Reliable localization of pedestrians in a smart home using multi-sensor data fusion

Lina Achaji

Master of Research Report

Reliable localization of pedestrians in a smart home using multi-sensor data fusion

Submitted in fulfilment of the requirements for the Master degree of research in

Technology of Medical and Industrial Communicating Systems

From the:
Lebanese University – Faculty of engineering

Report presented by:

Lina Achaji

Defended on July 31th 2019 in front of the jury

Dr. Mohamad Khalil: Professor at the Lebanese University – Director of the Azm Center for Research in Biotechnology.
Dr. Jihan Khoder: Professor at the Lebanese University, Faculty of engineering.

Supervisors

Dr. Mohamad Daher: Associate Professor at the Lebanese University, Faculty of Technology.
Dr. Maan El Badaoui El Najjar: Professor at the University of Lille.
Dr. François Charpillet: Research Director INRIA, Nancy.

People should pursue what they're passionate about.
That will make them happier than pretty much anything else.

Elon Musk

Contents

1 Introduction **1**
- 1.1 AAL Systems . . . 1
- 1.2 Positioning and tracking approaches . . . 2
- 1.3 Previous Work . . . 3
  - 1.3.1 Existing Work . . . 3
  - 1.3.2 Work Continuity . . . 4
  - 1.3.3 Key contributions . . . 4

2 Sensors in use **6**
- 2.1 The INRIA Nancy - Grand Est SmartTiles . . . 6
  - 2.1.1 Architecture of the Inria SmartTiles prototype . . . 6
  - 2.1.2 Sensors embedded in the tile . . . 7
  - 2.1.3 Specification of the load sensor . . . 7
- 2.2 Inertial Measurement Units . . . 8
  - 2.2.1 Connected Watch – TicWatch: Wear OS by Google . . . 9
  - 2.2.2 TEA CAPTIV – T-Sens Motion (IMU) . . . 10

3 Sensor Fusion **11**
- 3.1 Definition and Motivation . . . 11
- 3.2 Kalman Filter . . . 12
  - 3.2.1 Kalman filter algorithm for a simplistic localization scenario . . . 13
  - 3.2.2 Equations Generalization . . . 14
  - 3.2.3 KF Limitations . . . 15
- 3.3 Extended Kalman Filter . . . 16
- 3.4 Informational Form (IF) of Kalman Filter . . . 16
  - 3.4.1 Informational Filter (IF) advantages . . . 17

4 DataSet Gathering **18**
- 4.1 Smart-Tiles raw Data . . . 19
  - 4.1.1 Load sensors Data Processing . . . 19
  - 4.1.2 Tiles Selection . . . 19
- 4.2 IMU Data from the Connected Watch . . . 21
  - 4.2.1 Android API . . . 22
  - 4.2.2 Time Synchronization . . . 22
- 4.3 T-Sens Motion (IMU) Data . . . 22
- 4.4 Sensors Location . . . 23

5 Proposed Models **24**
5.1 Bio-mechanical Transformation . . . 24
5.2 Work Strategy . . . 25
5.3 Prediction model - The velocity model . . . 25
5.4 Observation Model – Correction step . . . 27
5.4.1 W estimated Matrix . . . 27
5.4.2 W measured calculation . . . 28
5.4.3 Observation Matrix . . . 28

6 Multi-person localization **29**
6.1 Overall Procedure . . . 29
6.2 Gating . . . 30
6.3 Data Association . . . 31
6.4 Nearest Neighbor Standard Filter (NNSF) . . . 31
6.5 Global Nearest Neighbor GNN . . . 32
6.6 Informational Formalism for multi-tracking . . . 33

7 Results and Discussion **35**
7.1 Data Processing from tiles . . . 35
7.2 Fusion of the data . . . 36
7.2.1 Multi-person tracking fusion results . . . 39
7.3 Ground Truth - accuracy evaluation . . . 40
7.3.1 CoP computation . . . 40
7.3.2 Ground truth results . . . 41

Conclusion **42**

Appendices **43**

A Mathematical Derivation for Velocity Model **44**

B Load Pressure $\hat{W}$ Calculation **46**

References **49**

List of Figures

1.1 Indoor positioning technologies categorized by Mautz [3] 3

2.1 A 3D model of the "Intelligent Apartment" prototype located at Inria Nancy 6
2.2 Tiles distribution inside the smart apartment. 7
2.3 Image of the SparkFun SEN-10245 load sensor 8
2.4 The SparkFun SEN-10245 load sensor specifications 8
2.5 Inertial Measurement Unit 8
2.6 TicWatch E2 9
2.8 TEA CAPTIV T-REC 10
2.7 T-sens Motion IMU 10

3.1 Block diagram of sensor fusion 11
3.4 Kalman Filter Block Diagram 15

4.1 Experiment of two persons walking in the apartment 18
4.2 Snapshot from tiles Data 19
4.3 Load Distribution Before Normalization 20
4.4 Load Distribution after Normalization 20
4.5 Weight distribution tracing 20
4.6 Barycentre Position using a different color for each tile 20
4.7 Dead-reckoning, where the accelerometer measurements (external specific force) and the gyroscope measurements (angular velocity) are integrated to position and orientation. 21
4.8 Integrated orientation for the position in x- (blue), y- (green) and z- direction (red) . 21
4.9 Integrated position for rotation around the x(blue), the y(green) and the z(red) -axis 21
4.10 T-sens Data 22
4.11 Wearable Sensors Position 23

5.1 Work strategy 25

6.1 Problematic of the multi-person association. 30
6.2 Gating illustration 30
6.3 GNN Vs. NNSF 32
6.4 Error Function with reference to X and Y 34

7.1 Walking in straight line and returning back. 35
7.2 Walking in straight line and returning back after cleaning the data. 36
7.3 Proposed trajectory. 36
7.4 Tiles observations. 36
7.5 Fusion result - Approach 1. 37

7.6 Comparison between W_3 and $\hat{W}_3$. 37
7.7 Fusion results using approach 2. 38
7.8 θ evolution. 38
7.9 Theoretical trajectory for person one and two. 39
7.10 Result for multi-person fusion 39
7.11 Contact forces and moments acting on the sole. 40
7.12 CoP plot from the tiles, the force plates, and the fusion result. 41

A.1 Motion carried out by a robot moving with constant velocities v and w. 44

B.1 A person having a weight of W and a position coordinates (x_l,y_l) in a tile plan of dimension L. 46

List of Tables

3.1 KF Versus IF 17

4.1 Data Format for each Tile 19

List of Symbols

The next list describes several symbols that will be later used within the body of the report

AAL	Ambient Assisted Living
BDS	BeiDou Navigation Satellite System
COM	Center of Mass
CoP	Center of Pressure
EKF	Extended Kalman Filter
GALILEO	Galileo Positioning System
GLONASS	Global Navigation Satellite System
GNN	Global Nearest Neighbor Filter
GPS	Global Positioning System
IF	Informational Kalman Filter
IMU	Inertial Measurement Unit
KF	Kalman Filter
MEMS	Micro Electro-mechanical System
NNSF	Nearest Neighbor Standard Filter
RFID	Radio Frequency Identification
RMSE	Root Mean Square Error
UWB	Ultra Wideband

Abstract

The integration of sensors in future intelligent habitats as well as on people with the objective of helping them in their daily tasks is a concept that has become common.

This concept is based on the deployment of sensor networks in the environment (floor tiles, cameras, RFID...) that became active and communicating, in addition to wearable sensors carried by people (inertial unit, Gyroscope ...).

The work carries out the global task of assistance to the person which breaks down into several functions, amongst them are: the sure and reliable localization of pedestrians and the tracking of movements of static and dynamic objects. For the methodological part, it primarily consists of developing robust fusion algorithms for the localization of one or more persons. However, recognition generally fails when several people stand or walk together, preventing successful tracking. This work provides a tracking and association technique which uses the previous state information to associate between tracks. Secondly, a fault tolerance part, consists of integrating a layer of diagnosis for detection and identification of sensor faults to ensure a safe and reliable system using informational based approaches (using KL divergence for example for information measures).

From the standpoint application, a part of this project is dedicated to data acquisition campaigns on the smart apartment platform of INRIA Nancy - Grand Est. Data fusion algorithms will be tested and validated with real data.

