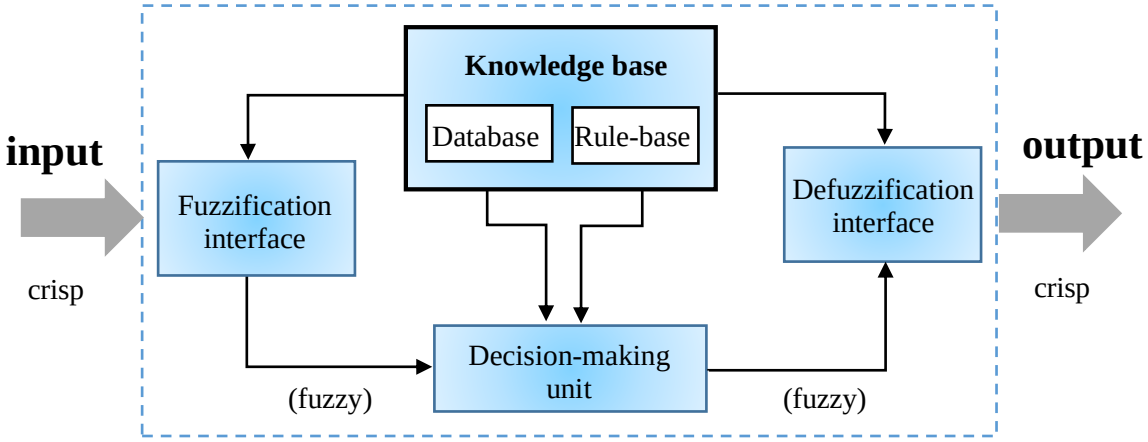


Figure 2.2. Fuzzy inference system [57].

The performance of a FIS is as follow: the crisp input is converted into fuzzy values by using a fuzzification method. After fuzzification the rule base is evaluated in the decision-making unit by fuzzy reasoning. The rule base and the database are jointly referred to as the *knowledge base*. A defuzzification method is used to convert fuzzy values to the real world values which are the FIS outputs.

The steps of *fuzzy* reasoning, i.e. the inference operations upon fuzzy IF–THEN rules, performed by a FIS are:

1. Compare the input variables with the membership functions on the antecedent part to obtain the membership values in each fuzzy set denoted by a linguistic label. This step is the fuzzification process
2. Combine the membership values on the premise part to get the *firing strength* (*weight*) value for each rule.
3. Generate the qualified consequents (either fuzzy or crisp) of each rule depending on the firing strength.
4. Aggregate the qualified consequents to produce a crisp output. This step is the defuzzification process.

2.2.1.1 Fuzzification

In order to perform inferences inside the FIS, using an inference engine and fuzzy operators, it is necessary to transform the real-valued input information into fuzzy sets. This transformation is carried out through a process known as *fuzzification*. In general the purpose of fuzzification is to map the inputs to values from 0 to 1 using a set of input membership functions.

Specifically, if x is an input variable to the FIS, and $x = x_0 \in U$ is an input value, then the output of the process of fuzzification is a fuzzy set in U, $F = fuzzifier(x_0)$; where the operator *fuzzifier* transforms the real input value x_0 to a linguistic value or fuzzy set, F. In FIS, *singleton fuzzification* is the most utilized. This method maps the input x to a fuzzy singleton, F, with membership function:

$$\mu_F(x) = \begin{cases} 1 & \text{if } x = x_0 \\ 0 & \text{in any other case} \end{cases} \tag{2.13}$$

2.2.1.2 The inference mechanism

The inference mechanism is an interpreter of the rule base, the task of the inference mechanism is the calculation of a fuzzy conclusion from a set of fuzzy IF-THEN rules and one or more conditions. The fuzzy conclusion is obtained employing an inference mechanism called *approximate reasoning*, derived from fuzzy logic theory. A fuzzy rule base is integrated by a set of linguistic rules expressed in the form IF-THEN. The part IF of the rule is known as the antecedent, and the part THEN of the rule is known as the consequent. Suppose two input variables x and y, a fuzzy control rule is:

$$\Re_i\text{: IF } (x \text{ is } A_i \text{ and } y \text{ is } B_i) \text{ THEN } (z \text{ is } C_i) \tag{2.14}$$

where $i = 1, 2, \ldots, n$; n = number of rules; x, y and z are linguistic variables; A_i, B_i and C_i are linguistic values of the linguistic variables x, y and z in the universes of discourse U, V and W, and characterized by the membership functions $\mu_{A_i}(x)$, $\mu_{B_i}(x)$ and $\mu_{C_i}(x)$, respectively.

Now, suppose that the rule base (2.14) includes a single rule written as "IF x is A and y is B THEN z is C":

Premise 1 (fact):	x is A' and y is B'
Premise 2 (rule):	IF x is A and y is B THEN z is C
Consequence (conclusion):	z is C'

where A' is a fuzzy set close to A and B' is a fuzzy set close to B. The rule in premise 2 is implemented by a *fuzzy implication* (or relation) R_i and is defined as:

$$R = A \times B \rightarrow C \quad \in \quad U \times V \times W \tag{2.15}$$

which membership function is specified by [58]:

$$\begin{aligned} \mu_R(x,y,z) &= \mu_{(A\times B)\rightarrow C}(x,y,z) \\ &= \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z) \end{aligned} \tag{2.16}$$

where the symbol $\wedge$ is used to denote the fuzzy operator of intersection (*min*), or fuzzy AND. Applying the compositional rule of inference [59], the fuzzy conclusion C' of the inference procedure is expressed as:

$$\begin{aligned} C' &= (A' \times B') \circ R \\ &= (A' \times B') \circ (A \times B \rightarrow C) \end{aligned} \tag{2.17}$$

where $\circ$ denotes the composition operator. Thus, using (2.16), the membership function of C' is evaluated as:

$$\begin{aligned} \mu_{C'}(z) &= \vee_{x,y}[\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)] \\ &= \vee_{x,y}[\mu_{A'}(x) \wedge \mu_{B'}(y) \wedge \mu_A(x) \wedge \mu_B(y)] \wedge \mu_C(z) \\ &= \{\vee_x [\mu_{A'}(x) \wedge \mu_A(x)]\} \wedge \{\vee_y [\mu_{B'}(y) \wedge \mu_B(y)]\} \wedge \mu_C(z) \end{aligned} \tag{2.18}$$

where the symbol $\vee$ is used to denote the fuzzy operator of union (*max*), or fuzzy OR. Then:

$$w_A = \vee_x [\mu_{A'}(x) \wedge \mu_A(x)] \tag{2.19}$$

$$w_B = \vee_y [\mu_{B'}(y) \wedge \mu_B(y)] \tag{2.20}$$

where w_A represents the degree of compatibility between A and A'; similarly, w_B represents the degree of compatibility between B and B'. Substituting (2.19) and (2.20) in (2.18) results:

$$\mu_{C'}(z) = w_A \wedge w_B \wedge \mu_C(z) \tag{2.21}$$

but, if it is defined $w = w_A \wedge w_B$, then (2.21) transforms to:

$$\mu_{C'}(z) = w \wedge \mu_C(z) \tag{2.22}$$

In (2.22) w is called the *firing strength* of this rule, and it represents the degree to which the antecedent part of the rule is satisfied [58]. A graphical interpretation of this result is shown in figure 2.3 when the fuzzy operators for union and intersection are selected to be the maximum (*max*) and minimum (*min*), respectively. In this case the symbol '$\circ$' is called

the *max-min* composition operator [60], and the whole inference procedure is called the *max-min compositional rule of inference* [61]. Observe in figure 2.3 that the resulting membership function for *C´* is equal to the membership function of *C* clipped by the firing strength *w*.

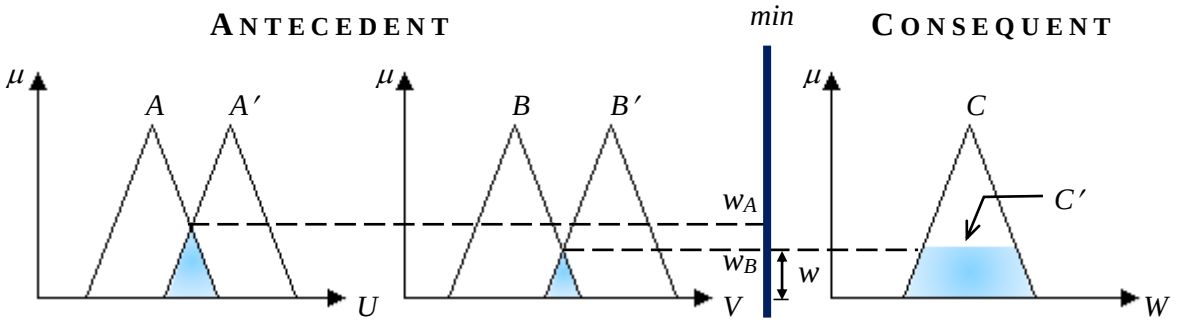


Figure 2.3. Approximate reasoning for a rule with two antecedents [54].

The previous development can be extended for the case of a rule base with *n* rules [54]:

fact:	*x* is *A*' and *y* is *B*'
$\Re_1$:	IF *x* is A_1 and *y* is B_1 THEN *z* is C_1
$\Re_2$:	IF *x* is A_2 and *y* is B_2 THEN *z* is C_2
	…
	…
$\Re_n$:	IF *x* is A_n and *y* is B_n THEN *z* is C_n
Conclusion:	*z* is *C'*

Then, each rule can be implemented as a fuzzy relation:

$$R_i = A_i \times B_i \rightarrow C_i \qquad \text{for } i = 1, 2, ..., \ n \tag{2.23}$$

Applying the *max-min* compositional rule of inference, the fuzzy conclusion *C´* of the inference procedure is expressed as:

$$\begin{aligned} C' &= (A' \times B') \circ (R_1 \vee R_2 \vee \ldots \vee R_n) \\ &= [(A' \times B') \circ R_1] \vee [(A' \times B') \circ R_2] \vee \cdots \vee [(A' \times B') \circ R_n] \\ &= C'_1 \vee C'_2 \vee \cdots \vee C'_n \end{aligned} \tag{2.24}$$

where C'_i is the inferred fuzzy set for rule *i*. Then, the membership function of each fuzzy set C'_i is obtained as:

$$\mu_{C'_i}(z) = \left[w_i \wedge \mu_{C_i}(z)\right] \qquad \text{for } i = 1, 2, \ldots, n \tag{2.25}$$

Finally, the membership function of the resulting fuzzy set *C'* inferred from the complete set of fuzzy rules is given by the union of the resulting conclusion derived from individual rules:

$$\mu_{C'}(z) = \left[w_1 \wedge \mu_{C_1}(z)\right] \vee \left[w_2 \wedge \mu_{C_2}(z)\right] \vee \cdots \vee \left[w_n \wedge \mu_{C_n}(z)\right] \tag{2.26}$$

where w_i indicates the degree of fulfillment of the *i*-th rule; $\mu_{C_n}(z)$ is the membership function of the fuzzy set *Ci* (*i* =1, 2, ..., *n*; *n* = number of rules). Figure 2.4 shows a graphical representation of the operation of fuzzy reasoning for the case described. Note, that in this case the singleton fuzzification procedure has been used to transform the inputs x_0 and y_0 into fuzzy sets [55].

The fuzzy reasoning in figure 2.4 employs *max* and *min* operators for fuzzy union and fuzzy intersection, respectively. However, there are others types of fuzzy reasoning or composition operators, the three most used are: *max-min*, *max-product* and *sum-product* [54], [55]. Table 2.1 presents a resume of the types of fuzzy reasoning.

Table 2.1. Types of fuzzy reasoning [54]

Composition operator	**Union operator**	**Intersection operator**
max-min	max	min
max-product	max	product
sum-product	sum	product

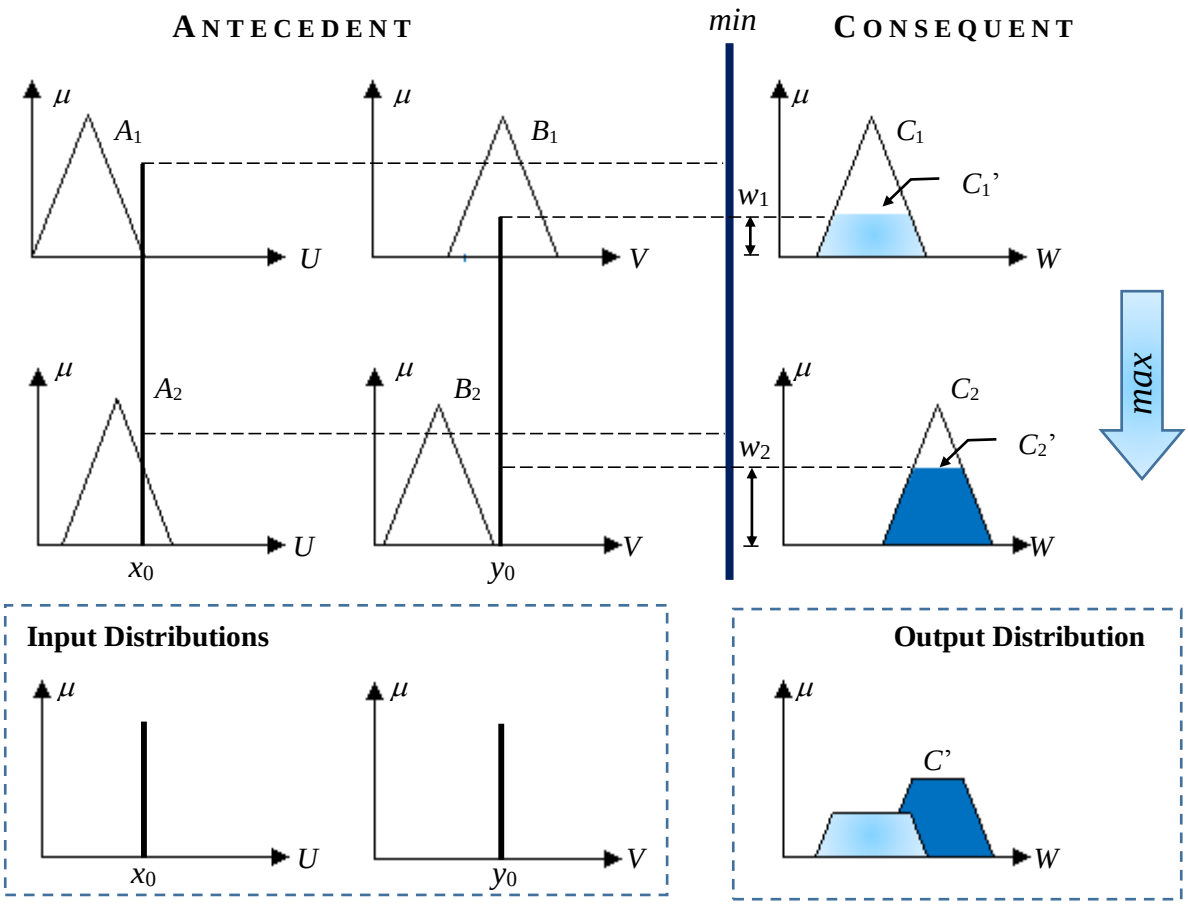


Figure 2.4. Fuzzy reasoning mechanisms, max-min compositional rule of inference [55].

The figure 2.5 presents the fuzzy reasoning when the *max-product* compositional rule of inference is used. The calculation of the fuzzy conclusion for this case is obtained as:

$$\mu_{C'}(z) = \left[w_1 \cdot \mu_{C_1}(z)\right] \vee \left[w_2 \cdot \mu_{C_2}(z)\right] \vee \cdots \vee \left[w_n \cdot \mu_{C_n}(z)\right] \tag{2.27}$$

If the *sum-product* compositional rule is used the fuzzy conclusion is obtained as:

$$\mu_{C'}(z) = \sum_{j=1}^{m} w_j \cdot \mu_{C_j}(z) \tag{2.28}$$

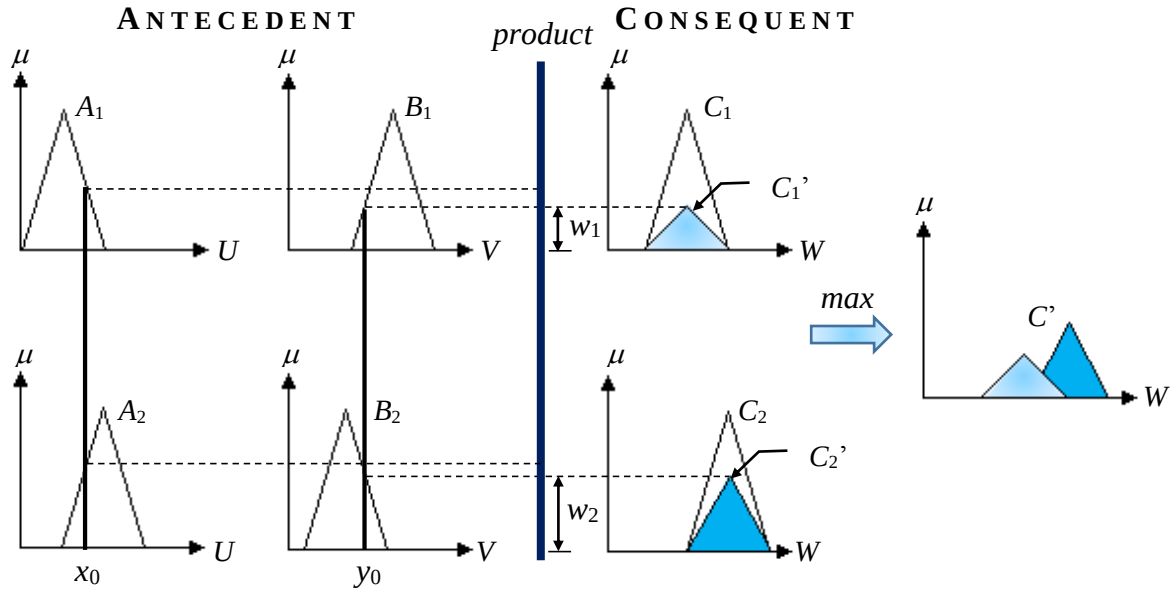


Figure 2.5. Fuzzy reasoning procedure, max-product compositional rule of inference [54].

2.2.1.3 Defuzzification

The defuzzification process is the conversion of a fuzzy set defined over an output universe of discourse into a space of crisp (non-fuzzy) values. In other words, the defuzzification process transforms the fuzzy conclusion C' into a crisp and concrete value z_0, which is given as the FIS output. The most often used defuzzification operators are:

- **Center of Area-Gravity:** this procedure, also called centroid, is the most prevalent and physically appealing of all the defuzzification methods. The centroid defuzzification method returns the center of area under the aggregated output membership function, $\mu_{C'}(z)$. It is given by the algebraic expression:

$$z_0 = \frac{\int_W \mu_{C'}(z)\, z\, dz}{\int_W \mu_{C'}(z)\, dz} \tag{2.29}$$

- **Bisector:** the defuzzified value of a fuzzy set C' is defined as the value where a vertical line divides the set into two sub-regions of equal area.

- **Mean of Maximum (MOM):** the defuzzified value of a fuzzy set C' is defined as a mean of all values of the universe of discourse, having maximal membership grades:

$$z_0 = \frac{\int_{Z'} z\, dz}{\int_{Z'} dz} \tag{2.30}$$

where $Z' = \{z \mid \mu_{C'}(z) = \mu^*\}$. In particular, if $\mu_{C'}(z)$ has a single maximum at $z = z^*$, then $z_0 = z^*$ [54].

- **Smallest of Maximum (SOM):** the SOM defuzzification method gives as crisp output the minimum, in terms of magnitude, of the maximizing z.
- **Largest of Maximum (LOM):** the LOM defuzzification method gives as crisp output the maximum, in terms of magnitude, of the maximizing *z*.

The figure 2.6 shows a graphic comparison of the different FIS crisp outputs obtained with each defuzzification method for a given fuzzy set *C*'.

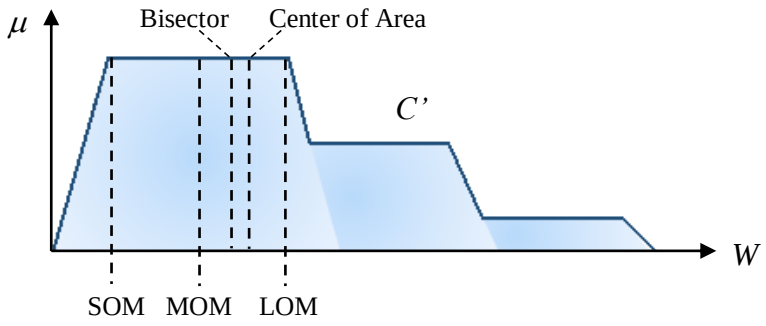


Figure 2.6. Defuzzification methods [54].

2.2.1.4 Types of Fuzzy Inference Systems

There are two principal models of fuzzy inference systems, *Mamdani Fuzzy Inference model* and *Takagi-Sugeno Fuzzy Inference model*, the main difference between the two models lies in the consequent part of the fuzzy rules. Mamdani fuzzy systems use fuzzy sets

as rule consequents whereas Takagi-Sugeno fuzzy systems employ linear functions of input variables as rule consequents [55].

1) Mamdani Fuzzy Inference model: is the most important fuzzy inference method. It was proposed by Mamdani and Assilian [62] as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. The distinctive characteristic of this type of FIS is that in both antecedent and consequent parts of the rules, the values of the variables used are defined by membership functions. Examples of Mamdani fuzzy inference systems are shown in figures 2.4 and 2.5, and they were discussed in the previous section.

2) Takagi-Sugeno Fuzzy Inference model: is so-called *Sugeno* or *Takagi–Sugeno–Kang Fuzzy Inference model*, this system was introduced by Takagi, Sugeno and Kang [63], in an effort to formalize a system approach to generating fuzzy rules from an input–output data set. The main characteristic of this type of FIS is that the fuzzy rules used have the form:

$$\text{IF } x \text{ is } A \text{ and } y \text{ is } B \text{ THEN } z = f(x,y) \tag{2.31}$$

where x, y and z are linguistic variables, A and B are fuzzy sets in the antecedent, and $z = f(x,y)$ is a crisp function in the consequent, but it can be any other function that can appropriately describe the output of the system within the fuzzy region specified by the antecedent of the rule. When $f(x, y)$ is a first-order polynomial, we have the *first-order* Sugeno fuzzy model. When f is a constant, we then have the *zero-order* Sugeno fuzzy model, which can be viewed either as a special case of the Mamdani FIS where each rule's consequent is specified by a fuzzy singleton.

The overall output of a Sugeno FIS model is obtained via a weighted average of the crisp outputs given by the fired rules, as it is graphically represented in figure 2.7 for a first-order Sugeno model.

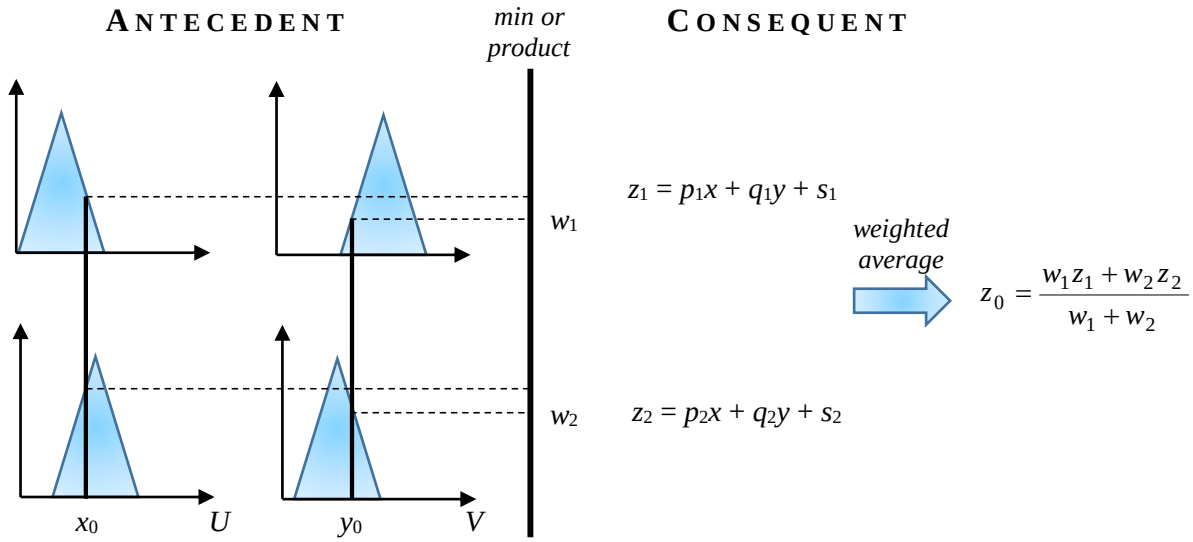


Figure 2.7. A first-order Sugeno fuzzy model [57].