This Project is carried out within the framework of a collaboration between the Lebanese University, the laboratory CRIStAL and INRIA Nancy Grand-Est.

Keywords: Sensor Fusion, Informational Filter, Multi-person Tracking, Sensing Floors, Inertial Sensors ...

Resumé

L'intégration de capteurs communicants (sols tactiles, caméras, RFID, capteurs domotiques...) devraient se généraliser dans les habitats du futur avec pour objectif d'aider les occupants dans leurs tâches quotidiennes. Certains capteurs pourront également être intégrés dans les vêtements des utilisateurs ou simplement portés (montre ou ceinture connectées, ...), capteurs dans lesquels on pourra intégrer des centrales inertielles ou autres gyroscopes.

Le travail proposé dans ce mémoire effectue une tâche d'assistance à la personne qui se décompose en plusieurs fonctions telles que : la localisation sûre et fiable de personnes et le suivi des mouvements d'objets statiques et dynamiques. Pour la partie méthodologique, elle consiste tout d'abord à développer des algorithmes de fusion pour la localisation d'une ou plusieurs personnes. Cependant, le suivi est particuliérement difficile et il échoue généralement lorsque plusieurs personnes se tiennent ou marchent à proximité. Ce travail présente une technique de suivi qui utilise les états précédents pour associer les observations aux différentes pistes posiibles. Dans un second temps nous nous sommes intéressés à la tolérance aux fautes. Pour cela nous avons intégré une couche de diagnostic pour la détection et l'identification des défauts capteurs. Cela permet de garantir un fonctionnement sûr et fiable. Le cadre théorique choisi est celui des filtres informationnels (en utilisant la divergence KL, par exemple pour les mesures d'information).

Une partie importante de ce projet a été dédiée à l'acquisition des données capteurs sur la plate-forme Habitat Intelligent de l'INRIA Nancy - Grand Est. Les algorithmes de fusion de données ont donc été testés et validés avec des données réelles.

Ce projet est réalisé dans le cadre d'une collaboration entre l' Université libanaise, les laboratoires CRIStAL et le Loria et l'INRIA Nancy Grand-Est.

Mots-clés: Fusion multi-capteurs, Filtre informationnel, Suivi multi-personnes, Capteurs de force, Capteurs inertiels ...

Acknowledgements

"It is not possible to prepare a project without the assistance & encouragement of other people. This one is certainly no exception."

Every work accomplished is a pleasure- a sense of satisfaction. However, a number of people always motivate, criticize and appreciate a work with their objective, ideas and opinions.

First, I wish to warmly thank my supervisors, Dr. François Charpillet for hosting me to work within INRIA Nancy – Larsen Team, and to provide me with his valuable guidance, help, and encouragement, Dr. Maan Al-Badaoui El Najjar and Dr. Mohamad Daher for their helpful contribution, knowledge, and practical advices.

I am very grateful to Louisa Takorabet, Pauline Houlgatte, Melanie Jouaiti, Yassine El Khadiri, and all the members of Larsen Team, for their help during my internship and particularly during the data acquisition experiments.

I extend my gratitude to the Lebanese University – Faculty of engineering for giving me this opportunity.

I would like to thank my inspirational person, Sebastian Thrun, for giving me the motivation to go forward in my career, and providing the best helpful online courses.

I also acknowledge with a deep sense of reverence, my gratitude towards my parents , my intelligent brother, and my creative sister, to support me in whatever I do and loving me unconditionally.

At last but not least gratitude goes to my special best friend, Natasha El-Khatib, for believing in me, sharing knowledge with me, and supporting me in all decisions during my last five years.

Chapter 1

Introduction

1.1 AAL Systems

In recent years, there has been a rapid introduction of new assisted living technologies due to a rapidly aging society. In fact, it is estimated that in 2060, the elderly are expected to account for 29.5 % of the total population in the EU-27. Altogether, the population of elderly people will almost double from 87.5 million in 2010 to 152.6 million in 2060 [1], while in the USA the population is also expected to become much older. By 2030, more than 20 % of U.S. residents are projected to be aged 65 and over, compared with 13 % in 2010 and 9.8 percent in 1970. [2]. In addition, people over 85 years usually require continuous monitoring. Therefore, taking care of elderly people has become a challenging and very important issue.

Ambient assisted living (AAL) can be defined as "the use of information and communication technologies (ICT) in a person's daily living and working environment to enable them to stay active longer, remain socially connected and live independently into old age" [1].
This field of research is growing rapidly. Enormous projects have been launched conducive to the establishment of novel AAL systems, aiming to grant several functionalities to the target users through taking advantage of new developments in sensor technology, reducing sizes and costs, and increasing processing power in computing devices.

The AAL research community borrows from a more mature field of research using a variety of sensors for activity recognition and behavior understanding, sensing directly via wearable sensors or indirectly through environmental sensors and analyzing the streaming data to infer something about the physical or cognitive status of the person observed.
Wearable sensors are employed in activity recognition, Inertial measurement units (IMUs) are the most frequently encountered. Due to the integration in most smartphones, they have been opportunistically used for activity recognition such as the sitting, walking, and running activities.

[1] goto *www.aal-europe.eu*

It's often necessary to locate an object or a person. The precision of the required localization varies from coarse (room level) to a more precise location within a room. Coarse location is provided by sensors that track motion (for single or low occupancy environments) or via radio-based technology.
More precise indoor localization and tracking can be achieved by using other types of sensors, e.g. sensing floor sensors, which that can serve to locate the person or multiple persons within an apartment, therefore we will talk about later it in the next chapters.

1.2 Positioning and tracking approaches

Positioning involves accurate localization of elderly people, including in both indoor and outdoor locations. There are several well established and widely used navigation systems for outdoor positioning, such as the Global Positioning System (GPS), the Global Navigation Satellite System (GLONASS), Galileo Positioning System (GALILEO) and BeiDou Navigation Satellite System (BDS). All of these are satellite-based systems for which ground-based sensors rely on signals from at least four satellites to estimate user coordinates. These technologies are currently accurate to approximately several meters for outdoor scenarios; however, they cannot be used to determine precise indoor locations because of the significant attenuation of satellite signals in buildings.

The current indoor positioning of satellite-based systems present a highly unacceptable error margin. Hence, the existing satellite-based positioning technologies can meet the demands of elderly care only for outdoor scenarios.

For elderly care scenarios, precise indoor positioning should work continuously in real-time. In recent decades, numerous indoor positioning approaches, such as Bluetooth, WiFi/WLAN, radio frequency identification (RFID), ultra-wideband (UWB), have been developed; However, these vary greatly in terms of their resolution, coverage, precision, technology, scalability, robustness, and security.

Considering the special demands of elderly care, solutions can be selected or developed by using existing technologies to address the problems. For example, through fusing two or more types of the existing technologies (see Fig. 1.1) using proper algorithm, the performance of the positioning system will be improved to a certain extent [4].

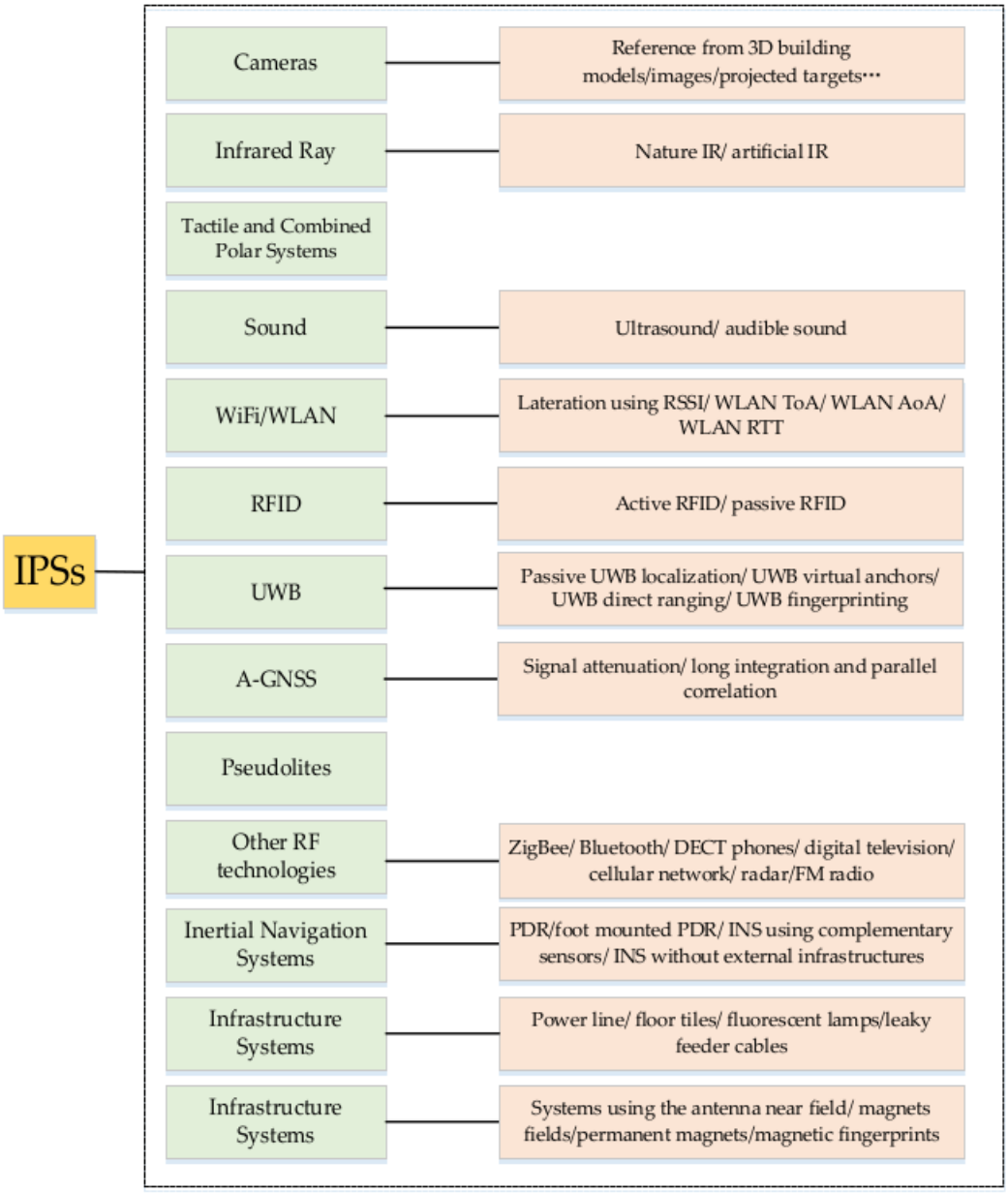


Figure 1.1: Indoor positioning technologies categorized by Mautz [3]

1.3 Previous Work

1.3.1 Existing Work

Several computing techniques and sensor technologies have been proposed in the past two decades to provide indoor localization systems for personal in-home staying. This field is quite challenging due to some faulty sensor measurements as well as people's random movements. Several projects

are developed around the world on this topic using multisource data fusion. In addition, there are many survey papers in the literature that present the latest indoor localization systems. In this section, we present some of papers working on the indoor localization aspect using different approaches.

In 2005, Nishkam Ravi and Liviu Iftode proposed a method for Indoor Localization Using Camera Phones where the location of the user is determined by comparing the received images with images stored in a database [5]. Fuqiang Gu, Milad Ramezani proposed also a method for fast and Reliable WiFi Fingerprint Collection for Indoor Localization [6]. Qigao Fan and Hai Zhang proposed an Improved Pedestrian Dead Reckoning Based on a Robust Adaptive Kalman Filter for Indoor Inertial Location System [7].

1.3.2 Work Continuity

Several works have been previously developed to localise, identify, and track: people, robots, and objects inside the INRIA smart apartment using the sensing floor prototype developed at Inria Nancy. This work represents a continuity of the previous developed work.

In 2016, Maxime Rio, Francis Colas, and Mihai Andries proposed a method for a probabilistic sensor data processing for robot localization on load-sensing floors [8]. It consists to track even the lighter robots with less than 4 cm of error. However, large-scale deployments necessitate cheap sensors which do not necessarily offer the same precision. With more noisy sensors, lighter robots might be difficult to track and precisely localize.

For this reason, in 2017, Mohamad Daher proposed a personal Localization System Based on Informational Formalism using data fusion between a simulated inertial measurement unit and a load pressure sensing floor [9]. In addition, a fault-tolerant fusion method is proposed using a purely informational formalism. The simulated inertial unit data are calculated using a camera-based sensing approach. By means of the mass center coordinates extracted from a simple RGB-D camera. Such proposition has the disadvantage that the privacy of the users can not be insured, and the integration of camera-based architecture inside large indoor places is not very easy to maintain.

1.3.3 Key contributions

This internship provides several contributions, that consist of innovative approaches for fusing load-sensing floor with multiple inertial measurement units (IMU) to localize one or more people walking inside the apartment . More precisely, the internship provides:

- Real Dataset gathering using data from the smart tiles, 6-axis inertial unit integrated inside a smart watch, and T-sens motion capture unit.
- Developing a robust prediction and observation model to fuse the multiple sensors using an informational formalism.
- Applying Bio-mechanical transformation using the "Inverted Pendulum model of human gait".
- Developing association and identification techniques to localize and track multiple pedestrians walking in the same time inside the apartment.
- Proof of concept and work validity using Force Plates for Ground Truth evaluation.

Chapter 2

Sensors in use

2.1 The INRIA Nancy - Grand Est SmartTiles

The Inria SmartTiles is a sensing floor prototype, conceived as a part of a bigger study of the habitat of the future (see Fig. 2.1). The floor is integrated into a real size mock-up of smart apartment, where spatially localised pressure sensing, memory, and computing power, are provided.

2.1.1 Architecture of the Inria SmartTiles prototype

The Inria SmartTiles prototype is a tiled sensing floor. The tiles are rigid, and have a size of 0.6 m by 0.6 m. Each tile can sense pressure, as well as memorise, compute and transmit data. Besides being equipped with a set of sensors, each tile also has an on-board processing unit, as well as a wireless and wired connection, which provides electric power.

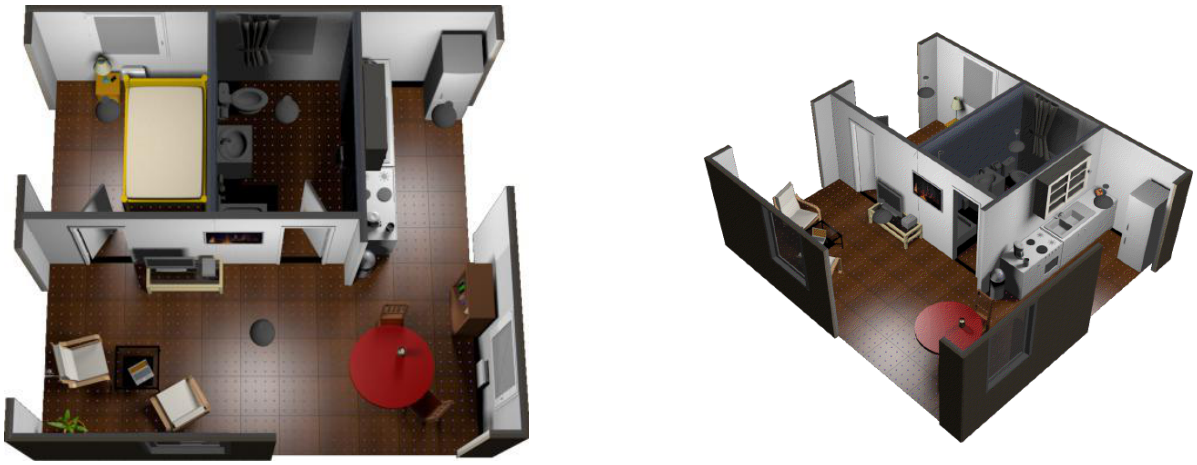


Figure 2.1: A 3D model of the ”Intelligent Apartment” prototype located at Inria Nancy

There exists a total of 104 tiles inside the smart appartment numerated as shown in Fig. 2.2.