2.2.2 Artificial neural networks

Artificial neural systems can be considered as simplified mathematical models of brain-like systems and they function as parallel distributed computing networks. However, in contrast to conventional computers, which are programmed to perform specific task, most neural networks must be taught, or trained. They can learn new associations, new functional dependencies and new patterns. Although computers outperform both biological and artificial neural systems for tasks based on precise and fast arithmetic operations, artificial neural systems represent the promising new generation of information processing networks [64].

An artificial neural network or simply a neural network (NN) is defined as a collection of processing elements (called neurons) and connection weights (generally denoted as w). These neurons and weights are structured to perform a mapping from an input space to an output space $R^n \rightarrow R^m$ [65]. A NN can have several layers, and each layer can have more than one neuron. Figure 2.8 shows a typical feedforward NN, which in practice is one of the most used.

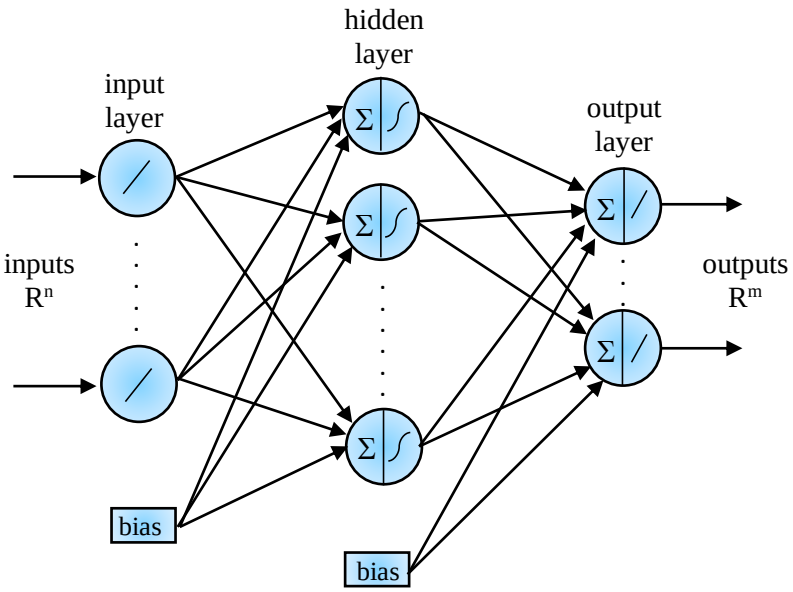


Figure 2.8. A typical feedforward Neural Network [54].

The main function of each neuron in a NN is the collection of all its weighted inputs, an evaluation of a predefined mathematical operation, usually a dot product followed by a non-linear function, and the production of a single output. A neuron with R inputs is shown in figure 2.9. The individual inputs p_1, p_2, …, p_R are each weighted by corresponding elements $w_{1,1}$, $w_{1,2}$, …, $w_{1,R}$ of the *weight matrix* $\mathbf{W}$, the neuron has a bias b, which is summed with the weighted inputs [56]. Mathematically, the neuron output can be written as:

$$y = f\left(\mathbf{W}\mathbf{p} + b\right) \tag{2.32}$$

where f is some activation function [56].

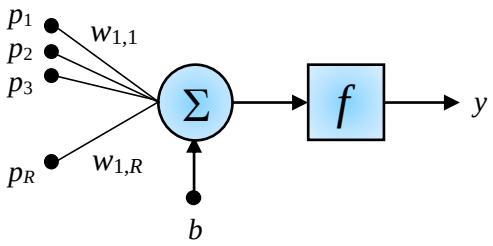


Figure 2.9. Multiple-input neuron [56].

The main characteristic of a NN is the storage of knowledge in the connection weights. This knowledge is acquired using a procedure called the *learning algorithm*. This algorithm successively adjusts the connection weights to find those values for which a better approximation is obtained to the desired output. There are many types of neural network learning algorithms. They fall into three broad categories: supervised learning, unsupervised learning and reinforcement (or graded) learning [56].

In *supervised learning*, the learning rule is provided with a set of examples (the *training set*) of proper network behavior:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\} \tag{2.33}$$

where $\mathbf{p}_Q$ is an input to the network and $\mathbf{t}_Q$ is the corresponding correct (*target*) output. As the inputs are applied to the network, the network outputs are compared to the targets. The learning algorithm is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets [56].

Reinforcement learning is similar to supervised learning, except that, instead of being provided with the correct output for each network input, the algorithm is only given a grade. The grade (or score) is a measure of the network performance over some sequence of inputs. This type of learning is currently much less common than supervised learning. In *unsupervised learning*, the weights and biases are modified in response to network inputs only [54], [56].

The network structure, which includes many neurons and connection weights, is what gives a NN its computational capabilities. The arrangement of neurons in layers or stages of processing is supposed to mimic the layered structure of a certain portion of the human brain. This scheme of multilayer NN (MNN) has tested better computational capabilities than the one with a single layer. In particular, MNNs, which use the error back propagation learning

algorithm, have been successfully applied in several areas including pattern recognition, system identification, and control systems [54], [56].

2.2.3 Adaptive Network based Fuzzy Inference System (ANFIS)

The ANFIS, the abbreviated of *adaptive-network-based fuzzy inference system*, is a fuzzy inference system implemented in the framework of adaptive networks. ANFIS can construct an input–output mapping based on both human knowledge, in the form of fuzzy IF-THEN rules, and stipulated input–output data pairs [57].

To describe ANFIS, consider the first–order Sugeno fuzzy inference system presented in figure 2.7. This system has two inputs x and y and one output z. Suppose that the rule base contains two fuzzy IF–THEN rules of Takagi-Sugeno type:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$

The corresponding equivalent ANFIS architecture to this fuzzy reasoning is illustrated in figure 2.10.

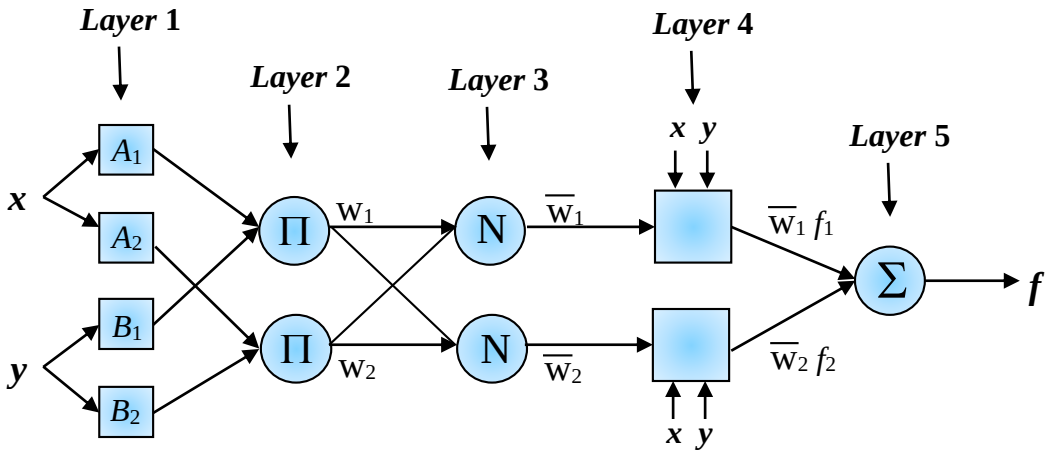


Figure 2.10. ANFIS architecture for a first–order Sugeno fuzzy inference system [57].

The node functions in the same layer are of the same function family as described below [57]:

- ***Layer 1*:** Every node *i* in this layer is a square node with a node function:

$$O_i^1 = \mu_{A_i}(x) \tag{2.34}$$

where *x* is the input to node *i*, and A_i, is the linguistic label (*small*, *large*, *etc.*) associated with this node function. In other words, O_i^1 is the membership function of A_i and it specifies the degree to which the given *x* satisfies the quantifier A_i. Usually, μ_{A_i} is a bell−shaped function with maximum equal to 1 and minimum equal to 0, such as:

$$\mu_{A_i}(x) = \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^2\right]b_i} \tag{2.35}$$

where a_i, b_i and c_i form the parameter set. As the values of these parameters change, the bell-shaped functions vary accordingly, thus exhibiting various forms of membership functions on linguistic label A_i. In fact, any continuous and piecewise differentiable functions, such as commonly used trapezoidal or triangular-shaped membership functions, are also qualified candidates for node functions in this layer. Parameters in this layer are referred to as *premise parameters*.

- ***Layer 2:*** Every node in this layer is a circle node labeled Π which multiplies the incoming signals and sends the product out. For instance,

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2 \tag{2.36}$$

each node output represents the firing strength of a rule.

- ***Layer 3:*** Every node in this layer is a circle node labeled *N*. The *i*th node calculates the ratio of the *i*th rule's firing strength to the sum of all rules' firing strengths:

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2. \tag{2.37}$$

For convenience, outputs of this layer will be called *normalized firing strengths*.

- ***Layer 4:*** Every node *I* in this layer is a square node with a node function

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i \left(p_i x + q_i y + r_i \right) \tag{2.38}$$

where $\overline{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer will be referred to as *consequent parameters*.

- ***Layer 5:*** This single node in this layer is a circle node labeled Σ that computes the overall output as the summation of all incoming signals, i.e.,

$$O_i^5 = \text{overall output} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{2.39}$$

Actually, one of the ANFIS advantages is that it uses a hybrid learning procedure for estimation of the premise and consequent parameters. In this process by keeping fixed the premise parameters, it estimates them in a forward pass and then in a backward pass by keeping fixed the consequent parameters the process would be continued. In the first path, the input would be forward and propagated and then by applying the least squared method the error would be calculated where is the third layer. Also, in the second path, the error which happens during the first step would be backward to and the premise parameters are updated by a gradient descent method. The details of the hybrid learning procedure that is used in an ANFIS are given in Jang [57], [58].

2.3 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is a transformation that can be used to analyze the temporal and spectral properties of non-stationary signals. The DWT is defined by the following equation [66]:

$$W(j,k) = \sum_j \sum_k f(x) 2^{-j/2} \psi\left(2^{-j} x - k\right) \tag{2.40}$$

The set of functions $\psi_{j,k}(n)$ is referred to as the family of wavelets derived from $\psi(n)$, which is a time function with finite energy and fast decay called the mother wavelet. The basis of the wavelet space corresponds then, to the orthonormal functions obtained from the mother wavelet after scale and translation operations. The definition indicates the projection of the input signal into the wavelet space through the inner product, then, the function $f(x)$ can be represented in the form [66]:

$$f(x) = \sum_{j,k} d_j(k) \psi_{j,k} \tag{2.41}$$

where $d_j(k)$ are the wavelet coefficients at level j. The coefficients at different levels can be obtained through the projection of the signal into the wavelets family as expressed in equations (2.42) and (2.43).

$$\left\langle f, \psi_{j,k} \right\rangle = \sum_l d_l \left\langle f, \varphi_{j,k+1} \right\rangle \tag{2.42}$$

$$\left\langle f, \psi_{j,k} \right\rangle = \frac{1}{\sqrt{2}} \sum_l c_l \left\langle f, \varphi_{j-1,2k+1} \right\rangle \tag{2.43}$$

The DWT analysis can be performed using a fast, pyramidal algorithm described in terms of multi-rate filter banks. The DWT can be viewed as a filter bank with octave spacing between filters. Each sub-band contains half the samples of the neighboring higher frequency sub-band.

In the pyramidal algorithm the signal is analyzed at different frequency bands with different resolution by decomposing the signal into a coarse approximation and detailed information. The coarse approximation is then further decomposed using the same wavelet decomposition step. This is achieved by successive high-pass and low-pass filtering of the time signal and a down-sampling by two [67], as defined by the following equations:

$$a_j(k) = \sum_m h(m-2k)\, a_{j+1}(m) \tag{2.44}$$

$$d_j(k) = \sum_m g(m-2k)\, a_{j+1}(m) \tag{2.45}$$

Figure 2.11 shows a two-level filter bank. Signals $a_j(k)$ and $d_j(k)$ are known as approximation and detail coefficients, respectively. This process may be executed iteratively forming a wavelet decomposition tree up to any desired resolution level.

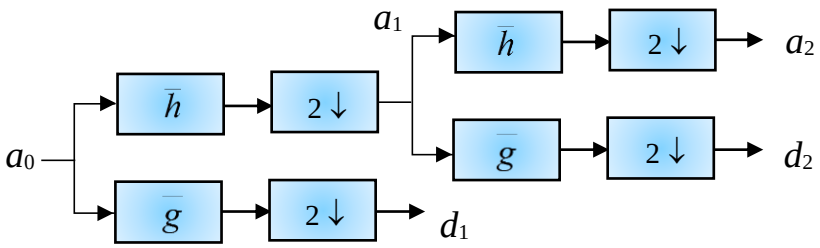


Figure 2.11. Two-level discrete wavelet filter bank scheme [67].

2.4 Summary

In this chapter an introduction about the main theoretical topics related to the system developed in this work was present. Kalman filter provides an efficient computational means to estimate the state of a process, in a way that minimizes the mean of the squared error. FISs form a consistent methodology to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning. NNs are capable of learning from examples and store this knowledge in network weights distributed throughout the net and ANFIS is a fuzzy inference system implemented in the framework of adaptive networks. The DWT is a transformation that can be used to analyze the temporal and spectral properties of non-stationary signals. The concepts presented here will be used in the next chapters to develop the PDR system.

Chapter 3

Attitude estimation using Neuro Fuzzy Adaptive Kalman Filter

The traditional Kalman filter formulation (from here referred to as TKF) assumes complete *a priori* knowledge of the process and measurement noise statistics, matrices *Q* and *R*. However, in most practical applications these matrices are initially estimated or, in fact, are unknown. The problem here is that the optimal performance of Kalman filtering strongly depends on stablishing correct initial process and measurement noise statistics. Inadequate initial statistics would be reflected in inaccuracies of estimated states, or it would introduce undesired biases to the estimates. In fact, in extreme cases, incorrect a priori information can cause practical divergence of the filter [69], [70].

The purpose of an adaptive Kalman filter formulation is to reduce the errors in the estimation by modifying or adapting the Kalman filter to the real data. Thus, adaptive filter formulation deals with the problem of having imperfect a priori information and provides an improvement in performance over the fixed filter approach [70].

This chapter presents the development of a Kalman Filter with Neuro-Fuzzy adaptation (KF-NFA) which is applied in attitude estimation, relying on information derived from triaxial accelerometer and gyroscope sensors contained in an inertial measurement unit (IMU). The adaptation process is performed on the filter statistical information matrix *R*, which is tuned using an Adaptive Neuro Fuzzy Inference System (ANFIS) based on the filter innovation sequence through a covariance-matching technique.

3.1 IMU data fusion for attitude estimation through Kalman filtering

Euler angles are defined as the rotation angles from the body frame to the navigation frame. Euler angles, roll ϕ, pitch θ and yaw ψ, can be derived from measurements provided by inertial sensors such as gyroscopes and accelerometers. In a conventional inertial navigation system, the attitude is calculated by integrating the angular rate obtained from the gyroscope signals. However, the gyroscope signals undergo an effect called *bias* or *drift*, which is the average output from the gyroscope when it is not undergoing any rotation [16].

Accordingly, the angular rate integration is not appropriate for calculating the attitude of pedestrians because the bias makes the attitude error to diverge over time. The bias effect on the integrated signal of a triaxial gyroscope can be seen in figure 3.1, which presents the results from an experiment where the gyroscope was rotated from 90° to –90° arbitrarily on each axis from an initial position of 0°.

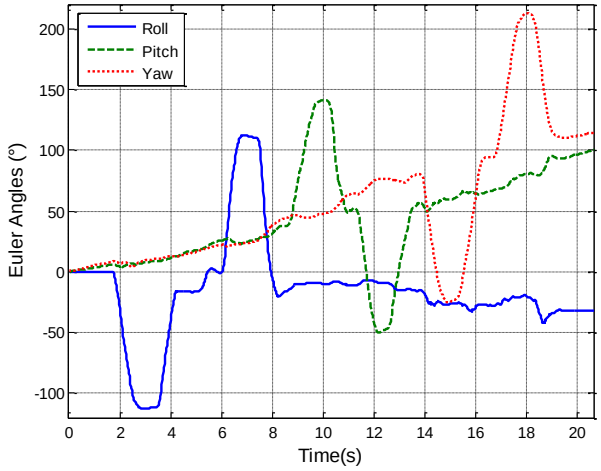


Figure 3.1. Bias effect over the Euler angles obtained by integration of the gyro signal.

The roll and pitch angles can be estimated using a triaxial accelerometer [a_x a_y a_z] through a geometric relation between the accelerations that act on each axis [71], according to the Eq. (3.1) and Eq. (3.2):

$$\phi = \tan^{-1}\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \tag{3.1}$$

$$\theta = \tan^{-1}\left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}\right) \tag{3.2}$$

The results of applying (3.1) and (3.2) are shown in figure 3.2 following the same procedure described previously. It is observed that these angles are very noisy and the bias effect is lower in comparison to the angles calculated with the integration of gyro signals. However, the principal disadvantage of this method is that the calculation of the angles could be affected by external accelerations.

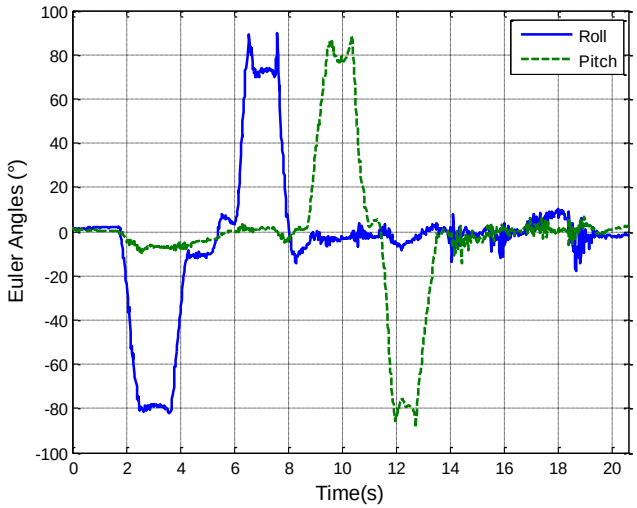


Figure 3.2. Euler angles, Roll and Pitch angles, calculated by a triaxial accelerometer.

In the case of the yaw angle, it can be calculated based on the roll and pitch values and the measurements from a triaxial gyroscope, as follows:

$$\dot{\psi} = -\omega_x \sin\theta + \omega_y \sin\phi \cos\theta + \omega_z \cos\phi \cos\theta \tag{3.3}$$

where ω_x, ω_y and ω_z are the angular rates along the three axis. Therefore, the yaw angle can be obtained by numerical integration of Eq. (3.3), and it is shown in figure 3.3.

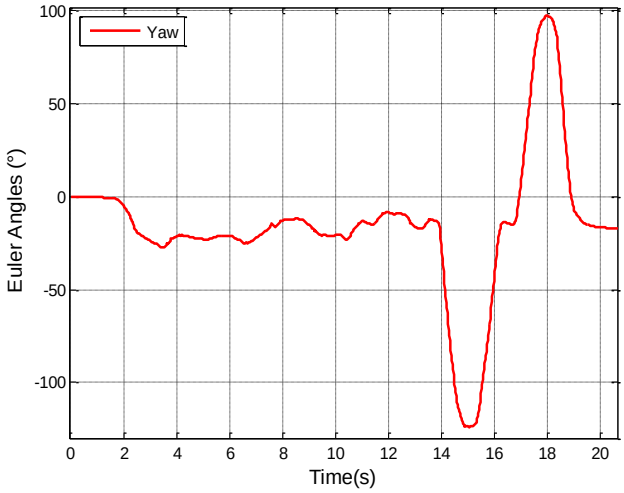


Figure 3.3. Yaw angle calculated by a triaxial gyroscope and the pitch and roll angle.

Kalman-based sensor fusion is a common approach to pursue attitude estimation in navigation systems. For that purpose, several tuning procedures based on the innovation sequence have been proposed [69], [70]. In this work, the fusion of gyroscope and accelerometer measurements is carried out using the Kalman filter algorithm with a Neuro-fuzzy adaptive tuning procedure based on the innovation sequence. Figure 3.4 shows a block diagram of the scheme used to perform attitude estimation based on the IMU information.

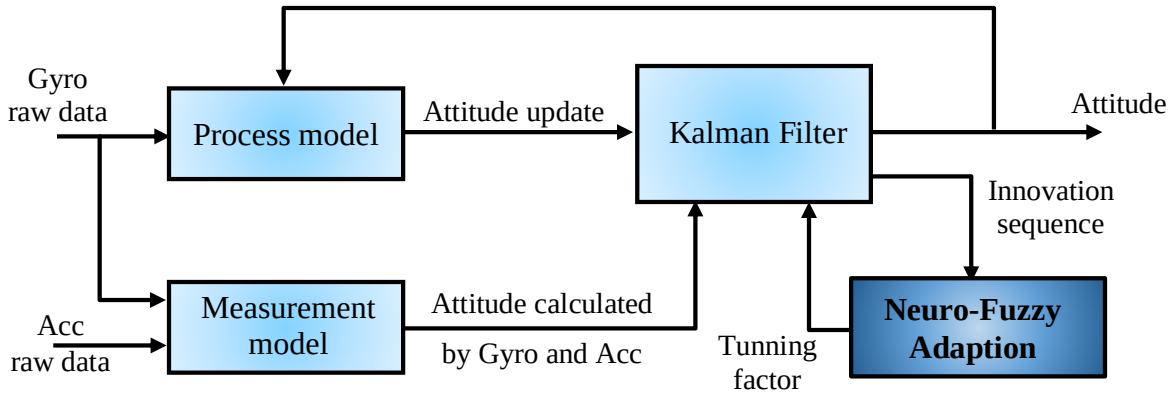


Figure 3.4. Kalman filter structure for attitude estimation.

3.1.1 The process and measurement model

The *process model* is represented by the equations (3.4) and (3.5), where the state vector is x_k, ϕ, θ, and ψ are the estimated roll, pitch, and yaw angles in degrees, respectively; b_ϕ, b_θ, and b_ψ are the bias which are estimated by the filter in degrees per second, dt is the sample rate and u_k is the input vector containing the gyroscope raw data.

$$x_{k+1} = A \cdot x_k + B \cdot u_k \tag{3.4}$$

$$\begin{bmatrix} \phi \\ b_\phi \\ \theta \\ b_\theta \\ \psi \\ b_\psi \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & -dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -dt & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -dt \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ b_\phi \\ \theta \\ b_\theta \\ \psi \\ b_\psi \end{bmatrix} + \begin{bmatrix} dt & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & dt & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_x \\ 0 \\ \omega_y \\ 0 \\ \omega_z \\ 0 \end{bmatrix}_k \tag{3.5}$$

During the process, the angles ϕ, θ, and ψ are updated integrating the gyroscope raw data and performing a correction in order to eliminate the estimated bias:

$$\phi_{k+1} = \phi_k + \left(\omega_{xk} - b_{\phi_k}\right) \cdot dt$$
$$b_{\phi_{k+1}} = b_{\phi_k} \tag{3.6}$$

$$\theta_{k+1} = \theta_k + \left(\omega_{y_k} - b_{\theta_k}\right) \cdot dt$$
$$b_{\theta_{k+1}} = b_{\theta_k} \tag{3.7}$$

$$\psi_{k+1} = \psi_k + \left(\omega_{z_k} - b_{\psi_k}\right) \cdot dt$$
$$b_{\psi_{k+1}} = b_{\psi_k} \tag{3.8}$$

The *measurement model* uses the angles obtained from the triaxial accelerometer and the gyroscope, providing an estimation of the z_k vector:

$$z_k = H_k x_k \tag{3.9}$$

where:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.10}$$

Noise arising from angular velocities measurements affects the process model as they are governing the general behavior of the process, whereas measurements obtained from inertial sensors are subject to measurement noise. These noise signals are assumed to be uncorrelated zero-mean Gaussian white noise sequences, with covariance matrices *Q* and *R*, representing process noise and measurement noise covariance matrices, respectively.

As described in the previous section, the Kalman filter formulation requires statistical information of matrices *Q* and *R*. In this work the values of *Q* and *R* matrices were acquired experimentally by maintaining the inertial module iNEMO® [72] to stand for one week in a vibration free environment. Tables 3.1 and 3.2 show the statistical moments obtained from the experiment. The process noise and measurement noise covariance matrices are then represented as expressed in equations (3.11) and (3.12):

$$Q = \text{diag}\left[\sigma^2_{\omega_x} \quad \sigma^2_{b_{\omega_x}} \quad \sigma^2_{\omega_y} \quad \sigma^2_{b_{\omega y}} \quad \sigma^2_{\omega_z} \quad \sigma^2_{b_{\omega_z}}\right] \tag{3.11}$$

$$R = \text{diag}\left[\sigma^2_{acc_x} \quad \sigma^2_{acc_y} \quad \sigma^2_{acc_z}\right] \tag{3.12}$$

Table 3.1. Gyroscope measurement statistical moments.