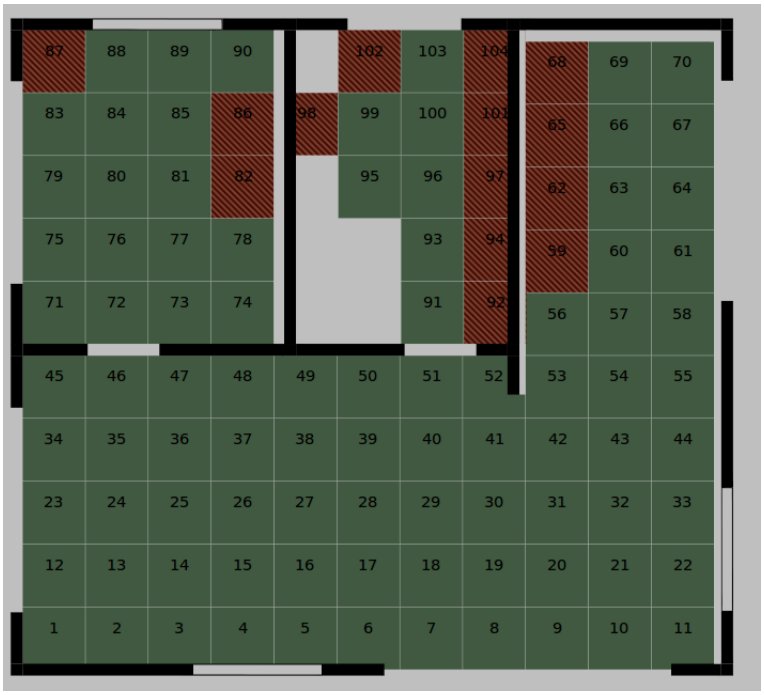


Figure 2.2: Tiles distribution inside the smart apartment.

2.1.2 Sensors embedded in the tile

Each tile is equipped with:

- 4 pressure sensors located at the corners of the tile.
- 1 accelerometer in the center of the tile.
- 1 magnetometer.

The pressure sensors employed (strain gauge load cells) measure the load forces exerted on the floor. The embedded accelerometers detect shocks that can be caused by objects or humans falling on the ground. The magnetometers serve to detect metallic masses located on the tiles, such as robots. Each tile also has 16 light-emitting diodes that provide visual feedback. The sensors are queried periodically for measurement data, with a sampling frequency around 50 Hz.

2.1.3 Specification of the load sensor

The Inria SmartTiles employ load sensors of the brand SparkFun SEN-10245, illustrated in (Fig. 2.3). The specification of this sensor is shown in (Fig. 2.4). The load sensors of the tile are

Figure 2.3: Image of the SparkFun SEN-10245 load sensor

Capacity	kg	40-50
Comprehensive Error	mv/v	0.05
Output Sensitivity	mv/v	1.0±0.1
Nonlinearity	%FS	0.03
Repeatability	%FS	0.03
Hysteresis	%FS	0.03
Creep	(3min)%FS	0.03
Zero Drift	(1min)%FS	0.03
Temp. Effect on Zero	%FS/10℃	1
Temp. Effect on Output	%FS/10℃	0.05
Zero Output	mV/V	±0.1
Input Resistance	Ω	1000±20
Output Resistance	Ω	1000±20
Insulation Resistance	MΩ	≥5000
Excitation Voltage	V	≤10
Operation Temp. Range	℃	0--+50
Overload Capacity	%FS	150

Figure 2.4: The SparkFun SEN-10245 load sensor specifications

subjected to noise. The measurement error of individual sensors is comprised between ±1.25 Kg, with the combined measurement error typically oscillating between ±2 Kg. This noise level will be used after to calculate the threshold for tiles activation.

2.2 Inertial Measurement Units

The term inertial sensor is used to denote the combination of a three-axis accelerometer and a three-axis gyroscope. Devices containing these sensors are commonly referred to as inertial measurement units (IMUs).

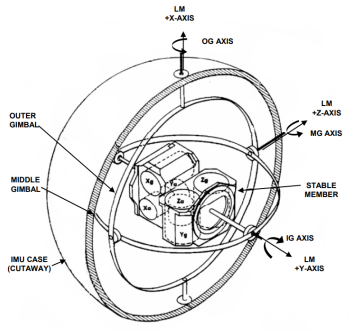


Figure 2.5: Inertial Measurement Unit

A gyroscope measures the sensor's angular velocity, i.e. the rate of change of the sensor's orientation.

An accelerometer measures the external specific force acting on the sensor. The specific force consists of both the sensor's acceleration and the earth's gravity. Nowadays, many gyroscopes and accelerometers are based on microelectromechanical system (MEMS) technology.

MEMS components are small, light, inexpensive, have low power consumption and short start-up times. Their accuracy has significantly increased over the years.

2.2.1 Connected Watch – TicWatch: Wear OS by Google

One of the most popular wearable devices used in indoor localization is the smart-watch. In part, its popularity comes down to the fact that it can be seen as an extension of a smart-phone, which, at the same time, its main advantage is to be a regular device, that it looks like a common watch easy to wear and use. Another advantage is that a smart-watch is always attached to the user, roughly speaking, one is less likely to forget a smart-watch on top of the beside table that a smart-phone.

Figure 2.6: TicWatch E2

The watch used is the TichWatch E2[1] (Fig. 2.6). This smart-watch runs Android Wear OS[2], it's a version of Google's Android operating system designed for smart watches. These watches have built-in sensors that measure motion, orientation, and various environmental conditions, capable of providing raw data with high precision and accuracy. Embodies a WiFi chip that is accessible through Android API, GPS, **3-axis accelerometer, 3-axis gyroscope and linear acceleration measurement with a frequency rate that can reach 100 Hz as maximum.** Connectivity is supported through WiFi and Bluetooth. The resolution of its 1.39" 400 x 400 pixels, battery duration around 48h, and it costs less than 150 $ at the time of this writing.

[1] see *https://www.mobvoi.com* for more informations.
[2] *https://wearos.google.com*

Figure 2.8: TEA CAPTIV T-REC

2.2.2 TEA CAPTIV – T-Sens Motion (IMU)

T-Sens Motion (Fig. 2.8) combines accelerometers, gyroscopes, magnetometers using powerful fusion algorithms. The measurements are extremely robust over time against vibrations, variations in the magnetic field and disturbed environments. The T-sens motion IMUs are wireless sensors with easy and unobtrusive fixation system. They can measure the orientation where the raw data is available as rotation quaternion. The Sampling rate is 64 Hz, battery recording time can reach 4h and dimensions of: 60 mm x 35 mm x 19 mm.

Figure 2.7: T-sens Motion IMU

The Key importance of T-sens sensor is the easy ability of synchronization between multiple T-sens sensors. T-Rec (Fig. 2.7) allows acquisition and synchronization of measurements from 1 to 16 wireless sensors.

Chapter 3

Sensor Fusion

3.1 Definition and Motivation

Sensor Fusion is the combining of sensory data or data derived from sensory data such that the resulting information is in some sense better and has less uncertainty than would be possible when these sources were used individually [10].

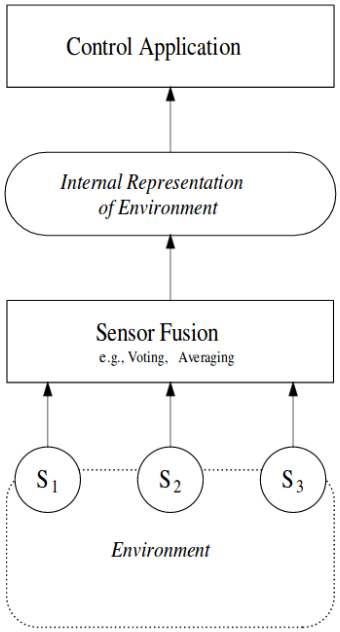


Figure 3.1: Block diagram of sensor fusion

Systems that employ sensor fusion methods expect a number of benefits over single sensor systems. A physical sensor measurement generally suffers from the following problems:

- **Sensor Deprivation:** The breakdown of a sensor element causes a loss of perception on the desired object.
- **Limited spatial coverage:** Usually an individual sensor only covers a restricted region.
- **Limited temporal coverage:** Some sensors need a particular set-up time to perform and to transmit a measurement, thus limiting the maximum frequency of measurements.
- **Imprecision:** Measurements from individual sensors are limited to the precision of the employed sensing element.
- **Uncertainty:** Uncertainty, in contrast to imprecision, depends on the object being observed rather than the observing device. Uncertainty arises when features are missing (e. g., occlusions), when the sensor cannot measure all relevant attributes of the percept, or when the observation is ambiguous [11]. A single sensor system is unable to reduce uncertainty in its perception because of its limited view of the object [12].

One solution to the listed problems is to use sensor fusion system which, is robust behavior against sensor deprivation can be achieved by using sensors with overlapping views of the desired object. This works with a set of sensors of the same type as well as with a suite of heterogeneous sensors (Fig. 3.1).

3.2 Kalman Filter

Kalman filtering is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe.
The Kalman filter remains an important tool to **fuse measurements from several sensors** to estimate in real-time the state of a robotic system.

Kalman filters are ideal for systems which are continuously changing. They have the advantage that they are light on memory (they don't need to keep any history other than the previous state), and they are very fast, making them well suited for real time problems and embedded systems [13].

A stochastic time-variant linear system is described by the difference equation and the observation model:

$$x_k = x_{k-1} + A_{k-1}u_{k-1} + w_{k-1}$$

$$z_k = H_k x_k + v_k$$

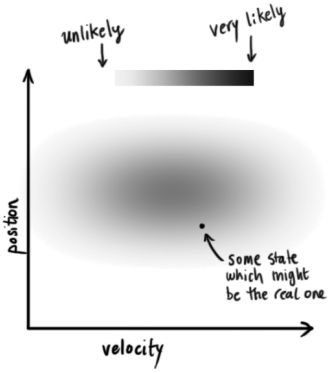
Where the control input u_k is a known nonrandom vector. The initial state x_0 is a random vector with known mean $\mu_0 = E[x_0]$ and covariance $P_0 = E[(x_0 - \mu_0)(x_0 - \mu_0)^T]$. In the following we assume that the random vector w_k captures uncertainties in the model and v_k denotes the measurement noise.

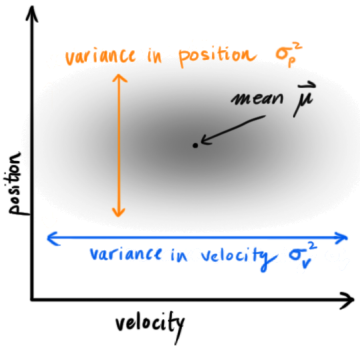
3.2.1 Kalman filter algorithm for a simplistic localization scenario

Suppose a robot moves along the horizontal axis with position p and velocity v. The state x is represented by:

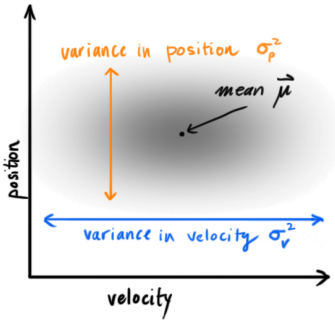
$$x = \begin{bmatrix} p \\ v \end{bmatrix}$$

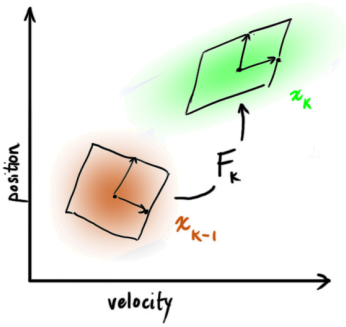
We do not know what is the actual position and velocity, there is a whole range of possible combinations that might be true, but some of them are more likely than others. The Kalman filter assumes that variables are random and Gaussian distributed. Each one has a mean value μ, which is the center of the random distribution, and a variance σ^2 which represents the uncertainty.



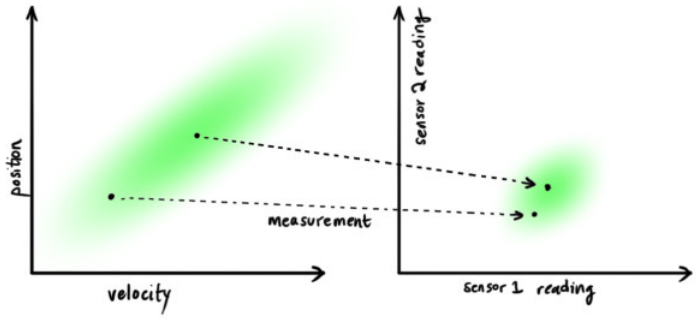


Next, we need a way to look at the current state (at time k-1) and predict the next state at time k. The prediction function works on the state variables and gives us a new distribution, We can represent this prediction step with a matrix F_k .

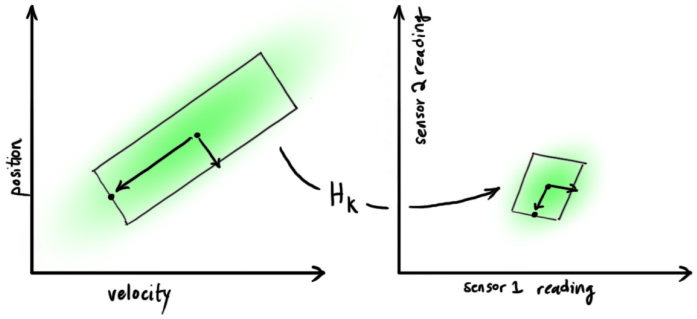




We might have several sensors which, give information about the state of our system. The sensors operate on a state and produce a set of measurements.



The units and scale of the reading might not be the same as the units and scale of the state we're keeping track of. We have to model the sensors with a matrix, H_k.



3.2.2 Equations Generalization

Dimension and description of variables:

X_k	$n \times 1$	State vector
F_k	$n \times n$	State Matrix
B_k	$n \times l$	Input Matrix
u_k	$l \times 1$	Input Vector
w_k	$n \times 1$	Process noise vector
Q_k	$m \times n$	Process noise Matrix
Z_k	$m \times 1$	Observation vector
H_k	$m \times n$	Observation Matrix
v_k	$m \times 1$	Measurement noise vector
v_k	$m \times 1$	Measurement noise vector
R_k	$m \times m$	Measurement noise covariance matrix

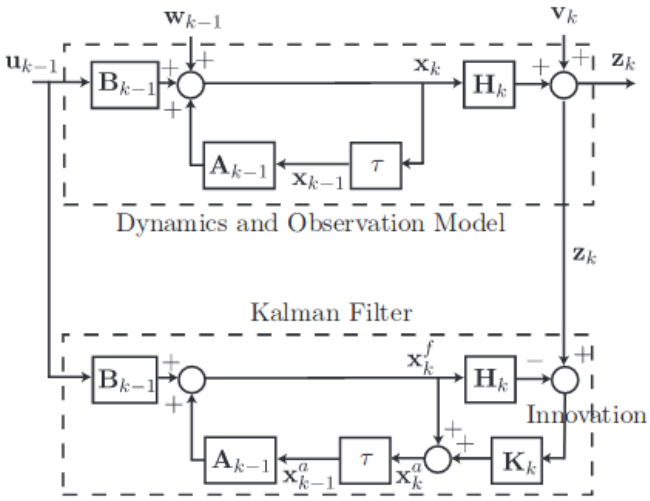


Figure 3.4: Kalman Filter Block Diagram

The sequential algorithm of KF (Fig. 3.4) is composed of two steps: **Prediction step** given by the total probability theorem and **Correction step** calculated by the Bayes Rule.

Prediction Step:

The estimation of the state $X_{k/k-1}$ and the covariance matrix $P_{k/k-1}$ of the system is realized as follows:

$$X_{k/k-1} = F_{k-1}X_{k-1/k-1} + B_{k-1}u_{k-1} \tag{3.1}$$

$$P_{k/k-1} = F_{k-1}P_{k-1/k-1}{F_{k-1}}^T + Q_{k-1} \tag{3.2}$$

Correction Step:

$$X_{k/k} = X_{k/k-1} + W_k(Z_k - H_kX_{k/k-1}) \tag{3.3}$$

$$P_{k/k} = P_{k/k-1} - W_kS_k{W_k}^T = (I - W_kH_k)P_{k/k-1} \tag{3.4}$$

$$W_k = P_{k/k-1}{H_k}^TS_{k-1} \tag{3.5}$$

$$S_k = H_kP_{k/k-1}{H_k}^T + R_k \tag{3.6}$$

3.2.3 KF Limitations

The KF is a linear filter that can be applied to a gaussian linear system. Unfortunately, linear systems do not exist in real life. Eventually we run across a system that does not behave linearly even over a small range of operation, and the standard KF no longer gives good results. In this case, we need to study a variant of kalman filter called Extended Kalman Filter.