Parameter		**X-axis**	**Y-Axis**	**Z-axis**	**Units**
mean	(μ)	0.0130	-3.7636	4.7626	°/sec
var	(σ^2)	0.0450	0.2270	0.2218	°/sec

Table 3.2. Accelerometer measurement statistical moments.

Parameter		**X-axis**	**Y-Axis**	**Z-axis**	**Units**
mean	(μ)	15.4932	-6.5478	48.4429	mg
var	(σ^2)	3.6190	3.6278	5.1917	mg

Optimal performance of Kalman filtering strongly depends of correct initial process and measurement noise statistics. Inadequate initial statistics would be reflected in inaccuracies of estimated states, or it would introduce undesired biases to the estimates, generating, in extreme cases, filter divergence. Adaptive filter formulation deals with the problem of having imperfect a priori information and provides an improvement in performance over the fixed filter approach.

3.2 Attitude estimation using Neuro-Fuzzy Adaptive Kalman Filtering

There are two approaches to the adaptive Kalman filtering problem: innovation-based adaptive estimation (IAE) and multiple-model-based adaptive estimation (MMAE) [73]. In IAE the adaptation is carried out on the covariance matrices of measurement and/or process noise, based on the changes in the innovation or residual sequences. In MMAE a bank of Kalman filters runs in parallel with different models for satisfying filter's true statistical information. MMAE has been used in several applications such as positioning systems [74] and attitude determination systems in microsatellites and spacecrafts [75], [76], with good results, however a drawback is its processing time and computational complexity. The work presented in this paper is based on an IAE approach.

3.2.1 Innovation based adaptive estimation algorithm

The innovation based adaptive estimation (IAE) approach is based on the improvement of the filter performance through the adaptive estimation of the filter statistical information, the matrices *Q* and/or *R*. The innovation sequence or filter residual sequence is the difference between the actual measurement vector and its estimate [70]:

$$Inn_k = z_k - H_k x_k^- \tag{3.13}$$

The innovation sequence represents the additional information available to the filter as a result of a new measurement z_k. For this reason the innovation sequence represents the

information content in the new observation and is considered the most relevant source of information for the filter adaptation. The occurrence of bad data first shows up in the innovation vector. In this way the innovation sequence reports the discrepancy between predicted and actual measurement. For an optimal filter, the innovation sequence is a linear combination of independent Gaussian random variables. Therefore, the innovation is a white Gaussian sequence of mean zero and covariance:

$$S_k = H_k P_k^- H_k^T + R_k \tag{3.14}$$

by checking whether the innovation sequence indeed possess their theoretical statistical properties the performance of the Kalman filter can be assessed.

In this work, an IAE scheme based on a neuro-fuzzy system is used to carry out the adaptation process on the statistical information contained in matrix *R*. The general idea behind this technique is that the actual value of the covariance of the residuals matches with its theoretical value. When the statistical values of the innovation sequence show discrepancies between the theoretical and actual covariance values, then a Neuro-Fuzzy inference system (FIS) adjusts a tuning factor applied to the matrix *R* causing a reduction in the discrepancy.

3.2.2 Adaptive adjustment of the measurement noise covariance matrix *R* with *Q* known

The adaptation process is carried out through the adjustment of the measurement noise covariance matrix *R* with *Q* known. Figure 3.5 shows a block diagram which schematically describes the whole process. The covariance matrix *R* represents the accuracy of the measurement instrument. Assuming that the noise covariance matrix *Q* is known, an IAE approach employing the principles of fuzzy logic is used to adaptively adjust the matrix *R*.

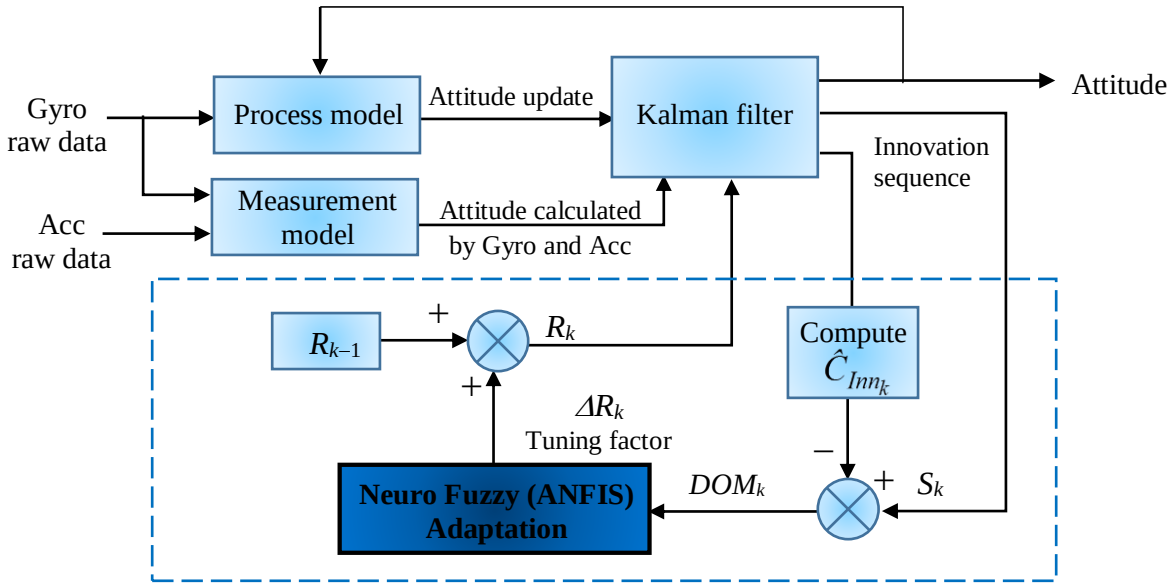


Figure 3.5. Kalman filter structure for attitude estimation.

This adaptation is carried out in three steps. First, the theoretical covariance of the innovation sequence is obtained from the Kalman filter algorithm by the equation (3.14). Second, the actual covariance $\hat{C}_{Inn_k}$ is defined as an approximation of the Inn_k sample covariance through averaging inside a moving estimation window of size M:

$$\hat{C}_{Inn_k} = \frac{1}{M}\sum_{i=i_0}^{k} Inn_i \; Inn_i^T \tag{3.15}$$

where $i_0 = k - M + 1$ is the first sample inside the estimation window. Third, if it is found that the actual value of the covariance has a discrepancy with its theoretical value S_k, then an adaptation algorithm derives adjustments based on the size of this discrepancy. The covariance matching technique is employed by the adaptation algorithm. Hence, a new variable called Degree of Matching (*DoM*) is defined to indicate the degree of discrepancy between S_k and $\hat{C}_{Inn_k}$; this is expressed as:

$$DoM_k = S_k - \hat{C}_{Inn_k} \tag{3.16}$$

From equation (3.14), note that an increment in R will increment S, and vice versa. Then the basic idea of the adaptation is to adjust R_k to vary S_k in accordance with the value of DoM_k, in order to reduce the discrepancy between S_k and $\hat{C}_{Inn_k}$.

The adaptation of the (i,i) element of R_k is performed in accordance with the (i,i) element of DOM_k using the following general rules:

1. IF S and $\hat{C}_{Inn_k}$ match almost perfectly ($DoM \cong 0$) THEN keep R unchanged.
2. IF S is greater than its actual value $\hat{C}_{Inn_k}$ ($DoM > 0$) THEN decrease R.
3. IF S is smaller than its actual value $\hat{C}_{Inn_k}$ $DoM < 0$) THEN increase R.

therefore, R is adjusted as follows:

$$R_k(i,i) = R_{k-1}(i,i) + \Delta R_k \tag{3.17}$$

An Adaptive Neuro Fuzzy Inference System (ANFIS) is used to generate the tuning factors ΔR_k for the diagonal elements of R_k, as represented in the block diagram shown in figure 3.5. ANFIS combines the learning capabilities of neural networks with the approximate reasoning of fuzzy inference systems.

The advantage of ANFIS over fuzzy systems is the use of a hybrid learning algorithm to identify the parameters of Sugeno-type fuzzy inference systems. The optimization method used for training the FIS membership function parameters is a combination of the least-squares estimation algorithm with the backpropagation gradient descent algorithm [57], [58].

In this work, the ANFIS architecture is the foundation to build the previous collection of IF-THEN rules with the appropriate membership functions to model a set of given input-output data. The input-output pairs are comprised by DoM_k (i,i) as the input linguistic variable and ΔR_k as the output variable.

3.3 The Neuro-Fuzzy Adaptive Kalman filtering test

The proposed system was implemented in MATLAB® using the ANFIS model included in the fuzzy logic toolbox. Figure 3.6 presents the obtained transition functions corresponding to input DOM_k related to the output ΔR_k, for the case of the roll (ϕ), pitch (θ), and yaw (ψ) angles. Note how the ANFIS output follows the transition function. Figure 3.7 shows the Gaussian membership functions for the input DOM_k that resulted from the training of the ANFIS. The number of membership functions was established empirically.

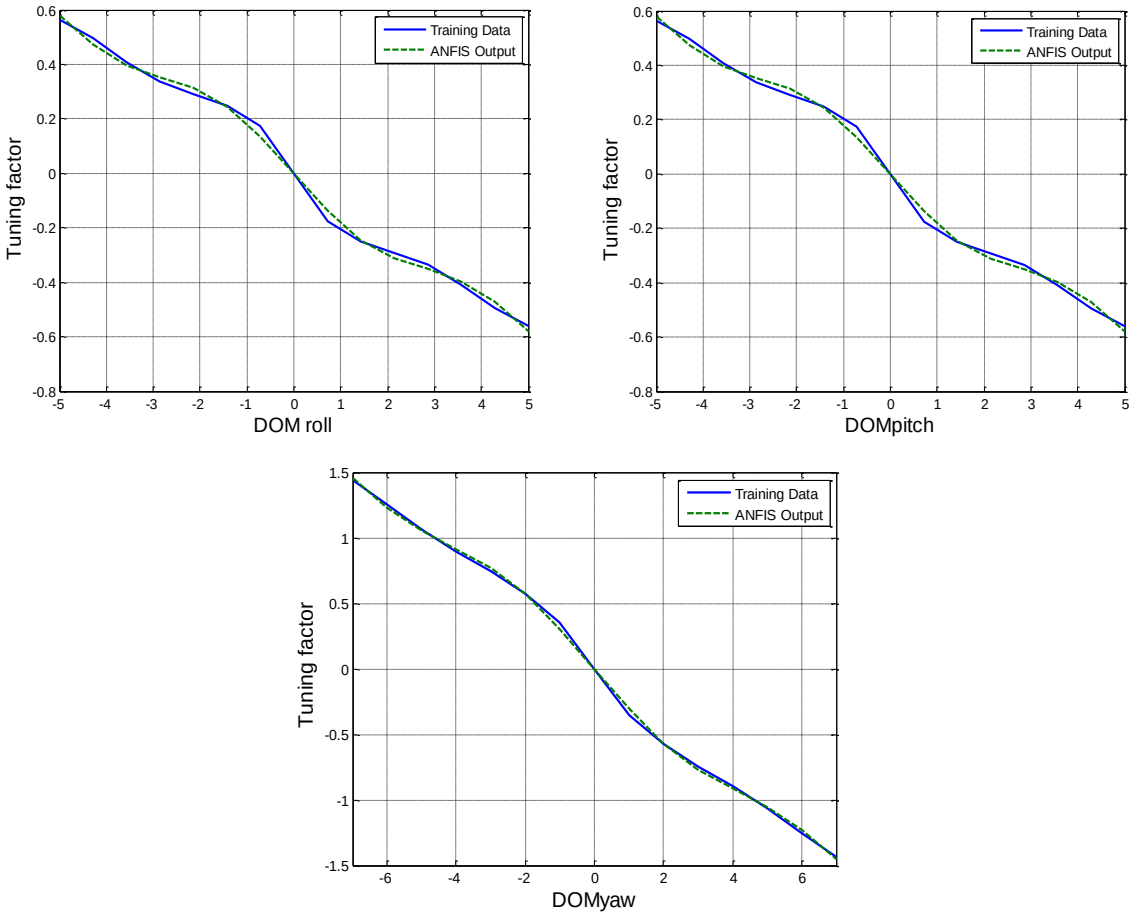


Figure 3.6. Outputs and the transition functions.

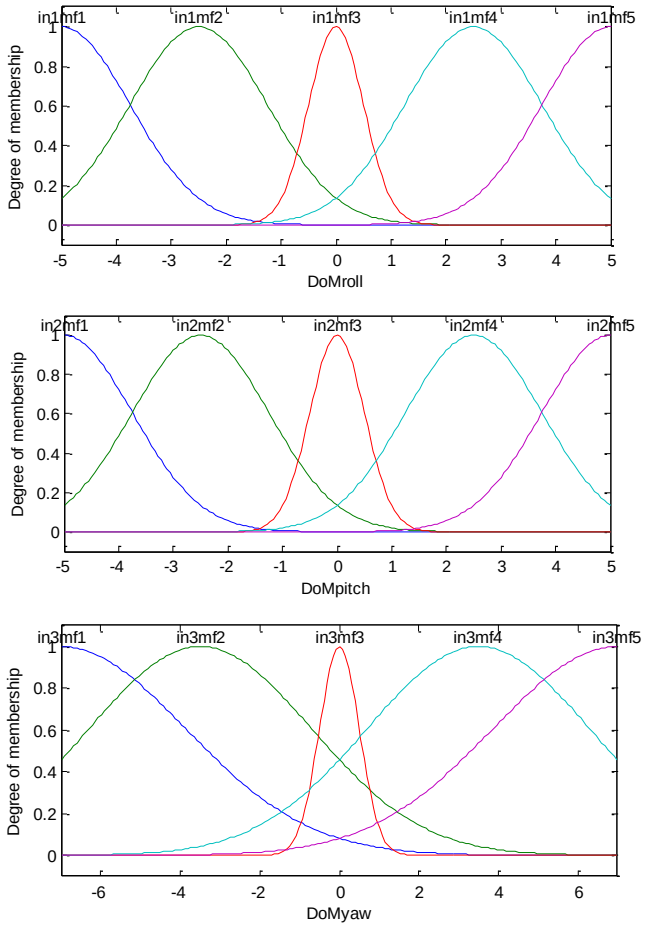
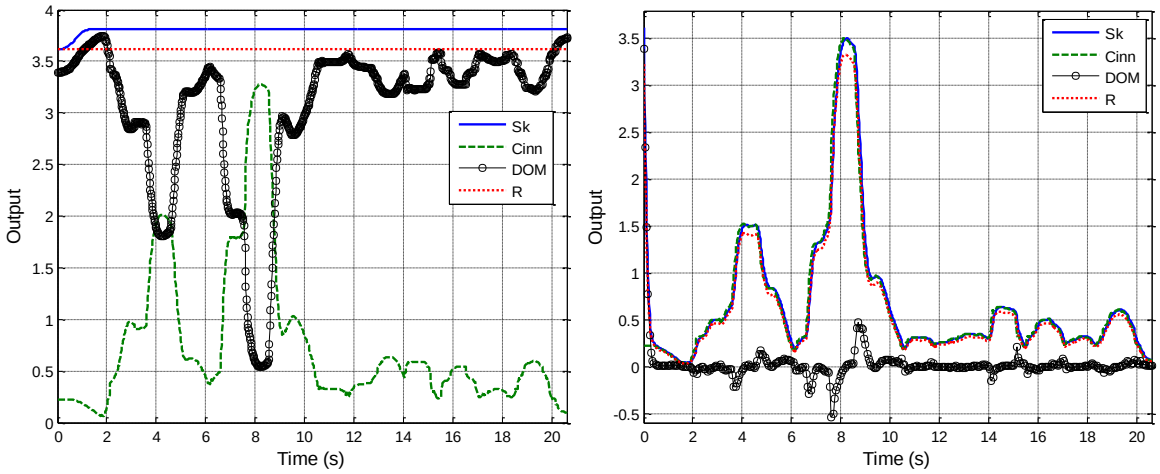


Figure 3.7. Gaussian membership functions for the input DOM_k.

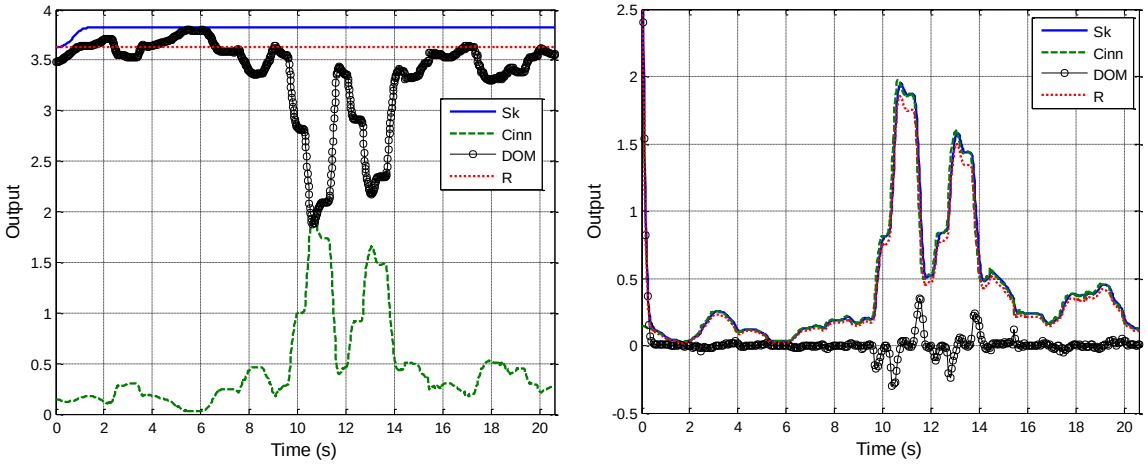
Figure 3.8(*a*) presents a comparison between the actual innovation covariance $\hat{C}_{Inn_k}$ and its theoretical value S_k, obtained when both, Q and R, are maintained fixed, i.e. Kalman filter results without adaptation for the roll angle. A discrepancy between the actual innovation covariance $\hat{C}_{Inn_k}$ and its theoretical value S_k can be noticed. Figure 3.8(*b*) presents the comparison between the actual innovation covariance $\hat{C}_{Inn_k}$ and its theoretical value S_k, resulted when R is adjusted using the proposed neuro-fuzzy system. It can be noticed that S_k and $\hat{C}_{Inn_k}$ remain almost equal, DoM is around zero and R oscillates around its true value. This situation arises due to the continuous adjustment of the values in the main

diagonal of *R*, produced by the equation 3.17. Figures 3.9 and 3.10 present the pitch and yaw adaptation results, respectively.



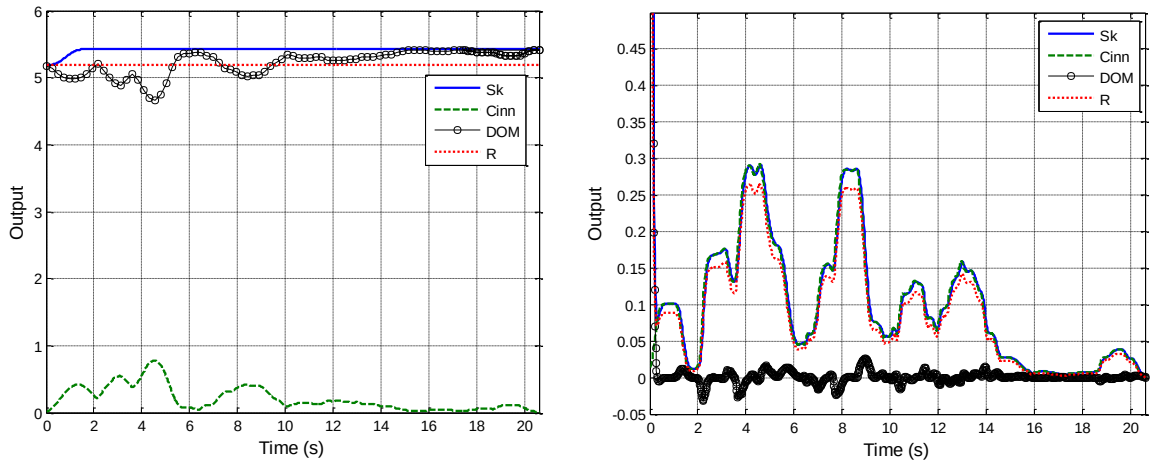
(*a*) (*b*)

Figure 3.8. (*a*) Kalman filter without adaptation at roll angle. (*b*) Kalman Filter with Neuro-Fuzzy adaptation at roll angle.



(*a*) (*b*)

Figure 3.9. (*a*) Kalman filter without adaptation at pitch angle. (*b*) Kalman Filter with Neuro-Fuzzy adaptation at pitch angle.



(*a*) (*b*)

Figure 3.10. (*a*) Kalman filter without adaptation at pitch angle. (*b*) Kalman Filter with Neuro-Fuzzy adaptation at pitch angle.

3.4 Summary

In this chapter a Kalman filter with Neuro-Fuzzy adaptation (KF-NFA) for attitude estimation has been presented. The adaptation is realized in the sense of dynamically adjusting the measurement noise covariance matrix R of a Kalman filter. This method uses the covariance-matching technique to determinate if adjustments to R are needed.

An experiment was developed to demonstrate the efficiency of the KF-NFA for attitude estimation. The description of this experiment and the obtained results will be discussed in Chapter 5.

Chapter 4

Step length estimation and activity detection in a PDR system

As mentioned in the previous sections, in Pedestrian Dead Reckoning (PDR) techniques, the estimation of each new position is based on the previous one derived from the last step, taking advantage of the sequential nature of pedestrian motion. Furthermore, displacements are calculated based on the estimation of a step length. The step length is a time-varying process which is strongly correlated to the velocity and the step frequency of the pedestrian [17]. For that reason some works have incorporated pedestrian activity classification in order to improve the accuracy of the calculation of the traveled distance, and consequently the localization accuracy [15], [17], [29], [36].

In this chapter, a fuzzy model aiming to perform step length estimation and activity detection in the context of a pedestrian dead reckoning system using inertial sensors, is presented. The fuzzy model uses a single inertial measurement unit with a triaxial accelerometer and gyroscope, which have adequate characteristics to be used in human activity analysis [21]. Effective algorithms are required to interpret the accelerometer data in the context of different activities. In this work a wavelet analysis is applied to triaxial accelerometer data in order to identify points in the signal where a pedestrian changes from one activity to another.

In search of incorporating the classification algorithm into a PDR system this work aggregates a dynamical method for estimating the step length using a Fuzzy Inference System (FIS), which uses the pedestrian activity as an additional input. The system uses a single device attached to the pedestrian waist rather than multiple devices distributed across the body or on the user's foot. In that way, the developed techniques can be easily incorporated into personal smartphones with IMU sensors, which nowadays are becoming of popular use. With that purpose, this work considers activities that a pedestrian performs in his daily life, such as: walking, walking fast, jogging, and running. Additionally, with the information provided by an altimeter, the system can detect when the pedestrian is going up or down stairs.

4.1 Fuzzy model for step length estimation

The task of determining the activity a pedestrian is executing is inherently a classification problem. Some researchers, particularly in the field of biomechanics, have determined that gait trajectory signals have nonlinear and non-stationary characteristics [77]. In nonlinear or complex classification problems, neural networks, which have gained prominence in the area of pattern recognition, have several properties that make them attractive.

Figure 4.1 presents the general block diagram of the proposed algorithm. Step length is estimated using a fuzzy inference system, with information obtained from positive-going and negative-going zero crossing direction of normalized signals obtained from the three-axis accelerometer. The system incorporates information corresponding to the subject activity, which is simultaneously obtained from the IMU signals. A feature vector constructed with statistical information obtained from wavelet coefficients, is fed into a multilayer perceptron neural network, which provides information about the subject activity.

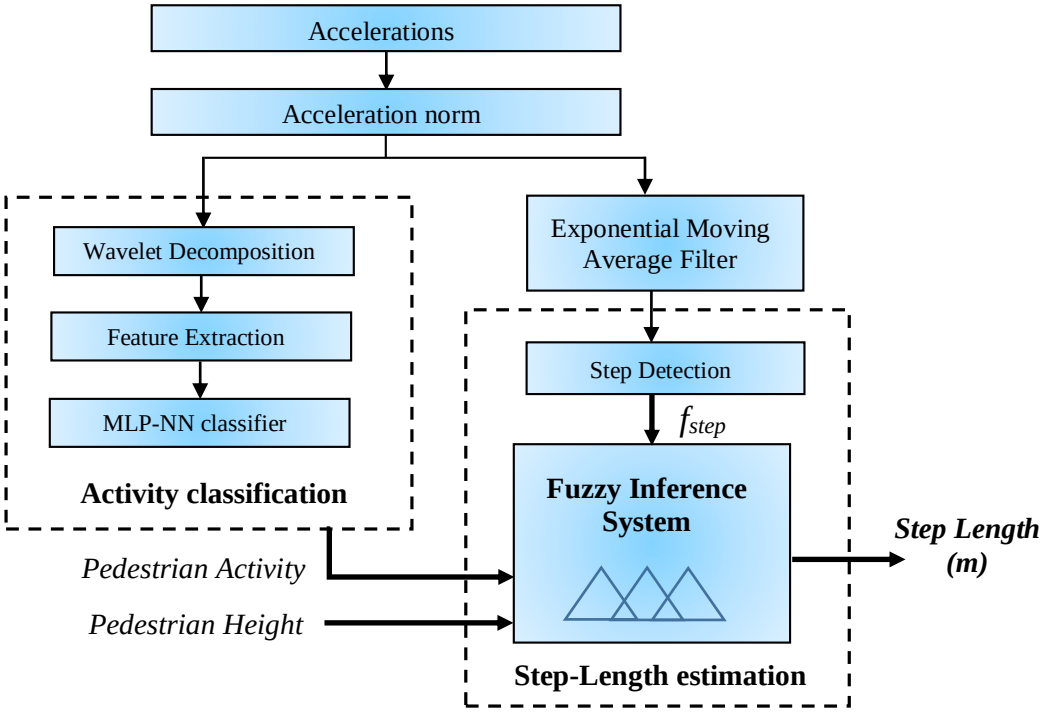


Figure 4.1. Block diagram of the proposed architecture for activity classification and step-length estimation.