3.3 Extended Kalman Filter

The extended Kalman filter (EKF) is the nonlinear version of the Kalman filter which linearizes about an estimate of the current mean and covariance [14]. The state transition and observation models don't need to be linear functions of the state but may instead be differentiable functions.

Consider the system defined by the following non-linear equations:

$$X_{k+1} = f(X_k, u_k) + w_k \tag{3.7}$$

$$Z_k = h(X_k) + v_k \tag{3.8}$$

Prediction Step:
In order to apply the Kalman filter, a linearization around the estimated trajectory is necessary in order to obtain a linear model of the error.

$$X_{k/k} = f(X_{k-1/k-1}, u_{k-1}) \tag{3.9}$$

$$P_{k/k-1} = F_{k-1} P_{k-1/k-1} {F_{k-1}}^T + Q_{k-1} \tag{3.10}$$

where F_k represent the Jacobian Matrix of f (.): $F_{k-1} = \left.\frac{\partial f}{\partial X}\right|_{X=X_{k-1/k-1}}$

Correction Step:
In order to apply the Kalman filter, a linearization around the estimated trajectory is necessary in order to obtain a linear model of the error.

$$X_{k/k} = X_{k/k-1} + W_k(Z_k - h(X_{k/k-1})) \tag{3.11}$$

$$P_{k/k} = P_{k/k-1} - W_k[H_k P_{k/k-1} {H_k}^T + R_k]{W_k}^T \tag{3.12}$$

$$W_k = P_{k/k-1} {H_k}^T [H_k P_k/k - 1 {H_k}^T + R_k]^{-1} \tag{3.13}$$

where H_k represents the Jacobian Matrix of h (.): $H_k = \left.\frac{\partial h}{\partial X}\right|_{X=X_{k-1/k-1}}$

3.4 Informational Form (IF) of Kalman Filter

The informational filter uses the informational form of the state vector and the matrix of covariances named respectively informational vector (y) and matrix informational (Y). It's the same of Kalman Filter from a mathematical approach. It's composed also from two steps, the **prediction step** coming from model prediction and the **correction step** coming from measurement observation.

Prediction Step:

$$Y_{k/k-1} = [F_{k-1} Y_{k-1/k-1} {F_{k-1}}^T + Q_{k-1}]^{-1} \tag{3.14}$$

$$y_{k/k-1} = Y_{k/k-1}[F_{k-1} X_{k-1/k-1} + B_{k-1} u_{k-1}] \tag{3.15}$$

$$= Y_{k/k-1}[F_{k-1} {Y_{k-1/k-1}}^{-1} y_{k-1/k-1} + B_{k-1} u_{k-1}] \tag{3.16}$$

Correction Step:

$$I - W_k H_k = P_{k/k} P_{k/k-1} \tag{3.17}$$

IF update:

$$Y_{k/k} = Y_{k/k-1} + {H_k}^T R^{-1} H_k \tag{3.18}$$

$$I_k = {H_k}^T R_k^{-1} H_k \tag{3.19}$$

3.4.1 Informational Filter (IF) advantages

Table 3.1: KF Versus IF

KF	IF
$P \to \infty$ if there is total uncertainty (Better not working with ∞ matrices)	$P \to$ 0 if there is total uncertainty (0 is better for calculations)
Do an observation step for each new observation and calculate all the matrices (The observations are correlated with the previous states)	Linear combination of all the observations using simple summation (The observations are not correlated with the previous states)

Chapter 4

DataSet Gathering

One of the most time consuming task of this internship is the acquisition of real data coming from the tiles and IMU sensors. Many experiments were carried out inside the INRIA apartment, for one and two people walking together at the same time, in the form of a loop and crossing in the same tile, for a walk in a straight line and for a person moving his body in the area of a single tile. Every person was wearing a connected watch and the T-sens IMU.
The experiments have been recorded using the kinect cameras (see Fig. 4.1) as a proof of concept, but the data from kinects are not included in the fusion algorithm.

Figure 4.1: Experiment of two persons walking in the apartment

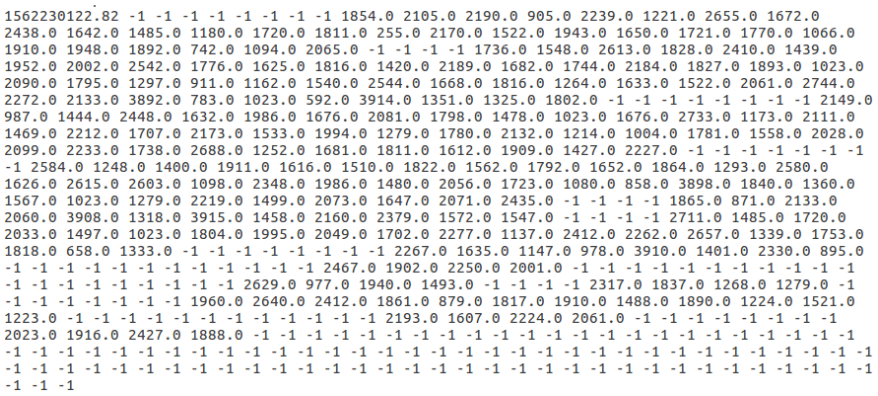


Figure 4.2: Snapshot from tiles Data

4.1 Smart-Tiles raw Data

The various nodes employed in the ambient intelligence tasks inside the smart apartment use Robot Operating System (ROS) for intercommunication. The format of the data sent by each tile is as follows:

Table 4.1: Data Format for each Tile

Sensor Type	**Sensor IP**	**msg ID**	**Timestamp**	**W1**	**W2**	**W3**	**W4**
Gauge	192.168.1.24	7090	1561995527	1639	1973	1690	2092

The final form of Data for all the tiles inside the apartment is given by: **timestamp 4x104 values** (see Fig. 4.2).

4.1.1 Load sensors Data Processing

The Data coming from the Rosbag File in (Fig. 4.2) is splitted, and indexed and ordered by the Timestamp value and Tile ID.
The (Fig. 4.3) is the distribution of Load Sensors value during the time of the experiment. As we can see the 4 Load sensors are not well calibrated at the beginning (W4 is always lower from the other load sensors even if we have a change in the values). A normalization step must be done before. All the sensors are maintained to zero mean value in order to calculate correctly the position of the person on each tile (Fig. 4.4).

4.1.2 Tiles Selection

During walking on floors, all the data given from the load sensors are registered at each time, so a selection method has to be applied to select the tile or the group of tiles on which a person or group

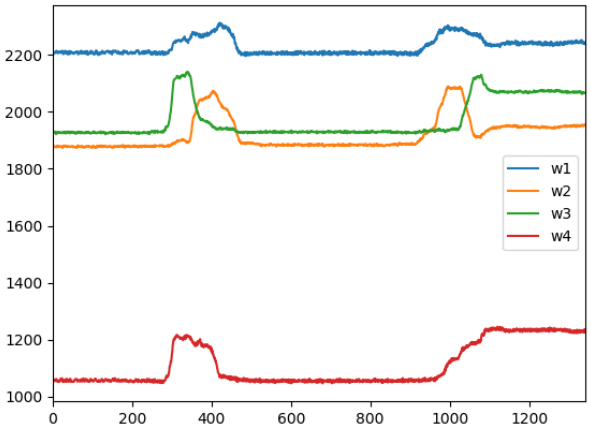


Figure 4.3: Load Distribution Before Normalization

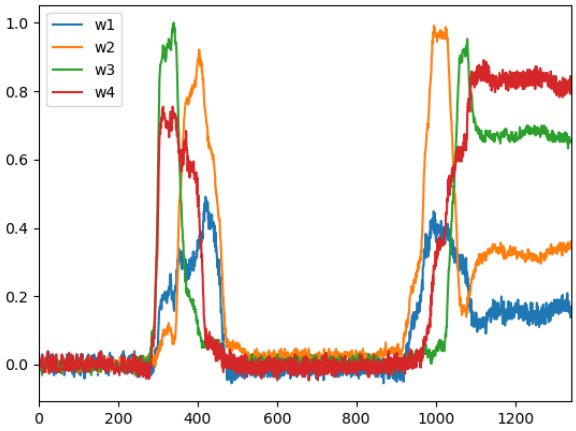


Figure 4.4: Load Distribution after Normalization

of persons are standing or walking on. The idea here is to apply a right thresholding technique to compare each time the value of the total weight on a tile (ΣW_i) with the calculated threshold. If the total applied weight is bigger from the threshold, so the tile is selected and enters in the observation vector later.