4.2 Pedestrian activity classification algorithm

Classification of pedestrian activity is achieved by using the measurements from the three axial accelerometer clusters of the STEVAL-MKI062V2 [72]. The STEVAL-MKI062V2 is an inertial measurements unit (IMU) which includes accelerometers, gyroscopes and magnetometers, as well as pressure and temperature sensors to provide 3-axis sensing of linear, angular and magnetic motion, complemented with temperature and barometer/altitude readings. Therefore, this unit constitutes a platform with 10 degrees of freedom (DOF). The STEVAL-MKI062V2 includes the LSM303DLH, which is a system-in-package featuring a 3D digital linear acceleration sensor. The LSM303DLH has a full-scale linear acceleration of ±2 g / ±4 g / ±8 g, which can be selected by the user as needed. Figure 4.2 shows the wearable sensor module and the way it is attached to the test subject.

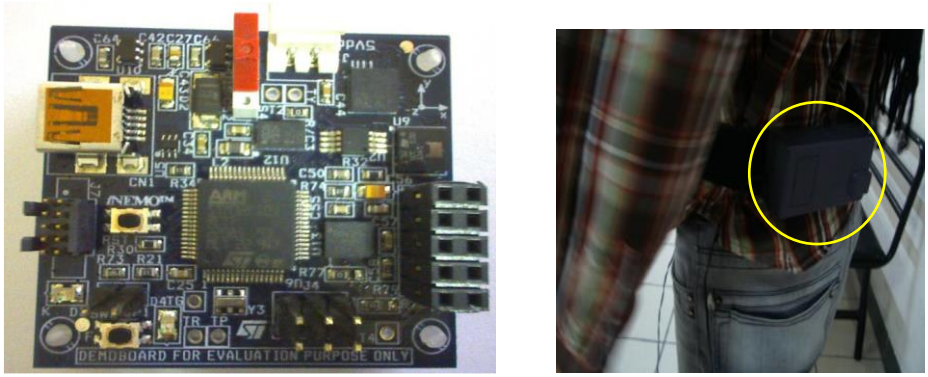


Figure 4.2. Wearable sensor module and the way it is attached to the test subject.

Four types of activity patterns were collected from 26 subjects. The height of these subjects is located in a range between 1.50 and 1.78 m. The four activity patterns types were specifically walking, walking fast, jogging and running. The data were collected from the triaxial accelerometer every 20 ms, which corresponds to 50 samples per second. A typical example of two seconds raw acceleration signal is shown in figure 4.3. Features were extracted from the norm of the acceleration raw signals corresponding to the three axis *Ax*, *Ay*, *Az*, at the time *k*. The norm vector is obtained as:

$$A_k = \sqrt{A_{x_k}^2 + A_{y_k}^2 + A_{z_k}^2} \tag{4.1}$$

The mean of the norm signal is obtained through averaging over a sliding window of *N* samples, with *N*=100 in this experiment.

$$\overline{A}_k = \frac{1}{N} \sum_{k-N-1}^{k} A_k \tag{4.2}$$

which in this case is equivalent to two seconds of data at a sampling frequency of 50 Hz. Subtracting (4.2) from the norm acceleration signal results in:

$$Acc_k = A_k - \overline{A}_k \tag{4.3}$$

where Acc_k is the normalized acceleration signal at time *k*, with the effect of gravity removed.

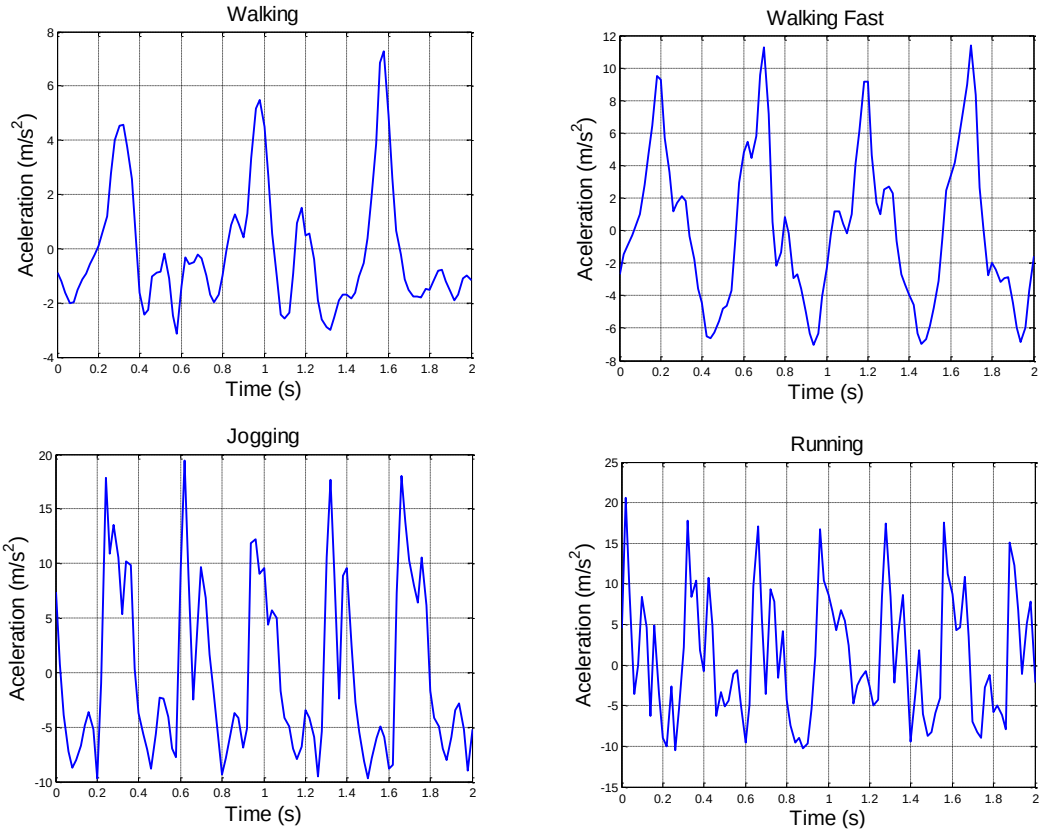


Figure 4.3. Raw and filtered normalized acceleration signal acquired during walking.

In this work, a feature extraction process is implemented using wavelet decomposition [66]. The original signal is decomposed into a series of approximation and detail coefficients containing spectral and temporal information. Statistical information of these coefficients is used to form the feature vector for classification purposes in a dynamical way [78]. Figure 4.4 shows a three-level filter bank. Sets *cA* and *cD* are known as approximation and detail coefficients, respectively.

The wavelet decomposition is obtained through a sliding window of 100 samples, using the wavelet toolbox included in MATLAB®, with a three-level wavelet decomposition applied to the norm of the acceleration signal and a Daubechies 5/7 mother wavelet. Figure 4.4 shows in shaded boxes the target frequency band signals selected for feature extraction. Two statistical features, mean and standard deviation, were extracted

from each coefficient, giving a total of eight attributes in order to discriminate the four activities; walking, walking fast, jogging, and running.

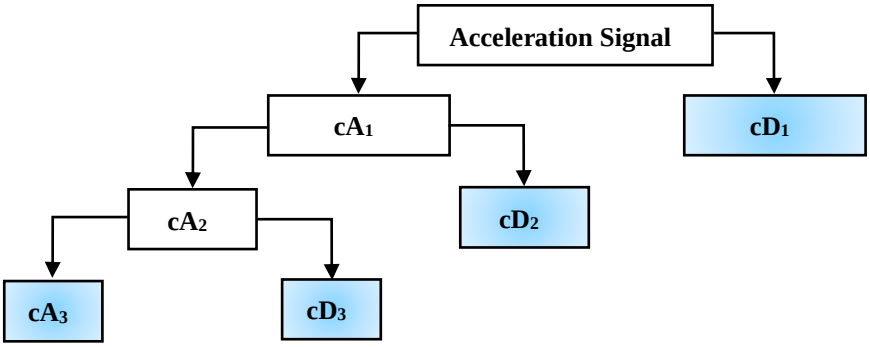


Figure 4.4. Wavelet decomposition.

4.3 Classification using a multi-layer perceptron neural network

Activity classification was performed using a feed forward multilayer perceptron neural network. The perceptron used in this work was trained using the Levenberg-Marquardt back propagation algorithm [79]. The backpropagation algorithm used in the training of multilayer perceptrons, is formulated as a non-linear least-squares problem. Essentially, the Levenberg-Marquardt algorithm is a least-squares estimation method based on the maximum neighborhood idea. Let $E(w)$ be an objective error function made up of m individual error terms $e_i^2(w)$ as follows:

$$E(w) = \sum_{i=1}^{m} e_i^2(w) = \left\| f(w) \right\|^2 \tag{4.4}$$

where

$$e_i^2(w) = (y_{di} - y_i)^2 \tag{4.5}$$

y_{di} is the desired value of output neuron i, and y_i is the actual output of that neuron.

The aim of the Levenberg-Marquardt algorithm is to compute the weight vector w such as $E(w)$ is minimized. In each iteration the weight vector is updated according to equation (4.6):

$$w_{k+1} = w_k + \delta w_k \tag{4.6}$$

where

$$\delta w_k = -\left(J_k^T f(w_k)\right)\left(J_k^T J_k + \lambda I\right)^{-1} \tag{4.7}$$

J_k is the Jacobian of *f* evaluated at w_k, λ is the Marquardt parameter, and *I* is the identity matrix.

The number of epochs in the training phase differs from one example to another, however, the Levenberg-Marquardt back propagation algorithm provided a fast convergence. Structure of the neural network consisted of two hidden layers with eight neurons and one output layer with four neurons corresponding to the four classification conditions. The neural networks toolbox included in MATLAB® was used in this experiment. Two-fold cross validation was employed in the validation stage. The procedure is explained in detail in the chapter 5.

4.4 Step length estimation algorithm

Pedestrian dead reckoning (PDR) is a navigation technique based on position estimation of a person travelling on foot. A PDR algorithm is expected to provide information about a pedestrian position during a natural walk. A PDR algorithm can be divided into three principal parts: step detection, step length estimation and heading determination. In this work, a proposal for step detection and step length estimation to be used in a PDR algorithm is presented. Next subsections describe in detail the proposed procedures.

4.4.1 Step detection

The step detection stage is fundamental in a PDR algorithm; if the step detection is inaccurate, then it will not be possible to determine the distance traveled by the subject, and consequently an estimation of the current position is not feasible [80]. There are two basic methods to detect pedestrian steps based on the measurements provided by the accelerometers: peak detection [81] and zero-crossing detections [18]. These methods consist of comparing the acceleration values with the predefined thresholds and taking the minimum step period into account.

In waist attached devices, peak detection is not appropriate because peaks of acceleration also occur in irregular motions, such as turning when avoiding obstacles on a crowded road. It is difficult to distinguish between these peaks and those measured during regular walking, thus step misdetections are prone to occur. Furthermore, methods based on comparing the acceleration values with the predefined thresholds are not appropriate when the pedestrian activity is not constant because the time and amplitude characteristics of the acceleration measures change during the walk.

So, in order to make the step count more reliable and to improve the accuracy of the distance traveled calculation we propose an algorithm to identify valid steps. Detection is performed by the positive-going and negative-going zero crossing direction of the normalized signal obtained from the three-axis accelerometer according to equation (4.3). In order to improve the performance of the step detection, an initial smoothing of the acceleration signal is performed. Raw normalized acceleration signals are passed through an exponential weighted moving average filter defined in equation (4.8), as follows:

$$\hat{A}cc_k = \alpha\, Acc_k + (1-\alpha)\, \hat{A}cc_{k-1} \tag{4.8}$$

where $\hat{A}cc$ indicates the estimated filtered acceleration signal, *Acc* is the raw acceleration signal, $0 < \alpha < 1$ is a filter parameter, and *k* is the discrete time variable. In this application

$\alpha = 0.3$ and the initial condition is assumed as the initial measured value of the acceleration signal. An example of the raw and filtered signals is shown in figure 4.5. Then, positive-going and negative-going zero crossing direction of the signal $\hat{A}cc$ are detected in order to determine the interval where a step has occurred. Once the algorithm determines the zero crossing positions it proceeds to detect the maximum peak position which is present between two consecutive zero crossings. An example of this procedure is shown in figure 4.6. Squares indicate the maximum peaks and circles indicate the zero crossings detected.

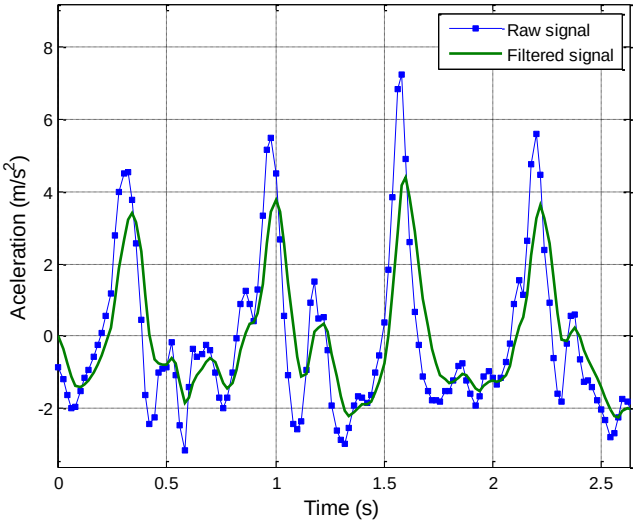


Figure 4.5. Normalized raw and filtered signals obtained during walking.

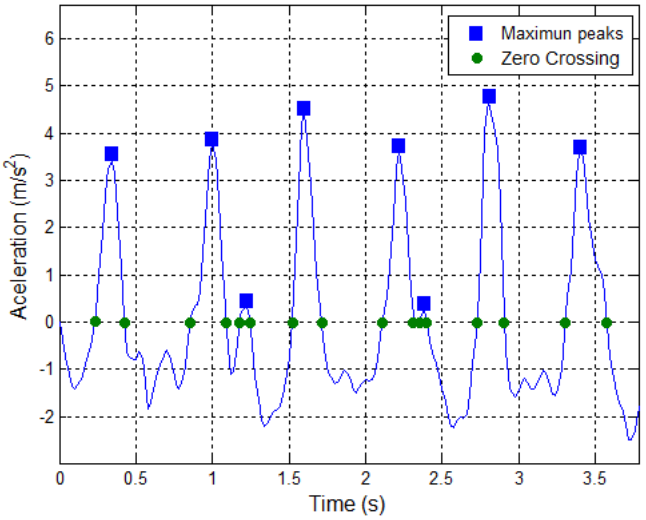


Figure 4.6. Zero Crossing and Maximum peaks detection.

The next procedure is to determine which of those peaks represent valid steps. A valid step occurs when the foot touches the ground and this is recorded by the accelerometer as a peak of large magnitude with short duration. As shown in figure 4.6, the valid steps correspond with the largest peaks. It can be noticed the presence of peaks with smaller magnitudes which do not correspond to valid steps.

In some reported works, valid steps are detected by comparing the acceleration signals with some predefined threshold [12], [18]. The proposed algorithm identifies the valid steps by calculating the area under each peak, using Riemann sum:

$$S = \sum_{i=1}^{n} f\left(x_i^*\right)\left(x_i - x_{i-1}\right), \qquad x_{i-1} \le x_i^* \le x_i \tag{4.9}$$

in this case, $\left(x_i - x_{i-1}\right)$ is the sample time 0.02 seconds. The procedure starts measuring the width of each detected peak in the time domain, which corresponds to the range comprised between two consecutive zero crossing with alternate directions. The length of the measured width is stored in a variable called *range* (R_{ng}) as shown in figure 4.7. Then, the equation 4.9 is computed using all the acceleration values contained inside R_{ng}. These values correspond to x_i^*.

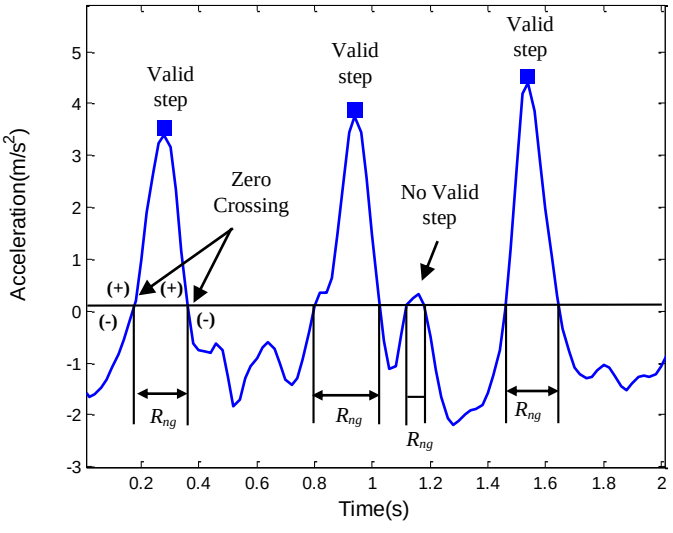


Figure 4.7. Scheme of peaks discrimination.

The algorithm compares the value of S against a sliding window average (S_{mean}). This comparison is performed by computing the ratio between S and S_{mean}. If the ratio is greater than some value previously specified the event is considered a valid step and its position is stored. Once the pedestrian's steps are detected, the step frequency is calculated based on the step period, which is obtained by measuring the time between consecutive valid steps.

4.4.2 Step length estimation

The step length assessment is fundamental to the calculation of distance traveled. In this work the step length is dynamically estimated using a Fuzzy Inference System (FIS). In the proposed FIS, three inputs linguistic variables are defined: *step frequency*, with partition in three fuzzy sets labeled as slow, regular and fast; *pedestrian activity*, defined by four fuzzy sets with labels walking, walking fast, jogging, and running; and *pedestrian height*, which is defined by three membership functions with labels short, medium, and high. The membership functions of the input linguistic variables are shown in figure 4.8.

The output variable is defined as *step length*, the linguistic values are defined by eight Gaussian fuzzy sets: small (SM), normal (NL), little large (LL), medium large (ML), large (LG), very large (VL), giant (GT) and very giant (VG). These eight fuzzy sets are shown in figure 4.9. Therefore, 36 rules complete the rule base defined in the FIS system. An example of these rules is:

IF (step-freq is *slow*) AND (activity is *walk*) AND (height is *short*)

THEN (step-length is *small*)

the fuzzy operators min, max-product and centroid method are used as the fuzzy operators of the intersection compositional rule of inference and for the defuzzification, respectively.

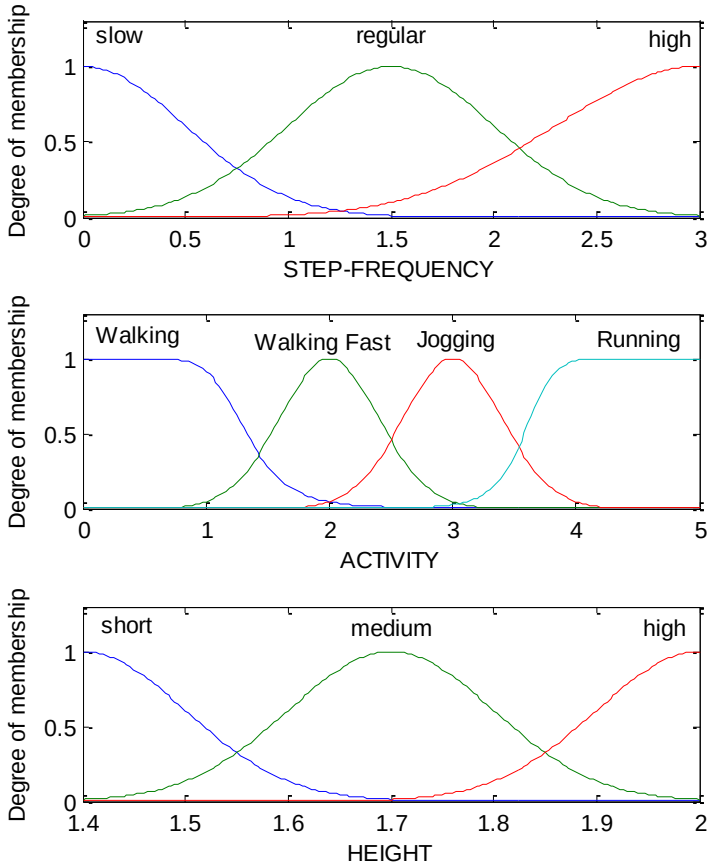


Figure 4.8. Membership functions of the input linguistic variables.

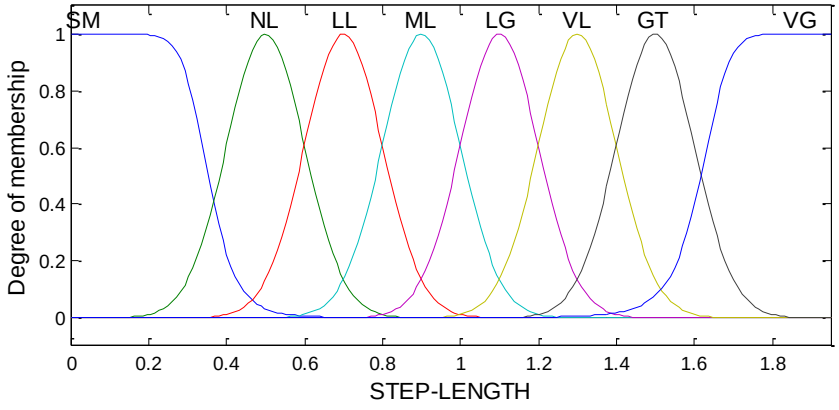


Figure 4.9. Membership functions of the output linguistic variable.

4.5 Summary

In this chapter an approach on Pedestrian Dead Reckoning (PDR) which incorporates activity classification over a Fuzzy Inference System (FIS) for step length estimation has been presented. In the proposed algorithm, the pedestrian is equipped with an Inertial Measurement Unit (IMU) attached to the waist, which provides three-axis accelerometer and gyroscope signals. The main goal is to integrate the activity classification and step-length estimation algorithms into a PDR system.

In order to improve the step-length estimation, several types of activities are classified using a Multi-Layer Perceptron (MLP) neural network with feature extraction based on statistical parameters from wavelet decomposition. This work focuses on classifying activities that a pedestrian performs routinely in his daily life, such as walking, walking fast, jogging and running. The step-length is dynamically estimated using a multiple-input–single-output (MISO) fuzzy inference system. The experimental tests and results will be described in the Chapter 5.

Chapter 5

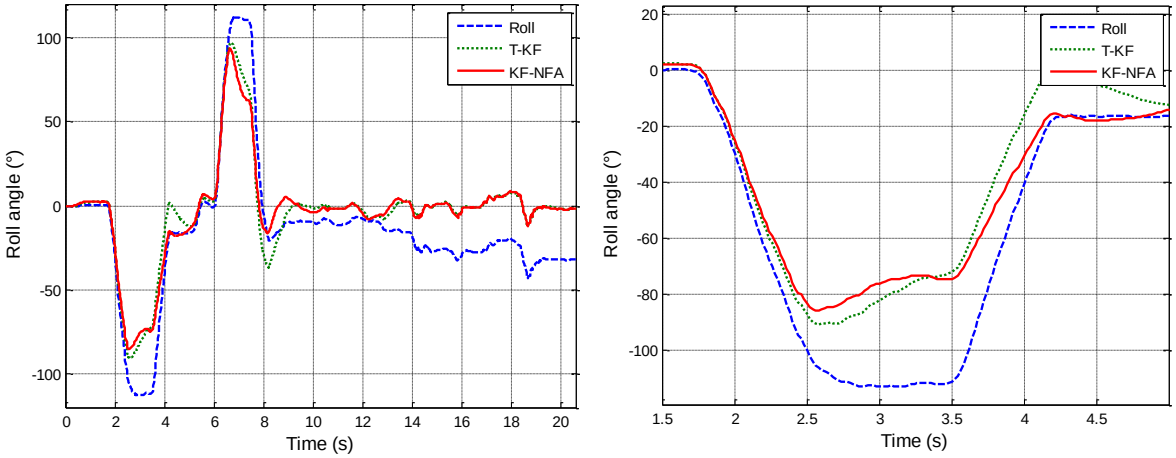
Experimental results

In order to validate the entire PDR system, in this chapter the description and results of the different experiments carried out are presented.