The Threshold can be deduced according to the noise value on the Tiles. An experiment is done before on the Tiles to calculate the noise distribution using various weights. The results show that the noise are linearly dependent with the weight applied on the tile (see Fig. 4.5) and that the (ΣW_i) is gaussian distributed (see Fig. 4.6).

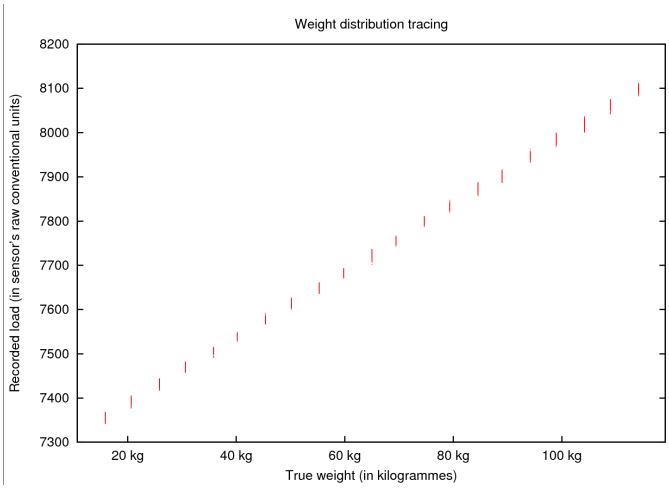


Figure 4.5: Weight distribution tracing

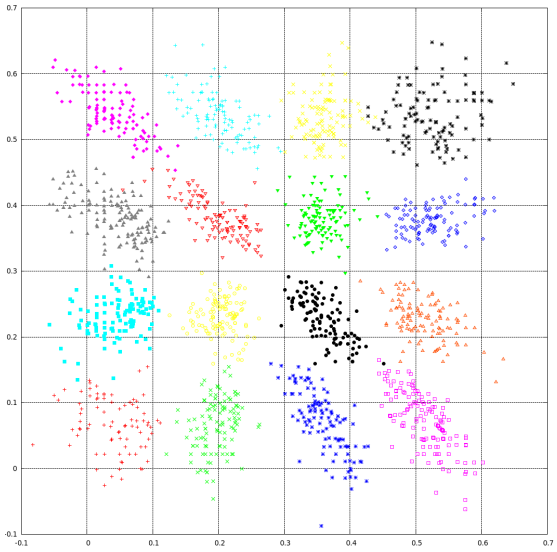


Figure 4.6: Barycentre Position using a different color for each tile

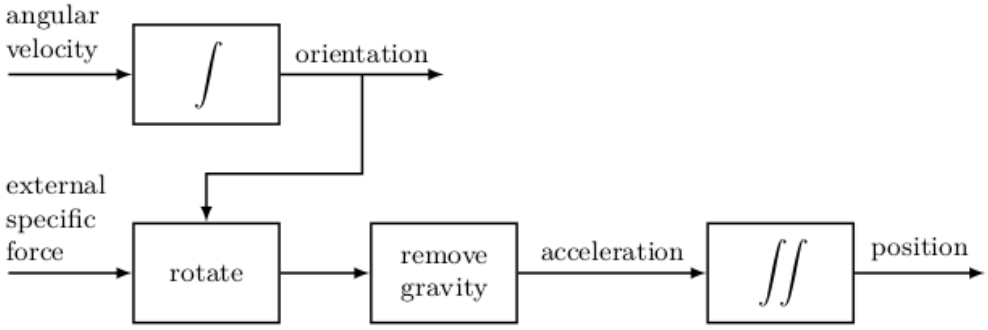


Figure 4.7: Dead-reckoning, where the accelerometer measurements (external specific force) and the gyroscope measurements (angular velocity) are integrated to position and orientation.

4.2 IMU Data from the Connected Watch

Inertial sensors are frequently used for navigation purposes where **the position and the orientation** of a device are of interest. Integration of the gyroscope measurements provides information about the orientation of the sensor. After subtraction of the earth's gravity, double integration of the accelerometer measurements provides information about the sensor's position.
To be able to subtract the earth's gravity, the orientation of the sensor needs to be known. Hence, estimation of the sensor's position and orientation are inherently linked when it comes to inertial sensors. The process of integrating the measurements from inertial sensors to obtain position and orientation information, often called dead-reckoning (Fig. 4.7). If the initial pose would be known, and if perfect models for the inertial sensor measurements would exist, the process illustrated in (Fig. 4.7) would lead to perfect pose estimates. In practice, the inertial measurements are noisy and biased. Because of this, the integration steps from angular velocity to rotation and from acceleration to position introduce integration drift. The drift is illustrated in (Fig. 4.8) and (Fig. 4.9) [17]. Because of this, inertial sensors need to be supplemented with other sensors and other models to obtain accurate position and orientation estimates.

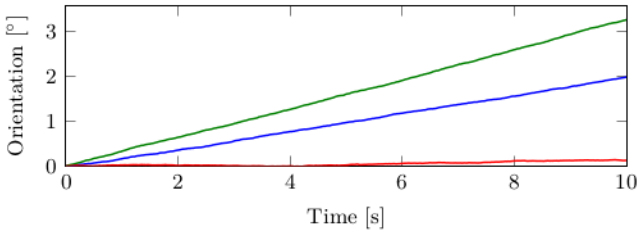


Figure 4.8: Integrated orientation for the position in x- (blue), y- (green) and z- direction (red)

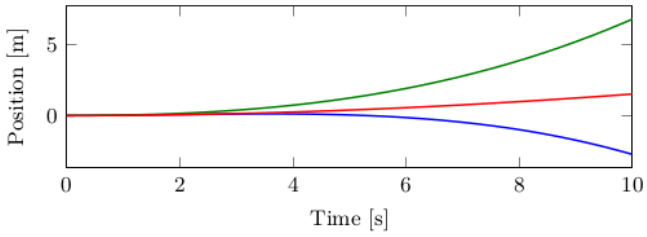


Figure 4.9: Integrated position for rotation around the x(blue), the y(green) and the z(red) -axis

4.2.1 Android API

An Android application has been developed to access the Sensor Manager built into the watch. The application writes, in the internal storage of the watch, the value of the linear acceleration and angular velocity on event change, with a maximum frequency of 100 Hz.

4.2.2 Time Synchronization

We can't use "currentTimeMillis" function or equivalent directly in the "onSensorChange" function because when we get there, the measure may have been accomplished long before. A proposed solution is to record the associated timestamp (event.timestamp), for each measure, which gives us the most reliable time. But the origin here is the origin of the watch's start time with a possible drift depending on the quality of the clock of the watch. To estimate the difference between this time and the true time, we have to calculate:
t_{now} = System.currentTimeMillis () t_o = SystemClock.elapsedRealtimeNanos()
$t_{now} - t_o$ (returned to the same units) will give us an estimate of the absolute start time;
So the absolute time is $t_{abs} = t_{now} - t_o +$ event.timestamp.

4.3 T-Sens Motion (IMU) Data

The T-sens IMU sensor is calibrated using the Captiv Software, and the data are recorded and saved as .csv file extension.