5.1 KF-NFA attitude estimation results

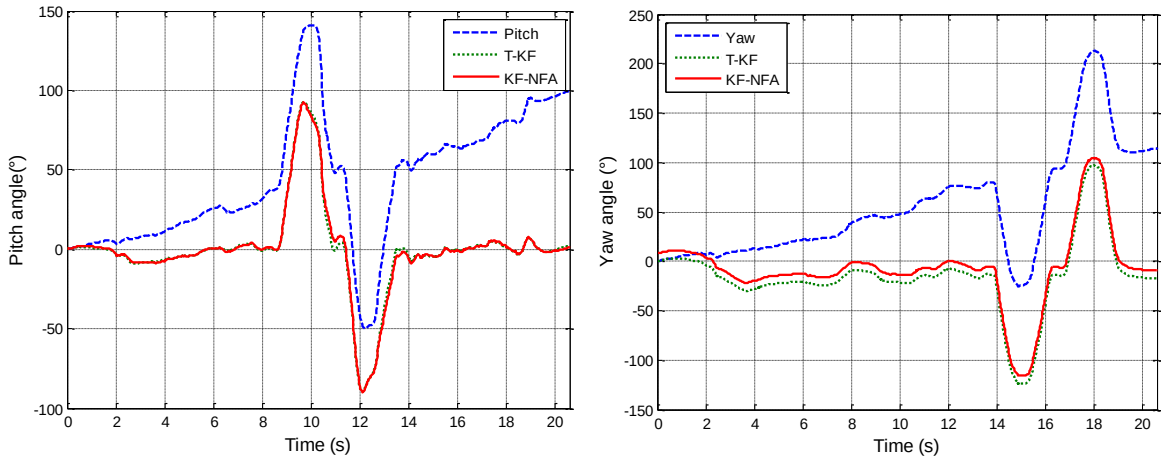
In order to demonstrate the efficiency and validity of the Kalman filter with Neuro-Fuzzy adaptation (KF-NFA) for attitude estimation, an experiment where the gyroscope was rotated from 90° to -90° on each axis from an initial position of 0° was performed. For comparison purposes, figures 5.1 and 5.2 present the attitude obtained using KF-NFA, the traditional Kalman filter (T-KF), and the attitude obtained from the gyroscope integration. Figure 5.1(*b*) shows a zoom of the roll angle in the interval where the sensor recorded an approximate rotation of -90°. The KF-NFA output is closer to -90°; hence a filter performance improvement is obtained.

In a previous work [82], a fuzzy scheme for the adaptation of the Kalman filter (KF-FLA) was reported. The drawback of a fuzzy logic approach is the definition of the fuzzy sets used in the fuzzy rules. A neuro-fuzzy adaptation scheme offers the advantage of adjusting, in an automatic way, the fuzzy sets used in the fuzzy rules. A Neuro-Fuzzy Inference System (ANFIS) is used to adaptively adjust the measurement noise statistical information for the Kalman filter based on the filter innovation sequence, which performs sensor fusion of gyroscope and accelerometer measurement signals.



(*a*) (*b*)

Figure 5.1. (*a*) Roll angle results. (*b*) –90° Roll angle.



(*a*) (*b*)

Figure 5.2. Euler angles results. (a) Pitch angle. (b) Yaw angle.

A comparison on percent mean square error (MSE) measured in each case using zero level as reference indicates an improvement when ANFIS-based adaptation (KF-NFA) is used, as indicated in Table 5.1.

Table 5.1. % MSE measured in the different adaptations

	Roll	**Pitch**	**Yaw**
Integration	19.01%	35.67%	53.37%
T-KF	8.9977%	7.9327%	29.9536%
KF-FLA	8.3975%	7.9145%	16.5305%
KF-NFA	8.3880%	7.7953%	16.5296%

5.2 Activity detection results

Several tests were carried out with 26 subjects wearing the IMU STEVAL-MKI062V2 attached to the waist. The height of these subjects is located in a range between 1.50 and 1.78 m. The subjects were asked to walk in a straight line a distance of 100 meters in each activity pattern twice. Data processing for activity classification and step-length estimation was carried out off-line using the described algorithm.

This experiment corresponds to the neural network-based activity classification. The available data is divided into two groups in order to perform two-fold cross validation. The first set is used for training and the second set is used for generalization; later on, both sets are interchanged and the process is repeated. Average classification rate is presented in the form of confusion matrices. Table 5.2 presents the confusion matrix obtained from the classification rate of the four activity patterns during the training process. Table 5.3 shows similar results corresponding to the generalization process.

Table 5.2. Classification accuracy (%) confussion matrix for the four walking patterns obtained by applying the training process

	Walking	Walking fast	Jogging	Running
Walking	92.3	7.7	0	0
Walking Fast	7.7	92.3	0	0
Jogging	0	0	88.45	11.55
Running	0	0	7.7	92.3
Overall Accuracy		**91.33%**		

Table 5.3. Classification accuracy (%) confussion matrix for the four walking patterns obtained by applying the testing process

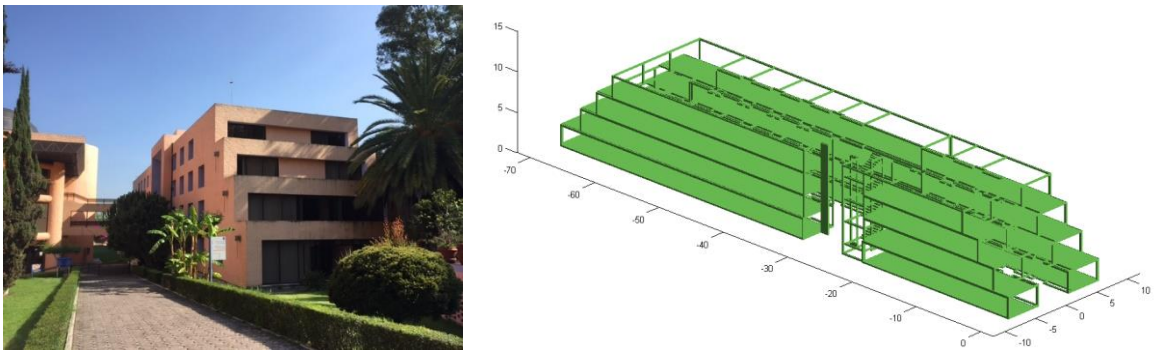
	Walking	Walking fast	Jogging	Running
Walking	84.61	15.39	0	0
Walking Fast	3.85	88.45	7.7	0
Jogging	0	7.7	84.6	7.7
Running	0	0	7.7	92.3
Overall Accuracy		**87.49%**		

The activity detection procedure was then incorporated in the step length estimation procedure as represented in the block diagram of figure 4.1. Subsequently, step length estimation and attitude estimation were used to develop a PDR system. The PDR system has been developed to provide information about location and movements of a person travelling in a GPS-denied environment. The next section provides the description of the different experiments performed to validate the PDR system.

5.3 PDR system results

In order to validate the PDR system, several experiments were performed in combined indoor/outdoor trajectories inside the building number one at the INAOE. In order to represent the trajectories of the true path, a 3-D computer assisted design (CAD) model of the building was built using CATIA V5, as shown in figure 5.3. The signal acquisition was carried out based on the inertial module iNEMO®, described in chapter 4, and in order to include height measurements, the DEMOSTBMPL3115A2 [83] was incorporated.

The DEMOSTBMPL3115A2 is a low-cost development kit for the Freescale MPL3115A2 pressure sensor. As shown in figure 5.4, this kit is bundled with all three boards that include the USB communication board, the interface board and the pressure sensor evaluation board. The MPL3115A2 employs a MEMS pressure sensor with I^2C interface to provide accurate pressure/altitude (50 to 110 kPa) and temperature (-40°C to 85°C) data.



(*a*) (*b*)

Figure 5.3. (*a*) Building one at the INAOE. (*b*) CAD model of the building.

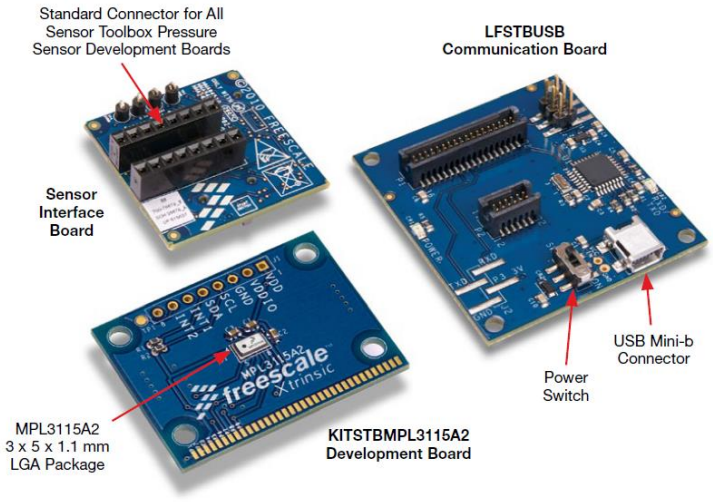


Figure 5.4. DEMOSTBMPL3115A2 [83].

As it was mentioned in the previous chapter, the sensors modules were attached to the waist of the test subject. Data were collected in real-time while the user completed the trajectories proposed in the different experiments. Each experiment was performed 5 times by 14 subjects. The height of these subjects is located in a range between 1.55 and 1.92 m. The data processing was performed off-line using the proposed PDR system implemented in MATLAB®.

5.3.1 Experiment 1

The first experiment is a combined indoor/outdoor trajectory. The trajectory is a closed path followed by a pedestrian who walks a straight line inside the building in order to get out and go around the building, returning to the starting point. The starting point is an access door to the building, which was established as the point [0, 0, 0]. The total distance of this trajectory is 299 m. From figure 5.5 to figure 5.7 the pedestrian position obtained with the proposed PDR system is presented. These figures show several views of the building, with a superimposed trajectory obtained through the PDR system, based on data collected during the experiments. Tables 5.4 and 5.5 present the quantitative results obtained from the step detection stage and the total distance derived from the step length estimation.

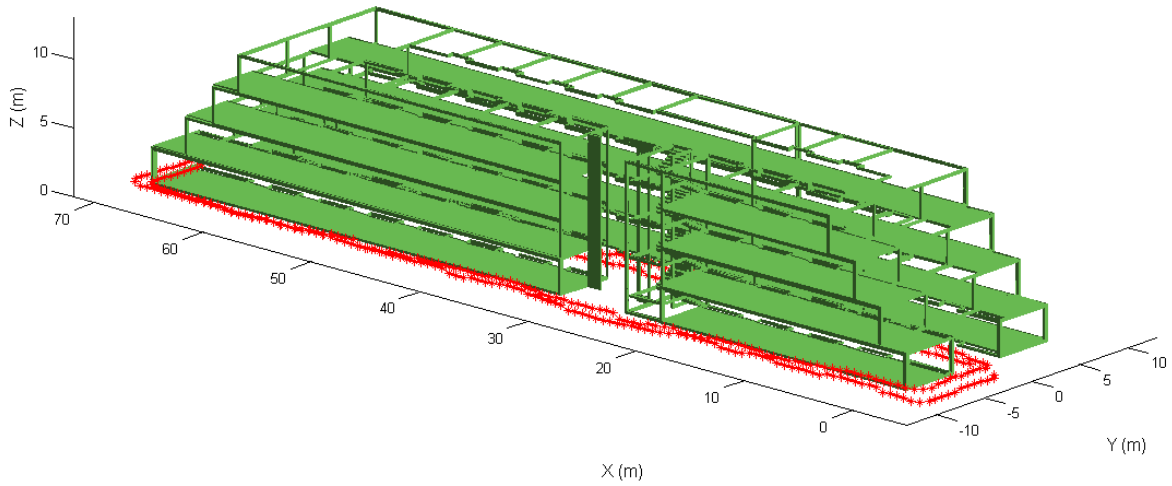


Figure 5.5. 3-D side view from experiment 1.

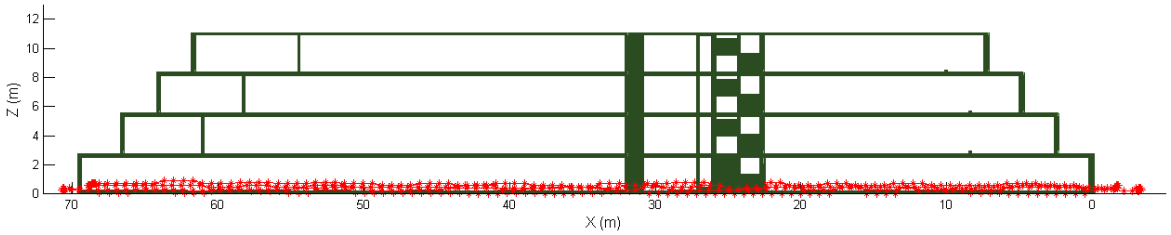


Figure 5.6. Lateral view from experiment 1.

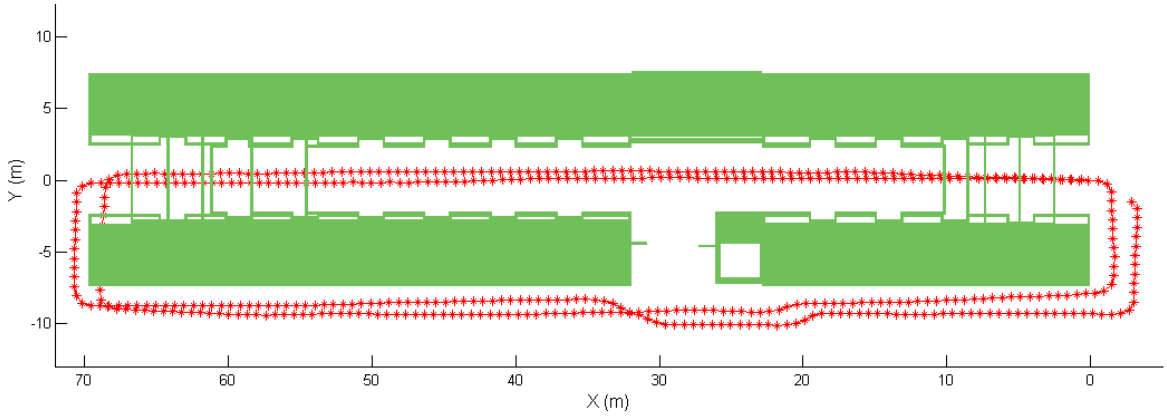


Figure 5.7. Top view from experiment 1.

Table 5.4. Step detection results from experiment 1.

Subject/ Height (m)		Number of detected steps										Error (%)
		test 1		test 2		test 3		test 4		test 5		
		Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	
S1	1.55	491	490	491	490	490	489	490	489	491	490	0.204%
S2	1.63	461	460	461	460	462	461	461	460	461	460	0.217%
S3	1.64	435	434	435	434	435	434	434	433	434	433	0.230%
S4	1.65	388	387	388	387	389	388	387	386	387	386	0.258%
S5	1.66	421	420	422	421	421	420	420	419	420	419	0.238%
S6	1.68	389	388	390	389	389	388	389	388	388	387	0.257%
S7	1.71	418	417	418	417	417	416	417	416	418	417	0.239%
S8	1.73	358	357	358	357	357	356	357	356	359	358	0.279%
S9	1.74	414	413	414	413	414	413	413	412	413	412	0.242%
S10	1.75	380	379	380	379	379	378	379	378	381	380	0.263%
S11	1.80	380	379	381	380	379	378	379	378	380	379	0.263%
S12	1.82	334	333	333	332	335	334	333	332	334	333	0.300%
S13	1.86	296	295	295	294	296	295	295	294	295	294	0.339%
S14	1.92	295	294	294	293	294	293	296	295	296	295	0.339%
										% Overall error		0.262%
										% Overall accuracy		99.738%

Table 5.5. Travelled distance estimation results for experiment 1 (299 m).

Subject	Step length average (m)	Estimated distance (m)					Error (%)
		test 1	test 2	test 3	test 4	test 5	
S1	0.581	300.045	300.1	298.8	300.53	299.62	1.37%
S2	0.657	301.728	301.638	301.738	302.59	302.718	0.59%
S3	0.674	291.734	299.304	298.084	299.51	300.434	1.54%
S4	0.669	258.38	299.51	299.06	299.49	299.00	13.23%
S5	0.696	291.523	299.18	300.97	300.93	299.38	1.01%
S6	0.721	279.076	299.31	298.01	297.52	297.09	6.37%
S7	0.739	307.256	301.84	301.2	301.20	301.55	4.35%
S8	0.75	267.101	300.57	298.81	298.82	299.2	10.20%
S9	0.733	302.108	299.72	299.16	300.40	299.66	0.35%
S10	0.751	283.759	299.83	300.45	298.04	299.74	4.41%
S11	0.754	277.313	298.43	297.53	298.1	297.87	6.97%
S12	0.774	257.021	298.66	298.41	300.39	300.36	13.43%
S13	0.898	263.936	299.61	300.04	299.53	301.93	11.65%
S14	839	312.976	299.89	300.42	301.27	300.01	4.86%
						% Overall error	5.737%
						% Overall accuracy	94.263%

5.3.2 Experiment 2

5.3.2.1 Test 1

This trajectory corresponds to an indoor situation which includes level changes. For the first test, the trajectory starts again in the access door to the building, the user walks a straight line inside the building until she/he arrives to the stairs, walk up the stairs to the next level, turn right and walks straight up towards the reading room "Guillermo Haro" and stops exactly in the office 1204, located at the first level of the building. The total distance of this trajectory is 48 m. From figure 5.8 to figure 5.11 the pedestrian position obtained with the proposed PDR system is shown. These figures show several views of the building, with a superimposed trajectory obtained through the PDR system, based on data collected during the experiments. Tables 5.6 and 5.7 present the quantitative results obtained from step detection stage and the total distance derived from the step length estimation.

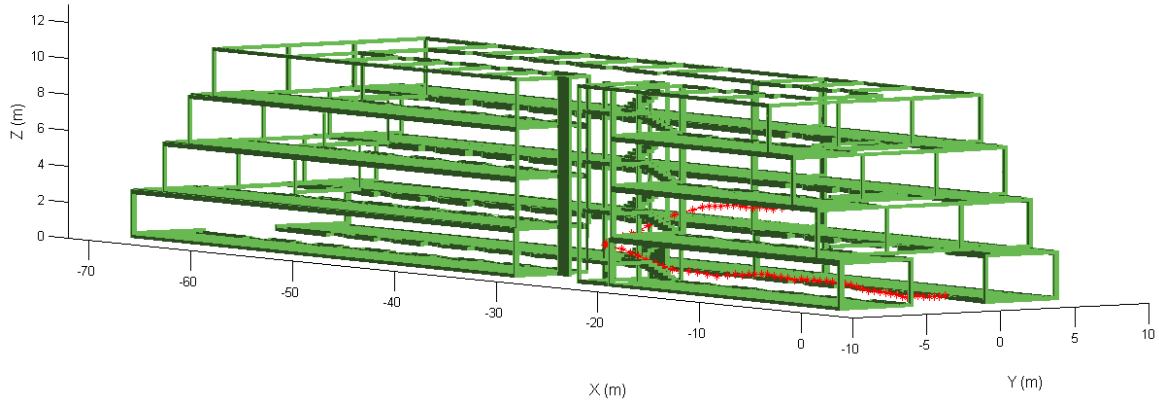


Figure 5.8. 3-D side view from experiment 2, test 1.

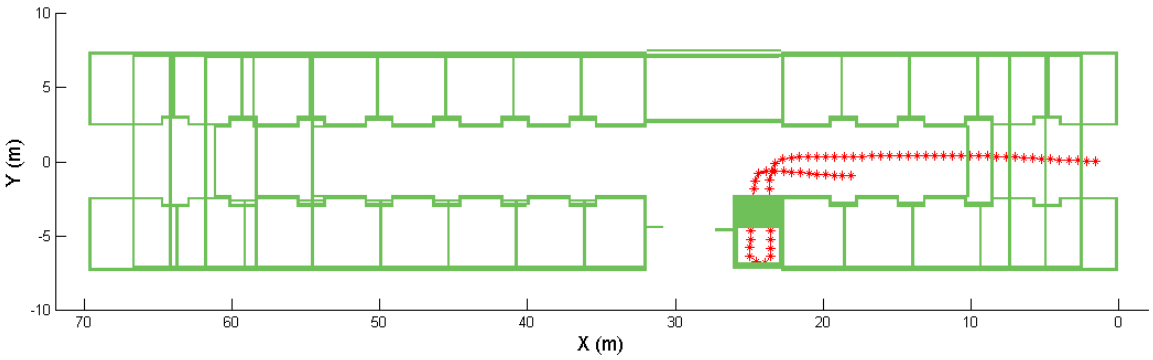


Figure 5.9. Top view from experiment 2, test 1.

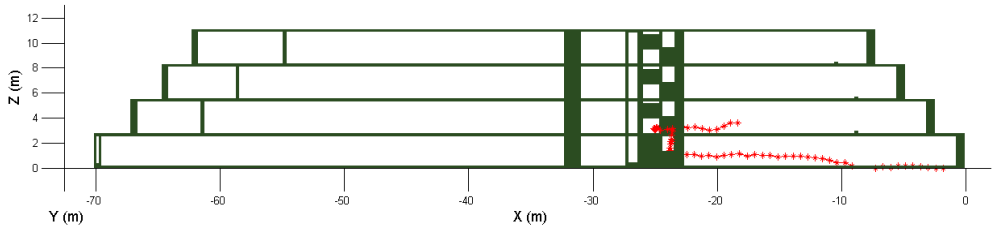


Figure 5.10. Lateral view from experiment 2, test 1.

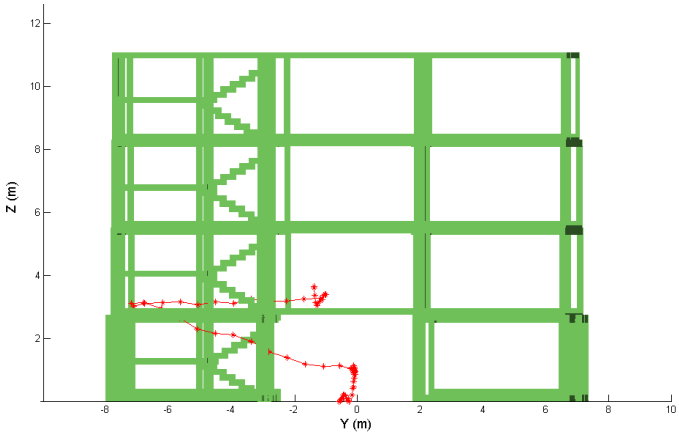


Figure 5.11. Frontal view from experiment 2, test 1.

Table 5.6. Step detection results from experiment 2, test 1.

Subject/ Height (m)		Number of detected steps										Error (%)
		test 1		test 2		test 3		test 4		test 5		
		Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	
S1	1.55	79	78	78	77	79	78	80	79	78	77	1.269%
S2	1.63	77	76	76	75	77	76	78	77	78	77	1.295%
S3	1.63	70	69	71	70	69	68	71	70	69	68	1.429%
S4	1.65	61	60	60	59	61	60	60	59	61	60	1.650%
S5	1.66	69	68	69	68	70	69	69	68	70	69	1.441%
S6	1.68	73	72	74	73	73	72	73	72	74	73	1.362%
S7	1.71	66	65	66	65	67	66	65	64	67	66	1.511%
S8	1.73	58	57	57	56	59	58	59	58	58	57	1.718%
S9	1.74	64	63	65	64	63	62	64	63	64	63	1.563%
S10	1.75	67	66	66	65	67	66	68	67	68	67	1.488%
S11	1.8	64	63	65	64	64	63	65	64	65	64	1.548%
S12	1.82	57	56	56	55	57	56	57	56	58	57	1.755%
S13	1.86	51	50	50	49	50	49	51	50	52	51	1.969%
S14	1.92	50	49	49	48	49	48	50	49	51	50	2.008%
										% Overall error		1.572%
										% Overall accuracy		98.428%

Table 5.7. Travelled distance estimation results for test 1 of the experiment 2 (48 m).

Subject	Step length average (m)	Estimated distance (m)					Error (%)
		test 1	test 2	test 3	test 4	test 5	
S1	0.575	46.147	48.43	48.88	49.82	48.97	4.68%
S2	0.67	50.257	49.992	49.70	50.642	50.33	2.45%
S3	0.676	45.957	46.48	47.28	46.84	47.00	7.73%
S4	0.695	40.978	49.47	49.58	48.62	47.35	8.34%
S5	0.696	46.631	45.67	46.49	46.18	45.42	0.85%
S6	0.729	50.319	50.32	50.61	50.21	50.44	3.80%
S7	0.757	48.461	48.82	48.06	47.08	47.68	0.21%
S8	0.739	41.36	43.45	42.99	44.29	42.09	1.91%
S9	0.761	47.195	48	48.23	46.01	46.63	2.27%
S10	0.767	49.836	48.64	50.51	49.47	48.84	4.69%
S11	0.875	54.25	50.75	49.57	48.8	48.61	14.24%
S12	0.875	48.125	47.81	47.51	49.35	50.08	0.77%
S13	0.99	48.531	49.92	50.01	50.48	48.91	11.03%
S14	1.047	53.388	53.07	54.54	55.34	55.27	2.98%
						% Overall error	4.710%
						% Overall accuracy	95.290%

5.3.2.2 Test 2

The second test uses the same path of test 1, but in the reverse direction. In this case the path starts from the office 1204 in front of the reading room “Guillermo Haro” at the first level, and ends at the access door of the building. The total distance of this trajectory is 48 m. From figure 5.12 to figure 5.15 a view of the building and the pedestrian position obtained with the proposed PDR system are shown. Tables 5.8 and 5.9 present the quantitative results.

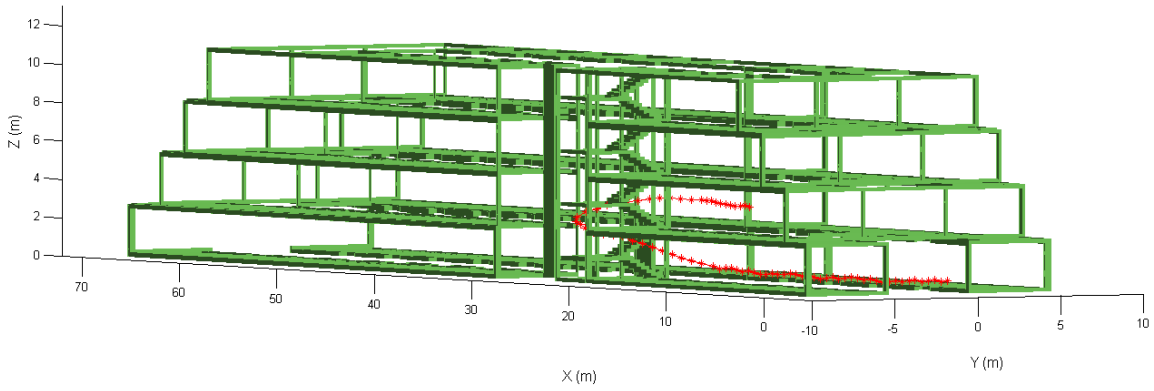


Figure 5.12. Side 3-D view from experiment 2, test 2.

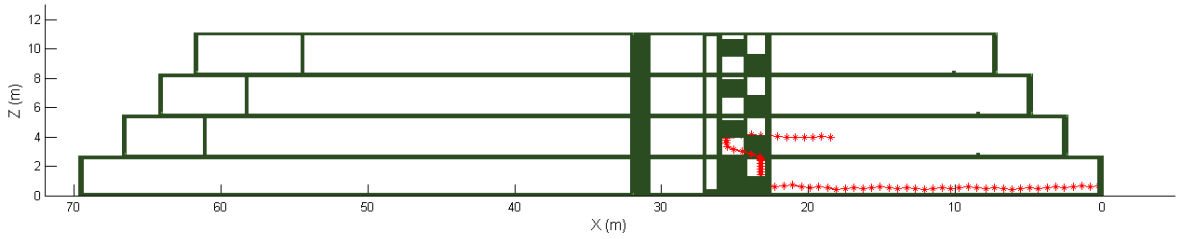


Figure 5.13. Lateral view from experiment 2, test 2.

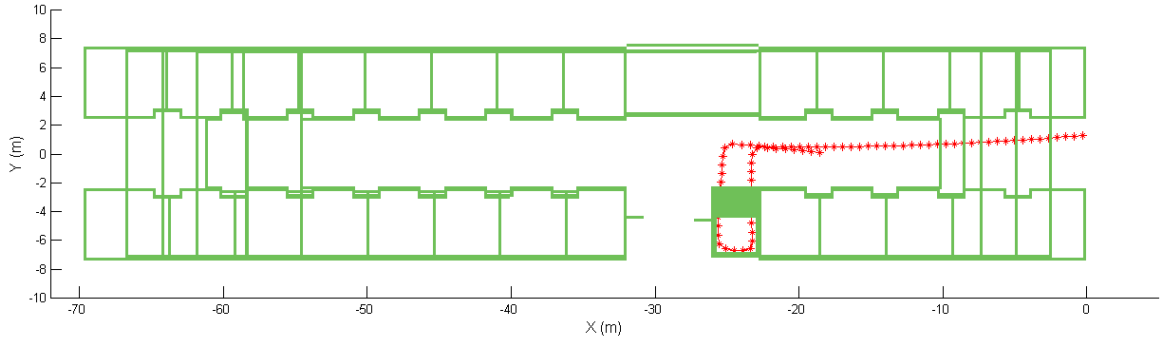


Figure 5.14. Top view from experiment 2, test 2.

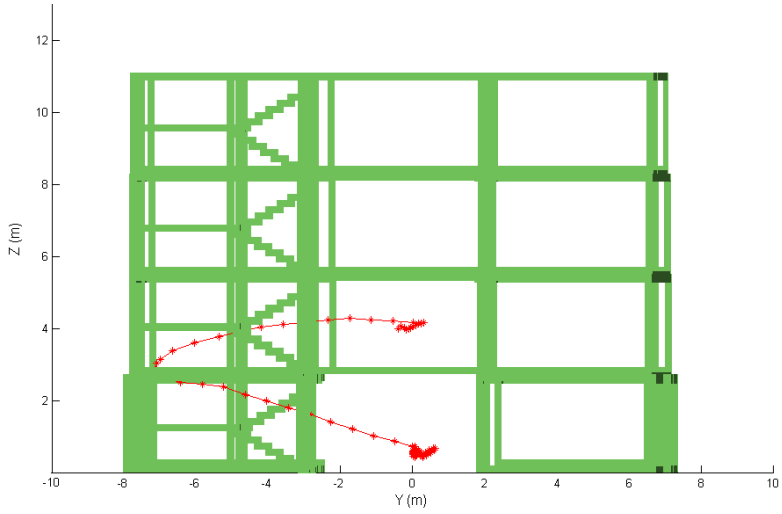


Figure 5.15. Frontal view from experiment 2, test 2.

Table 5.8. Step detection results from experiment 2, test 2.

Subject/ Height (m)		Number of detected steps										Error (%)
		test 1		test 2		test 3		test 4		test 5		
		Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	
S1	1.55	78	77	76	75	77	76	78	77	77	76	1.295%
S2	1.63	79	78	78	77	81	80	79	78	80	79	1.260%
S3	1.64	68	67	66	65	71	70	69	68	67	66	1.467%
S4	1.65	61	60	63	62	62	61	62	61	61	60	1.618%
S5	1.66	69	68	69	68	69	68	68	67	70	69	1.449%
S6	1.68	73	72	72	71	74	73	72	71	74	73	1.370%
S7	1.71	66	65	67	66	64	63	67	66	66	65	1.516%
S8	1.73	58	57	60	59	57	56	57	56	58	57	1.725%
S9	1.74	66	65	64	63	68	67	66	65	65	64	1.520%
S10	1.75	67	66	68	67	70	69	67	66	67	66	1.475%
S11	1.80	66	65	65	64	64	63	66	65	67	66	1.525%
S12	1.82	56	55	54	53	56	55	56	55	57	56	1.793%
S13	1.86	53	52	53	52	52	51	53	52	52	51	1.901%
S14	1.92	51	50	50	49	50	49	52	51	51	50	1.969%
										% Overall error		1.563%
										% Overall accuracy		98.437%

Table 5.9. Travelled distance estimation results for test 2 of the experiment 2 (48 m).

Subject	Step length average (m)	Estimated distance (m)					Error (%)
		test 1	test 2	test 3	test 4	test 5	
S1	0.575	48.43	47.71	48.93	47.3	47.83	0.42%
S2	0.67	49.992	49.772	52.042	51.172	50.292	2.20%
S3	0.676	46.48	45.74	45.41	47.68	47.32	5.72%
S4	0.695	49.47	48.27	49.95	49.59	48.37	11.77%
S5	0.696	45.67	45.56	47.87	45.92	46.63	3.59%
S6	0.729	50.32	49.85	50.01	50.53	49.4	0.22%
S7	0.757	48.82	47.85	49.33	48.98	48.96	8.21%
S8	0.739	43.45	44.86	44.42	44.25	43.45	12.63%
S9	0.761	48.00	47.87	46.46	48.87	47.79	8.49%
S10	0.767	48.64	50.36	49.18	48.65	48.77	1.22%
S11	0.875	50.75	49.91	49.82	48.81	50.21	9.18%
S12	0.875	47.81	50.05	50.3	49.4	49.63	9.67%
S13	0.99	49.92	49.66	48.43	49.97	48.75	8.72%
S14	1.047	53.07	55.58	53.39	54.48	54.36	1.63%
						% Overall error	5.976%
						% Overall accuracy	94.024%

5.3.3 Experiment 3

5.3.3.1 Test 1

For the first test in this experiment, the trajectory starts in the first floor, exactly in front of the office 1204 located in front of the reading room “Guillermo Haro”. The user walks in a straight line until he/she arrives to the stairs, then he/she goes downstairs to the ground floor, turns left and walks in a straight line towards the access door and stops. The total distance of this trajectory is 66 m. From figure 5.16 to figure 5.19 and tables 5.10 and 5.11 present the results.

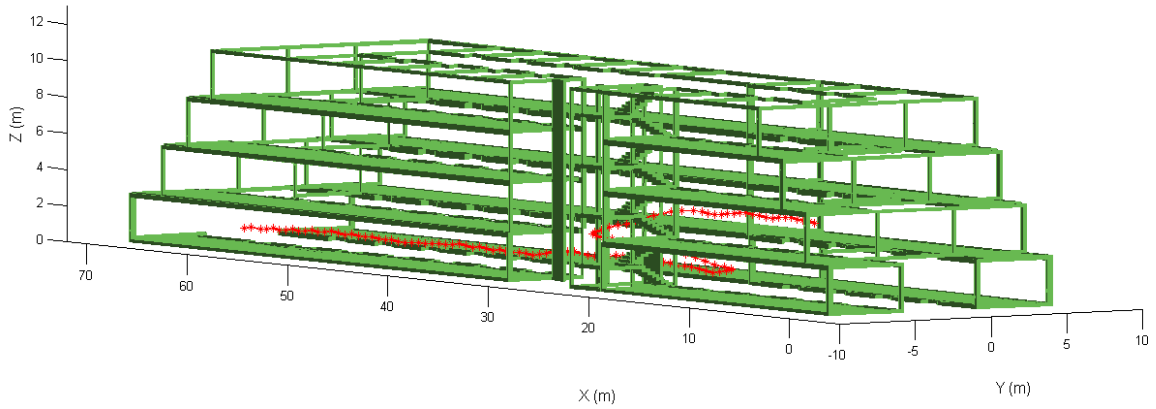


Figure 5.16. Side 3-D view from experiment 3, test 1.

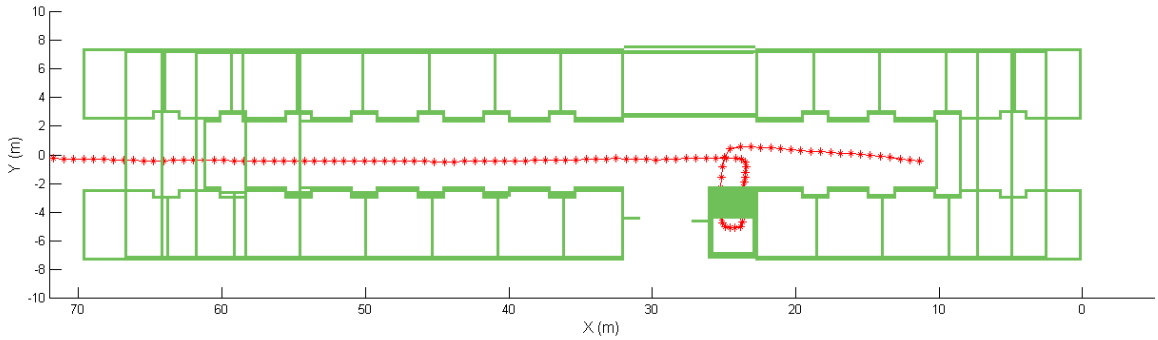


Figure 5.17. Top view from experiment 3, test 1.

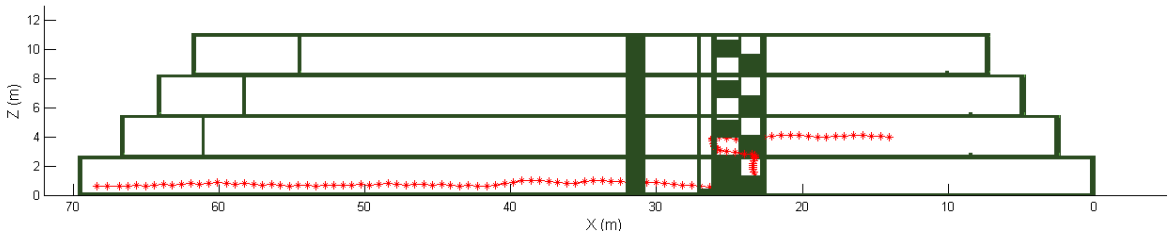


Figure 5.18. Lateral view from experiment 3, test 1.

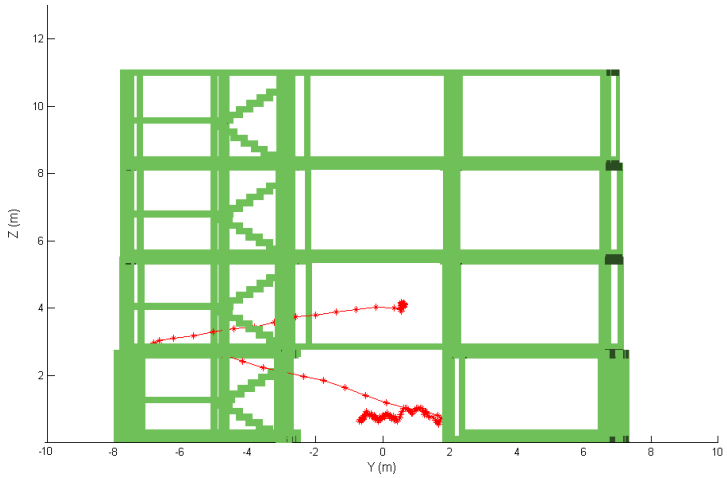


Figure 5.19. Frontal view from experiment 3, test 1.

Table 5.10. Step detection results from experiment 3, test 1.

Subject/ Height (m)		Number of detected steps										Error (%)
		test 1		test 2		test 3		test 4		test 5		
		Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	
S1	1.55	96	95	97	96	96	95	96	95	96	95	1.040%
S2	1.63	95	94	94	93	94	93	94	93	94	93	1.062%
S3	1.64	95	94	95	94	96	95	94	93	95	94	1.053%
S4	1.65	84	83	84	83	85	84	84	83	84	83	1.188%
S5	1.66	95	94	94	93	95	94	96	95	96	95	1.050%
S6	1.68	87	86	88	87	87	86	87	86	87	86	1.147%
S7	1.71	91	90	90	89	90	89	91	90	91	90	1.104%
S8	1.73	81	80	82	81	80	79	82	81	80	79	1.235%
S9	1.74	89	88	88	87	90	89	88	87	90	89	1.124%
S10	1.75	82	81	82	81	81	80	82	81	82	81	1.223%
S11	1.80	80	79	81	80	80	79	80	79	80	79	1.247%
S12	1.82	73	72	72	71	72	71	74	73	74	73	1.370%
S13	1.86	70	69	71	70	71	70	70	69	69	68	1.425%
S14	1.92	67	66	68	67	67	66	67	66	67	66	1.488%
										% Overall error		1.197%
										% Overall accuracy		98.803%

Table 5.11. Travelled distance estimation results for test 1 of the experiment 3 (66 m).

Subject	Step length average (m)	Estimated distance (m)					Error (%)
		test 1	test 2	test 3	test 4	test 5	
S1	0.687	46.147	67.13	66.81	66.28	67.95	23.76%
S2	0.698	71.918	65.63	65.92	66.04	65.66	4.01%
S3	0.702	65.273	65.253	67.103	66.613	65.653	5.41%
S4	0.71	56.99	67.98	67.48	66.06	66.08	8.20%
S5	0.703	65.393	67.363	65.173	66.413	65.253	4.02%
S6	0.7226	61.772	62.182	61.012	60.882	62.442	0.92%
S7	0.802	71.388	72.138	71.188	70.808	72.388	1.36%
S8	0.862	68.128	69.098	69.028	69.168	67.338	3.11%
S9	0.875	76.152	76.552	78.102	75.192	76.812	2.69%
S10	0.907	72.55	72.15	74.55	73.22	72.95	3.68%
S11	0.978	76.299	76.439	75.589	77.459	77.719	2.63%
S12	0.971	68.959	69.609	68.179	68.969	68.649	0.62%
S13	9.83	66.963	67.783	67.243	68.853	66.663	4.02%
S14	1.014	66.495	67.955	66.635	67.905	67.945	6.71%
						% Overall error	5.082%
						% Overall accuracy	94.918%

5.3.3.2 Test 2

The second test uses the same path of test 1, but in the reverse direction. In this case the person starts from the access door at the ground floor and stops in front the office 1204, at the first floor. The total distance of this trajectory is 66 m. From figure 5.20 to figure 5.23 and tables 5.12 and 5.13 present the results.

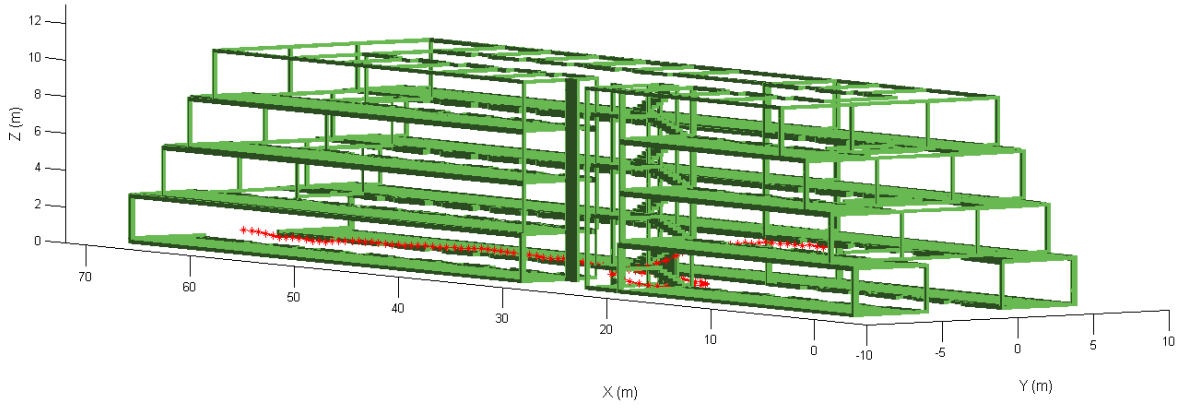


Figure 5.20. Side 3-D view from experiment 3, test 2.

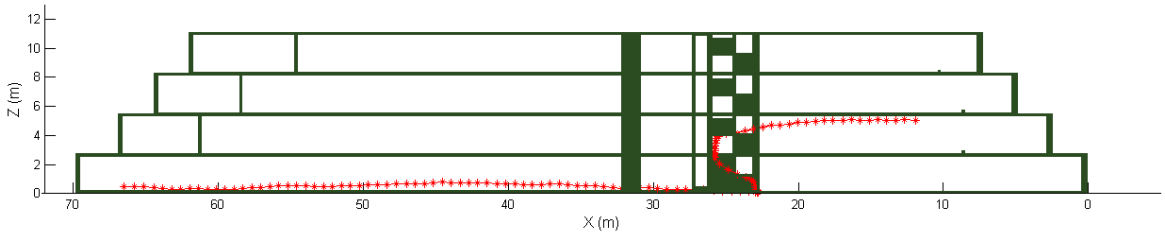


Figure 5.21. Lateral view from experiment 3, test 2.

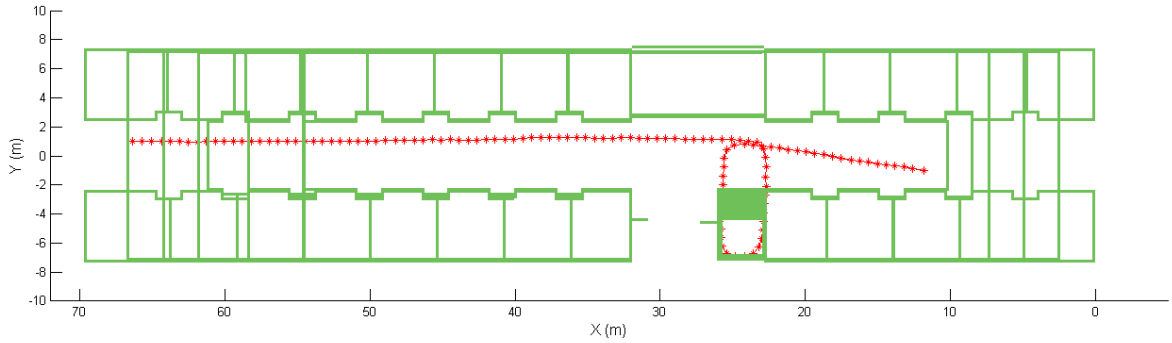


Figure 5.22. Top view from experiment 3, test 2.

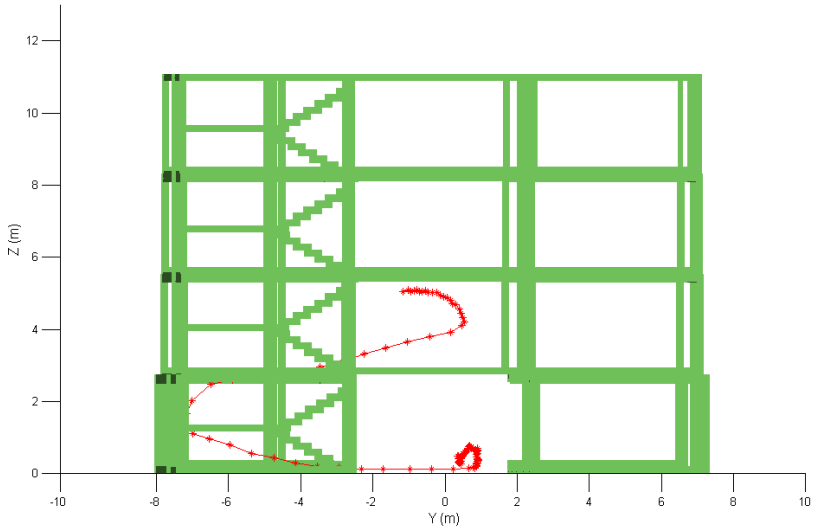


Figure 5.23. Frontal view from experiment 3, test 2.

Table 5.12. Step detection results from experiment 3, test 2.

Subject/ Height (m)		Number of detected steps										Error (%)
		test 1		test 2		test 3		test 4		test 5		
		Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	
S1	1.55	94	93	93	92	96	95	95	94	93	92	1.062%
S2	1.63	93	92	94	93	91	90	92	91	94	93	1.078%
S3	1.64	95	94	94	93	96	95	94	93	94	93	1.057%
S4	1.65	85	84	84	83	85	84	85	84	86	85	1.177%
S5	1.66	96	95	94	93	96	95	96	95	95	94	1.048%
S6	1.68	86	85	86	85	84	83	85	84	85	84	1.174%
S7	1.71	91	90	90	89	93	92	92	91	92	91	1.092%
S8	1.73	78	77	80	79	77	76	77	76	79	78	1.279%
S9	1.74	90	89	89	88	89	88	91	90	89	88	1.116%
S10	1.75	81	80	81	80	84	83	81	80	80	79	1.229%
S11	1.80	82	81	80	79	80	79	81	80	81	80	1.238%
S12	1.82	76	75	76	75	79	78	76	75	75	74	1.309%
S13	1.86	67	66	66	65	66	65	68	67	66	65	1.502%
S14	1.92	65	64	63	62	63	62	65	64	65	64	1.558%
										% Overall error		1.208%
										% Overall accuracy		98.792%

Table 5.13. Travelled distance estimation results for test 2 of the experiment 3 (66 m).

Subject	Step length average (m)	Estimated distance (m)					Error (%)
		test 1	test 2	test 3	test 4	test 5	
S1	0.687	66.14	67.3	67.04	65.2	65.15	1.26%
S2	0.698	65.88	67.58	65.65	66.5	68.13	1.86%
S3	0.702	67.413	66.463	66.093	66.933	66.363	10.57%
S4	0.71	67.35	66.57	65.75	67.53	65.68	4.36%
S5	0.703	65.643	65.463	65.813	64.443	66.973	2.10%
S6	0.7226	62.352	61.852	60.982	60.922	61.052	2.75%
S7	0.802	72.418	72.838	71.638	72.888	71.178	5.63%
S8	0.862	69.988	68.068	68.538	67.168	68.618	2.55%
S9	0.875	77.842	76.892	77.132	75.232	76.682	3.97%
S10	0.907	73.65	72.49	73.44	73.03	73.64	4.82%
S11	0.978	76.209	75.759	75.729	77.889	75.649	0.34%
S12	0.971	69.089	68.029	68.719	68.629	69.499	1.20%
S13	9.83	68.573	66.053	67.733	67.803	68.423	5.63%
S14	1.014	67.935	66.575	66.335	67.755	65.775	2.86%
						% Overall error	3.565%
						% Overall accuracy	96.435%

5.3.4 Experiment 4

5.3.4.1 Test 1

For the first test, the trajectory starts from the office 1204 in front of the reading room “Guillermo Haro”, located at the first floor of the building one. The user walks in a straight line until he/she arrive to the stairs, then he/she goes upstairs to the third floor, turns right and walks in a straight line towards the last cubiculum. This trajectory include three level changes with a total distance of 51 m. From figure 5.24 to figure 5.28 and tables 5.14 and 5.15 present the results.

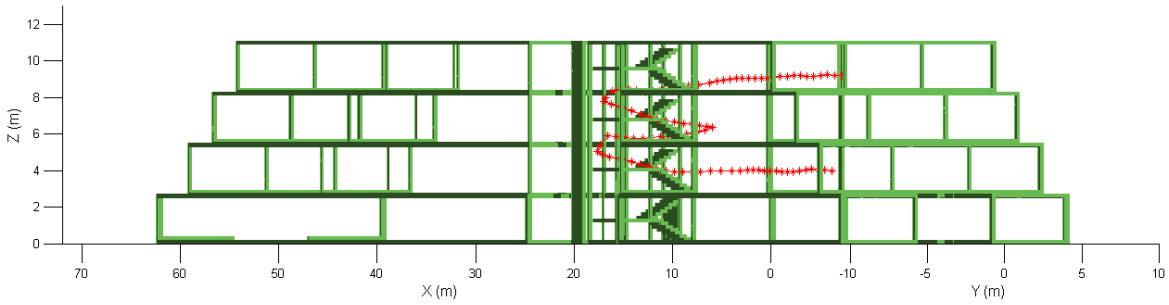


Figure 5.24. Side 3-D view from experiment 4, test 1.

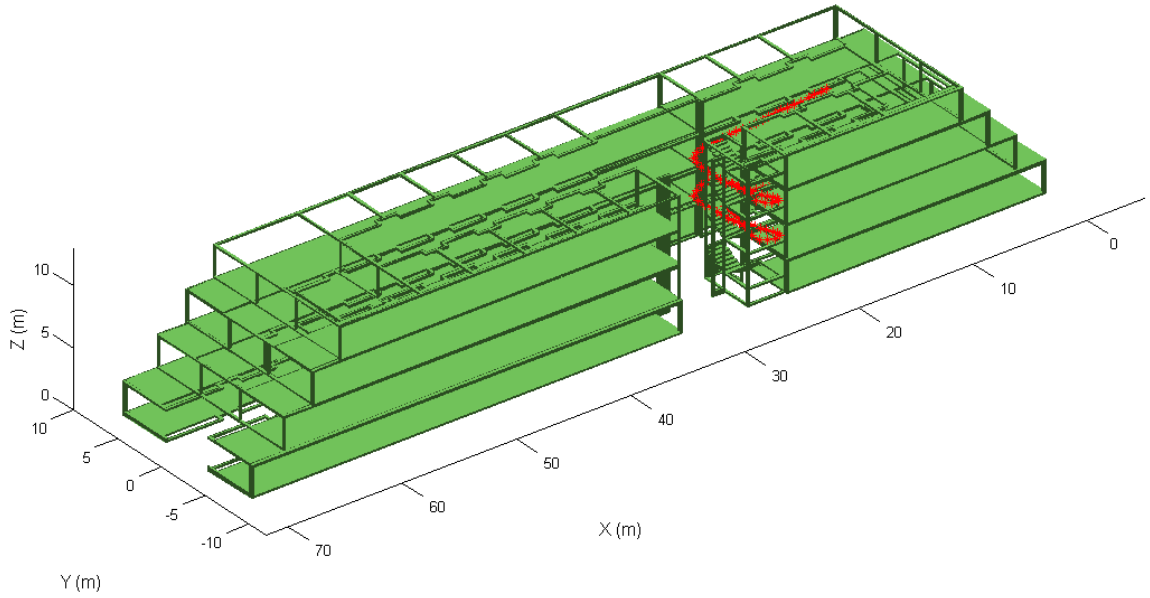


Figure 5.25. Side 3-D view 2 from experiment 4, test 1.

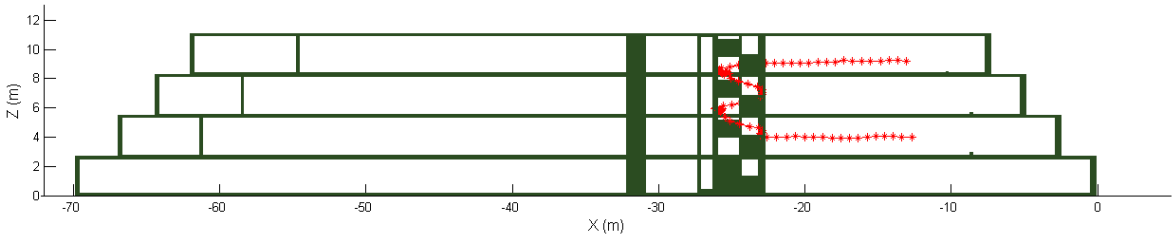


Figure 5.26. Lateral view from experiment 4, test 1.

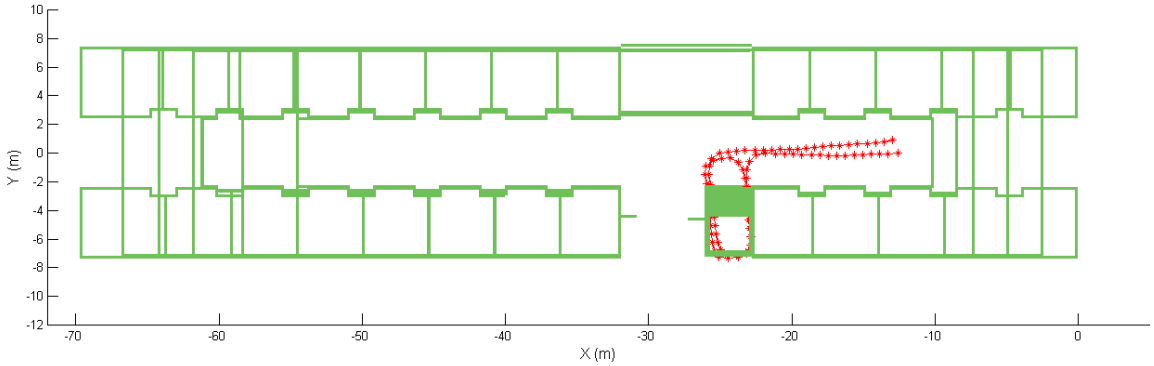


Figure 5.27. Top view from experiment 4, test 1.

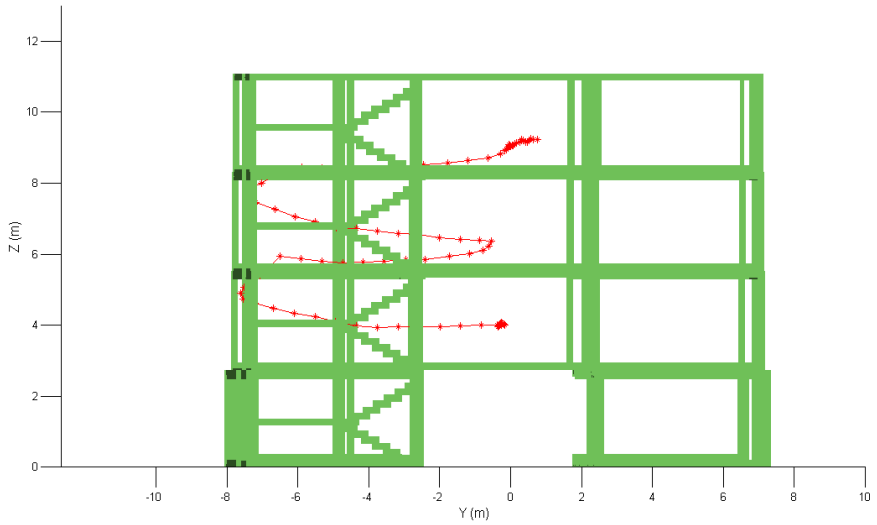


Figure 5.28. Frontal view from experiment 4, test 1.

Table 5.14. Step detection results from experiment 4, test 1.

Subject/ Height (m)		Number of detected steps										Error (%)
		test 1		test 2		test 3		test 4		test 5		
		Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	
S1	1.55	84	83	83	82	85	84	85	84	83	82	1.191%
S2	1.63	82	81	83	82	81	80	82	81	83	82	1.217%
S3	1.64	74	73	74	73	74	73	75	74	74	73	1.348%
S4	1.65	72	71	71	70	71	70	73	72	73	72	1.389%
S5	1.66	76	75	77	76	75	74	77	76	75	74	1.316%
S6	1.68	65	64	66	65	65	64	64	63	65	64	1.539%
S7	1.71	69	68	69	68	69	68	70	69	69	68	1.445%
S8	1.73	62	61	61	60	63	62	61	60	61	60	1.624%
S9	1.74	68	67	69	68	68	67	69	68	69	68	1.458%
S10	1.75	67	66	68	67	68	67	67	66	66	65	1.488%
S11	1.80	63	62	62	61	63	62	62	61	64	63	1.593%
S12	1.82	55	54	55	54	54	53	54	53	54	53	1.838%
S13	1.86	50	49	50	49	51	50	50	49	49	48	2.000%
S14	1.92	54	53	54	53	54	53	53	52	54	53	1.859%
										% Overall error		1.522%
										% Overall accuracy		98.478%

Table 5.15. Travelled distance estimation results for test 1 of the experiment 4 (51 m).

Subject	Step length average (m)	Estimated distance (m)					Error (%)
		test 1	test 2	test 3	test 4	test 5	
S1	0.694	46.147	54.485	53.175	53.475	52.255	8.13%
S2	0.705	56.376	55.586	56.706	57.166	57.656	2.86%
S3	0.702	50.511	52.451	51.821	50.831	50.451	6.95%
S4	0.705	49.315	50.795	50.815	48.925	49.095	4.81%
S5	0.707	52.305	51.715	53.275	51.635	51.645	1.82%
S6	0.724	45.591	50.32	50.93	50.75	50.1	14.33%
S7	0.792	53.071	54.171	53.201	52.951	54.841	5.43%
S8	0.868	52.08	53.66	52.82	53.84	53.85	11.23%
S9	0.877	57.865	57.165	57.535	58.075	57.985	1.21%
S10	0.91	59.151	60.841	58.381	58.771	59.221	1.03%
S11	9.74	59.416	59.046	60.346	58.996	59.406	0.22%
S12	0.975	51.67	51.81	52.35	53.24	52.71	6.64%
S13	0.978	49.92	51.44	51.57	51.06	50.64	4.22%
S14	1.054	53.255	54.735	53.975	54.965	54.615	9.90%
						% Overall error	5.625%
						% Overall accuracy	94.375%

5.3.4.2 Test 2

The second test is based on the same path described above in the previous test, but in this case starting from the third floor, and ending at the reading room "Guillermo Haro", at the first floor. The results are presented in the figures 5.29 to 5.32 and the tables 5.16 y 5.17.

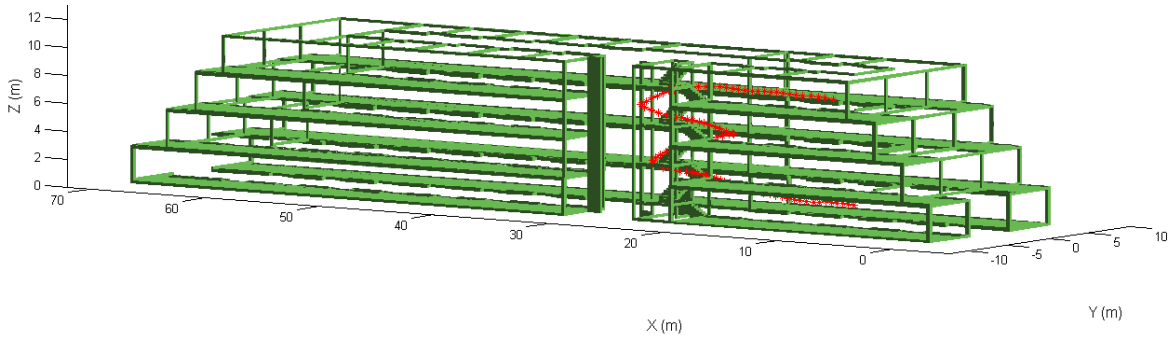


Figure 5.29. Side 3-D view from experiment 4, test 2.

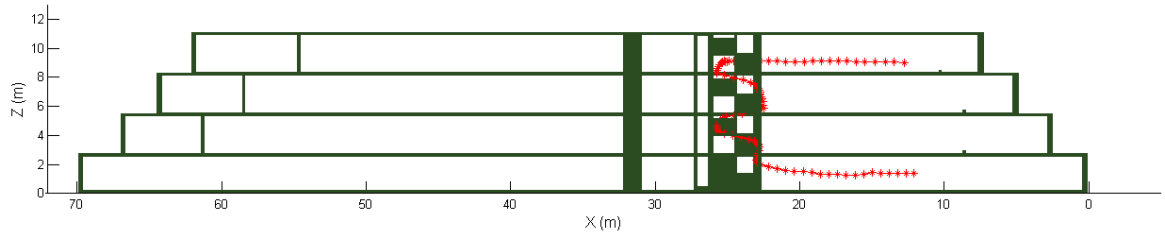


Figure 5.30. Lateral view from experiment 4, test 2.

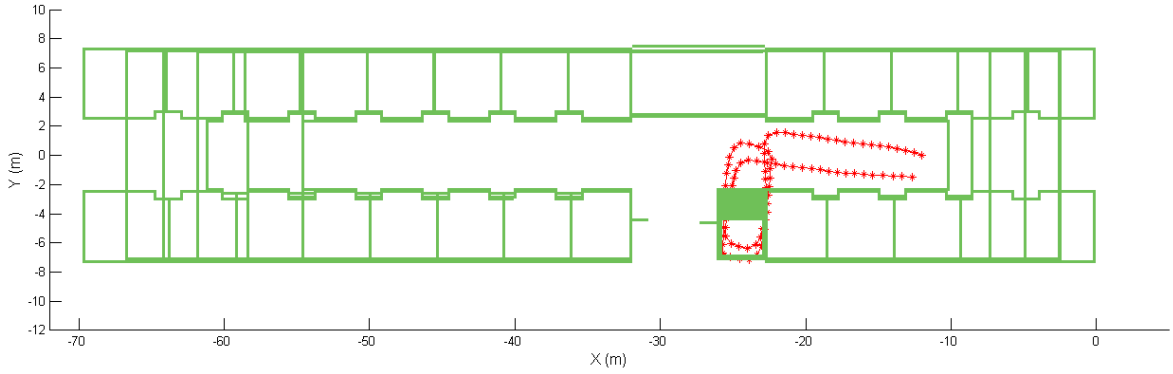


Figure 5.31. Top view from experiment 4, test 2.

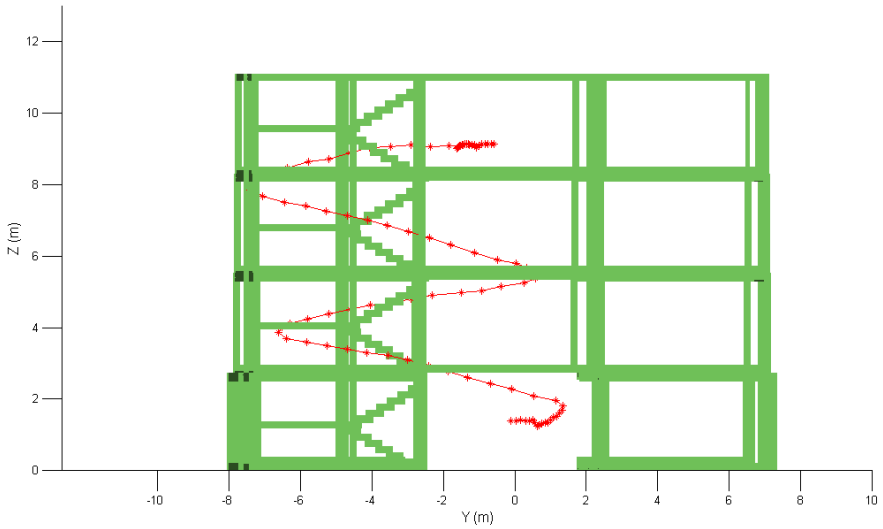


Figure 5.32. Frontal view from experiment 4, test 2.

Table 5.16. Step detection results from experiment 4, test 2.

Subject/ Height (m)		Number of detected steps										Error (%)
		test 1		test 2		test 3		test 4		test 5		
		Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	Real	Detec.	
S1	1.55	84	83	86	85	82	81	83	82	84	83	1.194%
S2	1.63	85	84	85	84	87	86	85	84	86	85	1.168%
S3	1.64	74	73	73	72	77	76	75	74	74	73	1.341%
S4	1.65	74	73	72	71	74	73	73	72	75	74	1.359%
S5	1.66	73	72	72	71	75	74	73	72	74	73	1.363%
S6	1.68	66	65	64	63	65	64	67	66	66	65	1.525%
S7	1.71	71	70	71	70	72	71	71	70	71	70	1.405%
S8	1.73	61	60	61	60	61	60	61	60	61	60	1.639%
S9	1.74	68	67	69	68	70	69	67	66	68	67	1.462%
S10	1.75	67	66	67	66	69	68	66	65	66	65	1.493%
S11	1.80	65	64	65	64	63	62	65	64	65	64	1.548%
S12	1.82	55	54	53	52	53	52	55	54	56	55	1.839%
S13	1.86	49	48	50	49	47	46	48	47	49	48	2.059%
S14	1.92	54	53	55	54	57	56	53	52	54	53	1.833%
										% Overall error		1.516%
										% Overall accuracy		98.484%

Table 5.17. Travelled distance estimation results for test 2 of the experiment 4 (51 m).

Subject	Step length average (m)	Estimated distance (m)					Error (%)
		test 1	test 2	test 3	test 4	test 5	
S1	0.694	54.605	53.635	52.305	54.115	51.855	5.10%
S2	0.705	57.106	58.136	55.576	57.946	57.856	8.41%
S3	0.702	50.431	51.771	51.531	50.141	51.121	4.83%
S4	0.705	51.375	49.385	48.985	48.955	48.515	1.30%
S5	0.707	54.385	52.735	51.915	51.585	52.265	2.60%
S6	0.724	51.68	52.9	52.61	51.95	52.62	13.25%
S7	0.792	52.811	54.101	54.261	52.291	52.461	1.07%
S8	0.868	51.34	53.08	53.24	53.55	54.07	9.37%
S9	0.877	57.535	57.175	59.135	57.395	58.575	0.85%
S10	0.91	59.891	60.141	59.151	58.261	59.431	1.89%
S11	9.74	59.736	59.146	59.326	59.076	59.876	0.13%
S12	0.975	52.46	53.36	53.04	53.09	52.33	11.48%
S13	0.978	51.27	50.62	51.25	51.84	50.28	5.47%
S14	1.054	54.375	54.615	53.835	53.265	53.245	5.75%
						% Overall error	5.107%
						% Overall accuracy	94.893%

5.3.5 General results

In general the step detection algorithm successfully detects the pedestrian steps, even when a change in the activity occurred. Table 5.18 presents the general results with a detection rate of 98.736%. Table 5.19 presents the results corresponding to step-length estimation, and the computation of total travelled distance presented an accuracy of 94.88% in average.

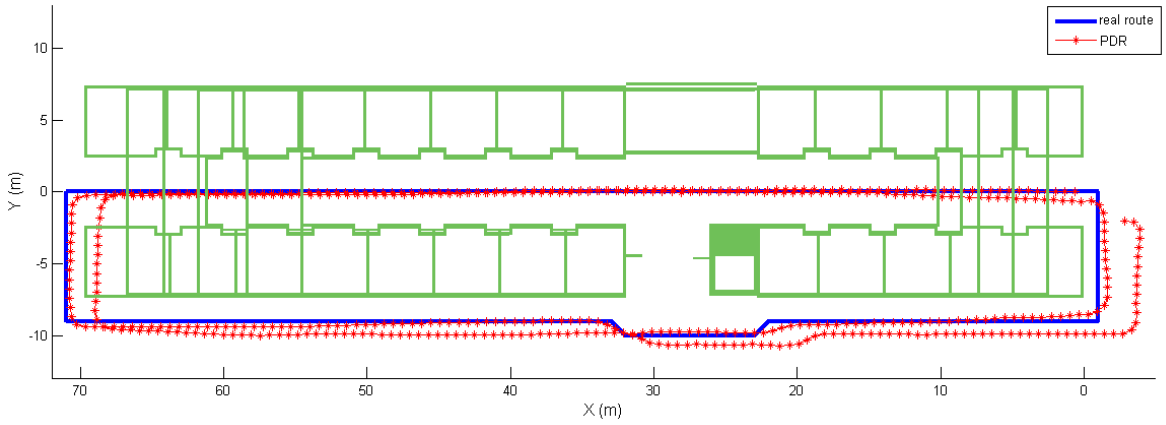
Table 5.18. Accuracy (%) results of step detection

Experiment 1	Experiment 2		Experiment 3		Experiment 4		Overall accuracy (%)	Overall error (%)
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2		
99.73	98.428	98.437	98.803	98.792	98.478	98.484	**98.74%**	**1.26%**

Table 5.19. Accuracy (%) results of distance estimation

Experiment 1	Experiment 2		Experiment 3		Experiment 4		Overall accuracy (%)	Overall error (%)
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2		
94.263	95.290	94.024	94.918	96.435	94.375	94.893	**94.88%**	**5.12%**

The position error was measured using only the results of the experiment 1, because this trajectory represents the longest travelled distance, therefore it presents the major cumulative error. Then, the mean position error is 0.77 m. Figure 5.33 shows the trajectories of the real trajectory and the obtained trajectory for the experiment 1.



5.33. Comparison between the real route and the obtained trajectory.

In order to compare the results presented in the table 5.19 with others results reported in the recent literature, table 5.20 presents a brief summary of other PDR systems. It is noteworthy to mention that all of these works use a higher number of sensors than the scheme proposed in this work. In [84] a sensor fusion framework for combining WiFi, PDR and landmarks is presented. The work in [85] proposes a new method for step detection and stride estimation in a PDR system. In [86] an axis-exchanged compensation and gait parameter analysis algorithm to improve the navigation accuracy in a PDR system is presented. The work presented in [87] proposes a new pedestrian dead reckoning (PDR)-based navigation algorithm which uses an extended Kalman filter to determinate the user's heading direction for each step by using magnetic, angular rate, and gravity (MARG)

sensors. Finally, in [88], the positioning algorithm implements a loosely coupled GPS/PDR integration.

Table 5.20. A brief summary of the PDR systems reported recent literature.

Ref.	Publication Year	System sensors	Evaluated test	Test subjects	Error	
					mean position error	mean distance error
[84]	2015	Smartphone sensors: accelerometer, magnetometer, gyroscope, and barometer.	Walking in a 2-D indoor trajectory	1	0.9945 m	
[85]	2015	Accelerometer, magnetometer, and gyroscope mounted on the ankle.	Walking in a 2-D indoor trajectory	1		2.35% of a travelled distance of 66.6 m
[86]	2015	Accelerometer, gyroscope, and magnetometers mounted on the waist.	Walking in a 2-D indoor trajectory	10		0.42% of a travelled distance of 40 m
[87]	2014	Accelerometer, gyroscope, and magnetometers mounted on the waist.	Walking in a 2-D outdoor trajectory	10	5.638 m	1.5821% of a travelled distance of 400 m
[88]	2014	Accelerometer, gyroscope, magnetometers, and GPS.	Walking in a 2-D outdoor trajectory	1	5 m	Below 3% of the total traveled distance.

5.4 Summary

In this chapter, the description and results obtained from the different developed experiments, in order to validate the PDR system, were presented. In relation to attitude estimation, the experimental results indicate an improvement when ANFIS-based adaptation (KF-NFA) is used. The activity detection procedure was incorporated into the step detection and the step length estimation algorithms providing an average classification rate of 87.49%. Attitude estimation and step length estimation were integrated into a Pedestrian Dead Reckoning system.

Chapter 6

Conclusions

This work presented the development of a Pedestrian Dead Reckoning (PDR) system, relying on information derived from triaxial accelerometer and gyroscope sensors contained in an inertial measurement unit (IMU), which was attached to the pedestrian waist. The PDR system has been developed aiming to provide information about location and movements of a person travelling in a GPS-denied environment.

In order to improve the accuracy of the attitude estimation reducing the time-varying drift, a Kalman Filter with Neuro-Fuzzy adaptation (KF-NFA) was developed. The purpose of an adaptive Kalman filter formulation is to reduce the errors in the estimation by modifying or adapting the Kalman filter to the real data. The adaptation process was performed on the filter statistical information matrix *R*, which is tuned using an Adaptive Neuro Fuzzy Inference System (ANFIS) based on the filter innovation sequence through a covariance-matching technique.

The KF-NFA estimates and corrects the Euler angles (roll, pitch and yaw) through the fusion of inertial measurements. The process model was modeled with linear equations representing the gyroscope, accelerometer and bias measures. Because of this, it was discarded the use of another fusion technique, as the Extended Kalman filter (EKF). The test results showed a better performance of the KF-NFA when it is compared with a traditional Kalman Filter, which was reflected in a better performance on the attitude estimation.

Besides, in order to improve the calculation of the distance traveled by the pedestrian, and consequently the localization accuracy, a dynamical method for estimating the step length using a Fuzzy Inference System (FIS) was developed, which uses the pedestrian's activity as an additional input. The valid steps are detected by calculating the area under each accelerometer peak and the different types of activities are classified using a Multi-layer Perceptron (MLP) Neural Network (NN) according to features extracted based on wavelet decomposition. The wavelet decomposition is applied to triaxial accelerometer data in order to identify points in the signal where a pedestrian changes from one activity to another.

The activity classification was focused on activities that a pedestrian performs routinely in his daily life, such as walking, walking fast, jogging and running. Results provided an accuracy of 98.74% for step detection, together with an average correct classification rate of 87.49%, which prove that wavelet decomposition and neural networks represents a good choice of feature extraction and classification system, respectively. Subsequently, the step length was dynamically estimated using a FIS which incorporates classification activity during the estimation process. This operation allows the system to adjust the computation of the total distance travelled by a pedestrian according to the ongoing activity. The accuracy on step-length estimation is about 94.88% in average.

The final tests includes the integration of height measurements obtained from the altimeter DEMOSTBMPL3115A2, which was attached over the IMU STEVAL-MKI062V2 to the pedestrian waist. This was done considering future applications, where the proposed algorithm could be implemented on a mobile platform, such as a smartphone or tablet. The use of the altimeter allowed to perform extensive testing in walking paths including stairs. The experiments were performed in combined indoor/outdoor trajectories in the premises of the building number one at the INAOE. The trajectories followed in these experiments were designed in such a way that level changes and 90° turns were present. The general results show a good performance in the PDR system. As future work it is proposed:

- To incorporate a GPS sensor to the PDR system. Thus, it will be possible in order to maintain a geographical reference of the pedestrian position when she is traveling in a GPS environment.
- To implement a real-time PDR system into a smartphone or tablet. Thus, a located based service (LBS) can be developed. Possible applications are information directory services to be used in public places, such as museums and shopping centers.
- To improve the classification accuracy.

Publications

- Mariana N. Ibarra-Bonilla, P. Jorge Escamilla-Ambrosio, Juan Manuel Ramírez-Cortes, "Attitude estimation using a Neuro-Fuzzy tuning based adaptive Kalman filter", *Journal of Intelligent and Fuzzy Systems*, IOS Press, (Science Citation Index and JCR), DOI 10.3233/IFS-141183, 2014.

- Mariana Natalia Ibarra-Bonilla, Ponciano Jorge Escamilla-Ambrosio, Juan Manuel Ramirez-Cortes, Jose Rangel-Magdaleno, and Pilar Gomez-Gil, "Step Length Estimation and Activity Detection in a PDR System Based on a Fuzzy Model with Inertial Sensors", *Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Series 7092, Springer* International Publishing Switzerland, 2014.

- Mariana N. Ibarra, J. Manuel Ramirez, P.Jorge Escamilla-Ambrosio, Pilar Gomez-Gil, "Pedestrian Dead Reckoning: a neuro-fuzzy approach with inertial measurements fusion based on Kalman filter and DWT", *IEEE International Seminar of Computational Intelligence ISCI 2014*, Computational Intelligence Society, Tijuana, Mexico October 28-29, 2014.

- Mariana N. Ibarra-Bonilla, P. Jorge Escamilla-Ambrosio, J. Manuel Ramirez-Cortes, Carlos Vianchada, "Pedestrian Dead Reckoning with Attitude Estimation using a Fuzzy Logic Tuned Adaptive Kalman Filter*", Latinamerican Symposium on Circuits and Systems LASCAS 2013*, Cusco, Peru, February, 2013.

- Mariana Natalia Ibarra, P. J. Escamilla-Ambrosio, J. M Ramirez-Cortes, "Pedestrian dead reckoning towards indoor location based applications", *8th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2011*, October 26-28, Yucatán, México, 2011.

References

[1]. M. Grewal, L.R. Weill, A.P. Andrews, "Global positioning systems, inertial navigation, and integration", USA, John Wiley & Sons, 2007.

[2]. G. Gartner, F. Ortag. "Advances in Location-based Services, Lecture Notes in Geoinformation and Cartography". Springer-Verlag, Berlin, Heidelberg, 2012.

[3]. S.A. Ahson and M. Ilyas. "Location-based services: fundamentals and operation", Boca Raton, CRC Press, Taylor & Francis Group, 2011.

[4]. F. Aubeck, C. Isert, D. Gusenbauer, "Camera based step detection on mobile phones", International Conference on Indoor Positioning and Indoor Navigation (IPIN), Guimarães, Portugal, pp. 21-23, September, 2011.

[5]. C. Hide, T. Botterill, M. Andreotti, "Vision-aided IMU for handheld pedestrian navigation", Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010), Portland, OR, pp. 534-541, September, 2010.

[6]. Y. Jang, S. Shin, J.W. Lee, S. Kim, "A Preliminary Study for Portable Walking Distance Measurement System Using Ultrasonic Sensors", 29th Annual International Conference of the IEEE EMBS, Lyon, France, pp. 5290–5293, August, 2007.

[7]. G. Girard, S. Cote, S. Zlatanova, Y. Barette, J. St-Pierre, P. Van-Oosterom, "Indoor Pedestrian Navigation Using Foot-Mounted IMU and Portable Ultrasound Range Sensors", Sensors, Vol. 11, No. 8, pp. 7606-7624, 2011.

[8]. J. Rudan, Z. Tuza, "Using LMS-100 laser rangefinder for indoor metric map building", IEEE International Symposium on Industrial Electronics (ISIE2010), Bari, Italy, pp. 525-530, July, 2010.

[9]. R.W. Levi, T. Judd, "Dead Reckoning Navigation System Using Accelerometer to Measure Foot Impacts", U.S. Patent 5 538 776, December, 1996.

[10]. S.H. Shin, C.G. Park, J.M. Lee, "Adaptive step length estimation algorithm using low-cost MEMS inertial sensors", IEEE Sensors Applications Symposium, San Diego, CA, USA, pp. 1-5, February, 2007.

[11]. M. Chowdhary, M. Sharma, A. Kumar, S. Dayal, M. Jumar, "Robust Attitude Estimation for Indoor Pedestrian Navigation using MEMS Sensors", Institute of Navigation GNSS, Nash-ville, Tennessee, September 2012.

[12]. S. K. Park, Y. S. Suh, "A Zero Velocity Detection Algorithm Using Inertial Sensors for Pedestrian Navigation Systems", Sensors, Vol. 10, No. 10, pp. 9163-9178, 2010.

[13]. Y.S. Suh, S.K. Park, "Pedestrian inertial navigation with gait phase detection assisted zero velocity updating", 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, pp. 336-341, February, 2009.

[14]. A.R. Jimenez, F. Seco, F. Zampella, C. Prieto, J. Guevara, "PDR with a Foot-Mounted IMU and Ramp Detection", Sensors, Vol. 11, No. 10, pp. 9393-9410, 2011.

[15]. K. Altun, B. Barshan, "Pedestrian dead reckoning employing simultaneous activity recognition cues", Measurement Science and Technology, Vol. 23, No. 2, pp. 1-20, 2012.

[16]. O.J. Woodman, "An introduction to inertial navigation", Technical Report 696, University of Cambridge, 2007.

[17]. Z. Sun, X. Mao, W. Tian, X. Zhang, "Activity classification and dead reckoning for pedestrian navigation with wearable sensors", Measurement Science and Technology, Vol. 20, No. 1, pp. 1-10, 2009.

[18]. M. N. Ibarra, P. J. Escamilla, J.M. Ramírez, "Pedestrian dead reckoning towards indoor location based applications", 8th International Conference on Electrical Engineering Computing Science and Automatic Control, Yucatán, México, October, 2011.

[19]. Winter D. A., "Biomechanics and motor control of human movement", second edition, John Wiley &Sons Inc, 1990.

[20]. M. Susi, V. Renaudin, G. Lachapelle, "Motion mode recognition and step detection algorithms for mobile phone users", Sensors, vol. 12, no. 2, pp. 1539-1562, 2013.

[21]. C. C. Yang, Y. L. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring", Sensors, vol. 10, no. 8, pp. 7772–7788, 2010.

[22]. G. Panahandeh, N. Mohammadiha, A. Leijon, P. Handel, "Continuous Hidden Markov Model for Pedestrian Activity Classification and Gait Analysis", IEEE Transactions on Instrumentation and Measurement, vol.62, no.5, pp.1073-1083, May, 2013.

[23]. Mannini A., Sabatini A. M., "Accelerometry-based classification of human activities using markov modeling", Computational intelligence and neuroscience, vol. 2011, pp. 2-9, 2011.

[24]. Preece, S.J., Goulermas, J.Y., Kenney, L. P. J., Howard D., "A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data", IEEE Transactions on Biomedical Engineering, vol.56, no.3, pp. 871-879, March, 2009.

[25]. Ravi N., Dandekar N., Mysore P., Littman M. L., "Activity recognition from accelerometer data", American Association for Artificial Intelligence, vol. 5, pp. 1541–1546, 2005.

[26]. Kwapisz J. R., Weiss G. M., Moore S. A., "Activity recognition using cell phone accelerometers", ACM SIGKDD Explorations Newsletter, vol. 12, no. 2, pp. 74-82, 2011.

[27]. Ning W., Ambikairajah, E., Lovell, N. H., Celler, B.G., "Accelerometry Based Classification of Walking Patterns Using Time-frequency Analysis", Engineering in Medicine and Biology Society (EMBS), 29th Annual International Conference of the IEEE , Lyon, France, pp.4899-4902, 22-26 August, 2007.

[28]. Yunqian M. A., HESCH, J. A., "Gait Classification Using Wavelet Descriptors in Pedestrian Navigation", ION GNSS, Portland, OR, pp. 1328-1337, 17-21 September, 2011.

[29]. G. Panahandeh, N. Mohammadiha, A. Leijon, P. Händel, "Chest-mounted inertial measurement unit for pedestrian motion classification using continuous hidden Markov model", 2012 IEEE International Instrumentation and Measurement Technology Conference, Graze, Austria, pp. 991-995, 13-16 May, 2012.

[30]. J.F. Li, Q.H. Wang, X. Liu, M.Y. Zhang, "An Autonomous Waist-Mounted Pedestrian Dead Reckoning System by Coupling Low-Cost MEMS Inertial Sensors and GPS Receiver for 3D Urban Navigation", Journal of Engineering Science and Technology Review, vol. 7, no. 1, pp. 9-14, 2014.

[31]. T. Moder, P. Hafner, K. Wisiol, M. Wieser, "3D Indoor Positioning with Pedestrian Dead Reckoning and Activity Recognition based on Bayes Filtering", Fifth International Conference on Indoor positioning and indoor navigation, 27-30 October, Buscan, Korea, 2014.

[32]. H. Leppäkoski, J. Collin, J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals", Journal of Signal Processing Systems, vol. 71, no. 3, pp. 287-296, 2013.

[33]. K.C. Lan, W.Y. Shih, "On Calibrating the Sensor Errors of a PDR-Based Indoor Localization System", Sensors, vol. 13, no. 4, pp. 4781-4810, 2013.

[34]. T. Gädeke, J. Schmid, W. Stork, K.D. Müller-Glaser, "Pedestrian dead reckoning for person localization in a wireless sensor network", International Conference on indoor positioning and indoor navigation (IPIN), 21-23 September, Guimarães, Portugal, 2011.

[35]. A. Mannini, A. M. Sabatini, "Accelerometry-based classification of human activities using markov modeling", Computational intelligence and neuroscience, vol. 2011, pp. 1-10, 2011.

[36]. J. Bancroft, D. GARRETT, G. Lachapelle, "Activity and environment classification using foot mounted navigation sensors", International Conference on Indoor Positioning and Indoor Navigation, 13-15 November, Sydney, Australia, 2012.

[37]. F. Tanveer, O. T. Waheed, A-U-Rehman, "Design and development of a sensor fusion based low cost attitude estimator", Journal of Space Technology, vol. 1, no. 1, pp. 45-50, 2011.

[38]. H. Fourati, N. Manamanni, "Position estimation approach by complementary filter-aided IMU for indoor environment", 12th Biannual European Control Conference, Zurich, Switzerland, 2013.

[39]. L. Ruotsalainen, J. Bancroft, H. Kuusniemi, G. Lachapelle, R. Chen, "Utilizing Visual Measurements for Obtaining Robust Attitude and Positioning for Pedestrians", Institute of Navigation GNSS, Nashville, Tennessee, pp. 2454-2461, September, 2012

[40]. X. Cui,Y. Y.Qin, Q. Zhou, Q. W. Fu, "Shoe-Mounted Personal Navigation System Based on MEMS", Advanced Materials Research, vol. 580, pp. 423-427, 2012.

[41]. D. Chen, W. Feng, Q. Zhao, M. Hu, T. Wang, "An infrastructure-free indoor navigation system for blind people", 5th International Conference on Intelligent Robotics and Applications, Montreal, Canada, pp. 552-561, October, 2012.

[42]. S. Ayub, B.M. Heravi, A. Bahraminasab, B. Honary, "Pedestrian Direction of Movement Determination Using Smartphone", 6th International Conference on Next Generation Mobile Applications, Services and Technologies, Paris, France, pp. 64-69, September, 2012.

[43]. C. W. Kang and C. G. Park, "Attitude estimation with accelerometers and gyros using fuzzy tuned Kalman filter", The European Control Conference, Budapest, Hungary, pp. 3713–3718, August, 2009.

[44]. N. Yadav, C. Bleakley, "Two stage Kalman filtering for position estimation using dual Inertial Measurement Units", IEEE Sensors Conference, Limerick, Ireland, pp. 1433-1436, October, 2011.

[45]. F. Zampella, M. Khider, P. Robertson, A. Jimenez, "Unscented Kalman filter and Magnetic Angular Rate Update (MARU) for an improved Pedestrian Dead-Reckoning", IEEE/ION Symposium on Position Location and Navigation Symposium, Myrtle Beach, SC, pp. 129-139, April, 2012.

[46]. H.G. De Marina, F.J. Pereda, J.M. Giron-Sierra, F. Espinoza, "UAV Attitude Estimation Using Unscented Kalman Filter and TRIAD", IEEE Transactions on Industrial Electronics, vol. 59, no. 11, pp. 4465-4474, November, 2012.

[47]. X. Hu, Q. Li, C. He, Y. Liu, "An adaptive kalman filter for three dimensional attitude tracking," IEEE International Symposium on VR Innovation, Singapore, pp.151-154, March, 2011.

[48]. E. Pulido-Herrera, H. Kaufmann, J. Secue, R. Quirós, G. Fabregat, "Improving data fusion in personal positioning systems for outdoor environments", Information Fusion, vol. 14, no. 1, pp. 45-56, 2013.

[49]. W. Li, J. Wang, "Effective Adaptive Kalman Filter for MEMS-IMU/Magnetometers Integrated Attitude and Heading", Reference Systems Journal of Navigation, vol. 66, no. 1, pp. 99-113, January, 2013.

[50]. W. Qin, W. Yuan, H. Chang, L. Xue, G. Yuan, "Fuzzy Adaptive Extended Kalman Filter for miniature Attitude and Heading Reference System", 4th IEEE International

Conference on Nano/Micro Engineered an Molecular Systems, Shenzhen, China, pp. 1026-1030, January, 2009.

[51]. R. Havangi, M.A. Nekoui, M. Teshnehlab, "Adaptive Neuro-Fuzzy Extended Kalman Filtering for robot localization", International Journal of Computer Science, vol. 7, no. 2, pp. 15-23, March, 2010.

[52]. R.E. Kalman, "A new approach to linear filtering and prediction problems", Transactions of the ASME-Journal of Basic Engineering, vol. 82, no. 1, pp. 35-45, 1960.

[53]. G. Welch, G. Bishop, "An Introduction to the Kalman Filter", The University of North Carolina at Chapel Hill, TR95-045, 1995.

[54]. P.J. Escamilla-Ambrosio, "Intelligent adaptive multisensor data fusion using hybrid architectures", Ph.D. dissertation, Department of Automatic Control and Systems Engineering, The University of Sheffield, 2003.

[55]. S.N. Sivanandam, S. Sumathi, S.N. Deepa, "Introduction to fuzzy logic using MATLAB", Springer-Verlag Berlin Heidelberg, 2007.

[56]. M.T. Hagan, H.B. Demuth, M.H. Beale, "Neural Network Design", Boston: Pws, 1996.

[57]. J.S. Jang, "ANFIS: adaptive-network-based fuzzy inference system", IEEE Transactions on Systems, Man and Cybernetics, vol. 23, no. 3, p. 665-685, 1993.

[58]. J.S. Jang, C.T. Sun, "Neuro-fuzzy Modeling and Control", Proceedings of the IEEE, vol. 83, no. 3, pp. 378-406, 1995.

[59]. L.A. Zadeh, "Outline of a New Approach to the analysis of Complex Systems and Decision Processes", IEEE Transactions on Systems, Man and Cybernetics, no. 1, pp. 28-44, 1973.

[60]. J.S. Jang, C.T. Sun, "Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence", Prentice-Hall, NJ, USA, 1997.

[61]. M. Brown, C. Harris, "Neurofuzzy Adaptive Modelling and Control", Prentice Hall International (UK) Limited, 1994

[62]. E. Mamdani, S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", International Journal of Man-Machine Studies, vol. 7, no. 1, pp. 1-13, 1975.

[63]. T. Takagi, M. Sugeno, "Fuzzy identification of systems and its applications to modelling and control", IEEE Transactions on Syst. Man and Cybernetics, no. 15, pp. 116-132, 1985.

[64]. R. Fuller, "Neural fuzzy systems", Advances in Soft Computing Series, Berlin, Germany: Springer-Verlag, 2000.

[65]. M.M. Gupta, D.H. Rao, "Neuro-Control Systems: A Tutorial, Neuro Control Systems Theory and Applications", Madan M. G. and Dandina H. R. Eds., IEEE Press, pp. 1-43, 1994.

[66]. M.B. Priestley, **"**Wavelets and time-dependent spectral analysis", Journal of Time Series Analysis, vol. 17, no.1, pp. 85–103, 2008.

[67]. M.A. Pinsky, "Introduction to Fourier Analysis and Wavelets", Graduate Studies in Mathematics, American Mathematical Society, 2009.

[68]. R. J. Fitzgerald, "Divergence of the Kalman filter", IEEE Transactions on Automatic Control, vol. 16, no. 6, pp. 736- 747, 1971.

[69]. P. J. Escamilla-Ambrosio, N. Mort, "Development of a fuzzy logic-based adaptive Kalman filter", European Control Conference, Porto, Portugal, pp. 1768-1773, September 4-7, 2001.

[70]. P. J. Escamilla-Ambrosio, N. Mort, "Adaptive Kalman filtering through fuzzy logic", Proc. of the 7th UK Workshop On Fuzzy Systems, Recent Advances and Practical Applications of Fuzzy, Neuro-Fuzzy, and Genetic Algorithm-Based Fuzzy Systems, Sheffield, U.K., pp. 67-73, October 26-27, 2000.

[71]. G. Schall, D. Wagner, G. Reitmayr, E. Taichmann, M. Wieser, D. Schmalstieg, B. Hofmann-Wellenhof, "Global pose estima-tion using multi-sensor fusion for outdoor augmented reality", 8th IEEE International Symposium on Mixed and Augmented Reality, Orlando, Florida, pp. 153-162, October, 2009.

[72]. iNEMO ST Microelectronics: UM0937. User manual [Visited 20-05-13]. On line: http://www.st.com/st-webui/static/active/en/resource/technical/document/user_manual/CD00271225.pdf.

[73]. C. Hajiyev, "Nonlinear Robust Adaptive EKF for Identification of EMAs Parameters in the Presence of Sensor Faults", 5th International Conference on Physics and Control, Leon, Spain, September, 2011.

[74]. V. Hassani, A.J. Sorensen, A.M. Pascoal, A.P. Aguiar, "Multiple model adaptive wave filtering for dynamic positioning of marine vessels", American Control Conference (ACC), Montreal, QC., pp.6222-6228, 27-29, June, 2012.

[75]. H. Stearns, M. Tomizuka, "Multiple model adaptive estimation of satellite attitude using MEMS gyros", American Control Conference (ACC), San Francisco, CA., pp. 3490-3495, June 29 -July 1, 2011.

[76]. K. Xiong, T. Liang, L. Youngjun, "Multiple model Kalman filter for attitude determination of precision pointing spacecraft", Acta Astronautica, vol. 68, no. 7, pp. 843-852, 2011.

[77]. B.J, West, N. Scafetta, "Nonlinear dynamical model of human gait", Physical review E, vol. 67, no. 5, 2003.

[78]. S.J. Preece, J.Y. Goulermas, L.P.J. Kenney, D. Howard, "A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities from Accelerometer Data", IEEE Transactions on Biomedical Engineering, vol. 56, no. 3, pp. 871-879, 2009.

[79]. H. Demuth, M. Beale, "Neural Network Toolbox for use with MATLAB", The MathWorks, Inc., 2001.

[80]. Y.K. Kim, J.H. Park, H.W. Kim, S.Y. Hwang, J.M. Lee, "Step Estimation in Accordance with Wear Position using the 3-axis Accelerometer", Proceedings of 3rd SPENALO International Symposium, Bexco, Busan, Korea, September, 2011.

[81]. Y. Nam, "Map-based indoor people localization using an inertial measurement unit", Journal of Information Science and Engineering, vol. 27, no. 4, pp. 1233-1248, 2011.

[82]. M.N. Ibarra, P.J. Escamilla-Ambrosio, J.M. Ramirez, C. Vianchada, "Pedestrian Dead Reckoning with Attitude Estimation using a Fuzzy Logic Tuned Adaptive Kalman Filter", 4th IEEE Latin American Symposium on Circuits and Systems, Cusco, Peru, February, 2013.

[83]. DEMOSTBMPL3115A2: Sensor Toolbox MPL3115A2 Development Kit. User manual [Visited 07-04-14]. On line: http://cache.freescale.com/files/sensors/doc/user_guide/DEMOMPL3115A2QSG.pd

[84]. Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y.C. Soh, L. Xie, Fusion of WiFi, "Smartphone Sensors and Landmarks Using the Kalman Filter for Indoor Localization", Sensors, vol. 15, No. 1, pp. 715-732, 2015.

[85]. T. Chen, L. XU, "An Improved Dead Reckoning Algorithm for Indoor Positioning Based on Inertial Sensors", 2015 International Conference on Electrical, Automation and Mechanical Engineering, Phunket, Thailand, pp. 369-371, July, 2015.

[86]. H. Zhang, J. Zhang, D. Zhou, W. Wang, J. Li, F. Ran, Y. Ji, "Axis-Exchanged Compensation and Gait Parameters Analysis for High Accuracy Indoor Pedestrian Dead Reckoning", Journal of Sensors, vol. 501, 2015.

[87]. Z. Tian, Y. Zhang, M. Zhou, Y. Liu, "Pedestrian dead reckoning for MARG navigation using a smartphone", EURASIP Journal on Advances in Signal Processing, vol. 2014, no. 1, pp. 1-9, 2014.

[88]. J. F. Li, Q. H. Wang, X. M. Liu, M. Y. Zhang, "An Autonomous Waist-Mounted Pedestrian Dead Reckoning System by Coupling Low-Cost MEMS Inertial Sensors and FPG Receiver for 3D Urban Navigation", Journal of Engineering Science and Technology, vol. 7, no. 1, 2014.