Nom : Mesure : Enregistrement / qx 127
Date de début : 01/07/2019 17:38:08,644
Fréquence : 64 Hz
Zéro : -0,30712890625
Unité : qx

Nom : Mesure : Enregistrement / qy 127
Date de début : 01/07/2019 17:38:08,644
Fréquence : 64 Hz
Zéro : -0,4765625
Unité : qy

Nom : Mesure : Enregistrement / qz 127
Date de début : 01/07/2019 17:38:08,644
Fréquence : 64 Hz
Zéro : 0,66162109375
Unité : qz

Nom : Mesure : Enregistrement / qw 127
Date de début : 01/07/2019 17:38:08,644
Fréquence : 64 Hz
Zéro : 0,48974609375
Unité : qw

Fréquence du fichier : 64 Hz

Date de début : 01/07/2019 17:38:08,644

Time;qx 127;qy 127;qz 127;qw 127;
0;0,30029296875;0,4892578125;-0,6630859375;-0,4794921875;
0,015625;0,30078125;0,48876953125;-0,66259765625;-0,48046875;
0,03125;0,30126953125;0,48834228515625;-0,662109375;-0,48138427734375;
0,046875;0,3017578125;0,48822021484375;-0,66156005859375;-0,48199462890625;
0,0625;0,30230712890625;0,48779296875;-0,66070556640625;-0,48291015625;

Figure 4.10: T-sens Data

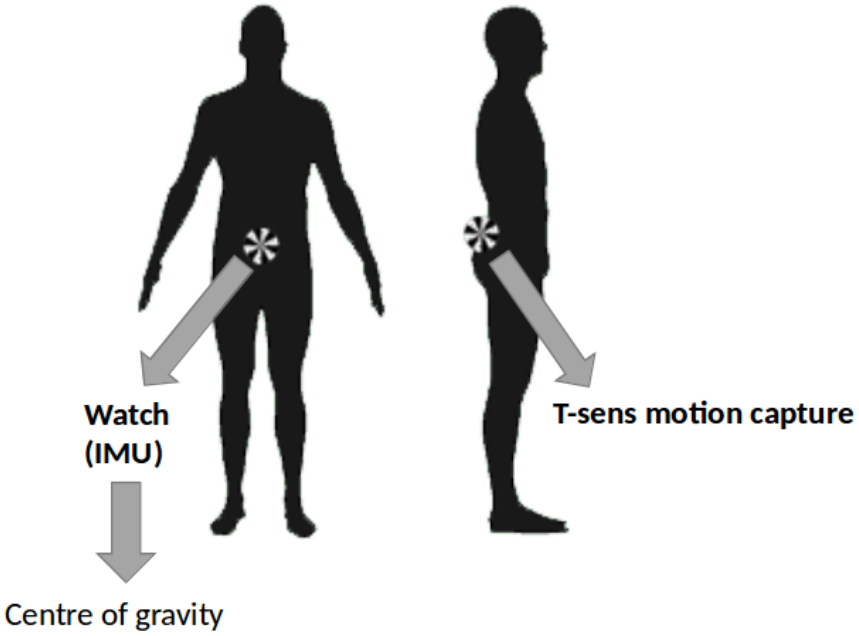


Figure 4.11: Wearable Sensors Position

A snapshot of the data is given in (Fig. 4.10), As we see The T-sens sensors return a quaternion vector $[qx; qy; qz; qw]$ every 0.015625 sec. The quaternion values are transformed to Rotation Matrix, giving us the 3 euler angles Roll, Pitch, and Yaw.

4.4 Sensors Location

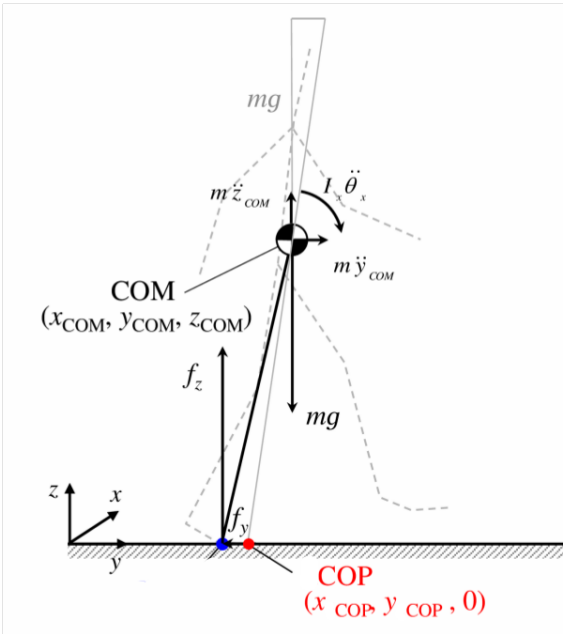
The watch and T-sens sensors are mounted on the belt (from the front and back side of the waist) as shown in (Fig. 4.11). So we can suppose that the IMU sensor is measuring the position of the center of gravity (center of mass). But for the fact that the Tiles give information about the position of the center of pressure, we must transform the position of the center of mass calculated from the IMU to the position of the projection of the center of pressure in the floor plan, and after that we can do the fusion between the 2 sensors.

Chapter 5

Proposed Models

5.1 Bio-mechanical Transformation

As we mentionned before (in section 4.4) a transformation have to be applied to transform the centre of mass(CoM) value to the projection of the centre of pressure(CoP). By definition, The field of pressure forces (normal to the sole) is equivalent to a single resultant force, exerted at the point where **the resultant moment is zero**, this point is termed CoP [21]. The CoP can be calculated based on the kinematics of the COM. **We assume an inverted pendulum model** with all mass at the COM location with a reaction force applied at the actual, global center of pressure of supporting foot (or feet when double support period) [16].



the (x,y) coordinate of which the CoP are described as:

$$x_{COP} = x_{COM} - \frac{\ddot{x}_{COM}}{\ddot{z}_{COM} + g} z_{COM} \tag{5.1}$$

$$y_{COP} = y_{COM} - \frac{\ddot{y}_{COM}}{\ddot{z}_{COM} + g} z_{COM} \tag{5.2}$$

5.2 Work Strategy

In this work, the data coming from the Accelerometer and Gyroscope of the watch will feed the prediction model, and the data coming from the tiles and the T-sens imu will be considered in the correction state to correct our predicted position (Fig. 5.1).

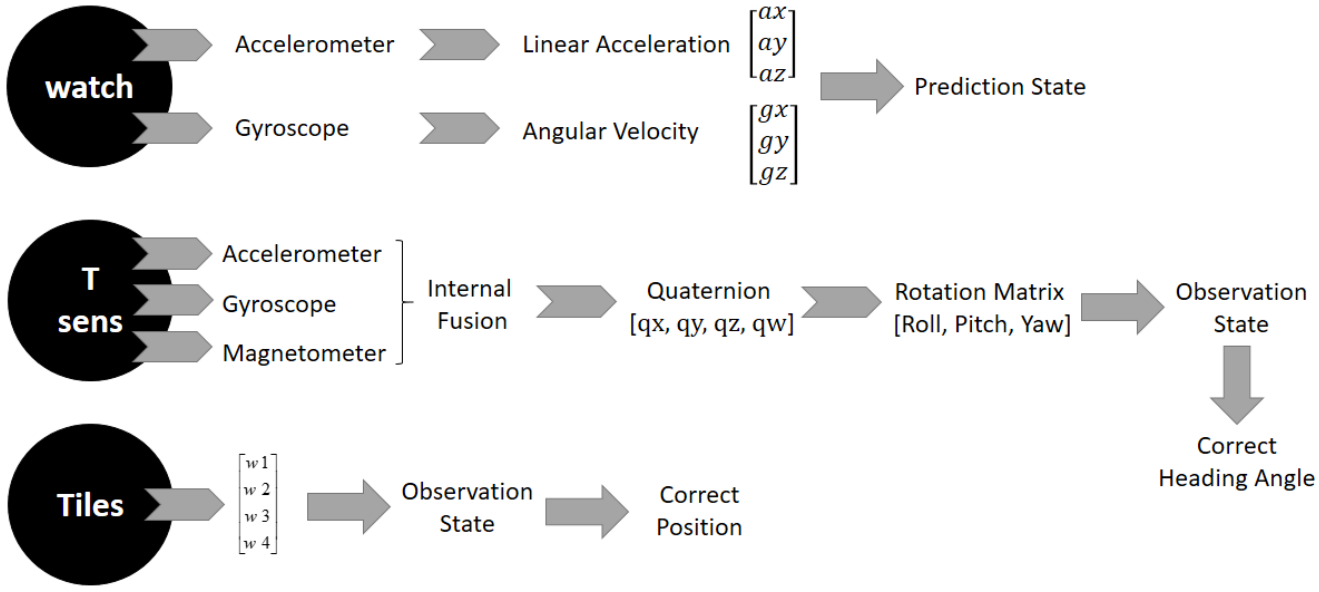


Figure 5.1: Work strategy

5.3 Prediction model - The velocity model

The velocity motion model assumes that we can control a robot through two velocities, rotational and translational velocities. This approach can be applied in the same manner for pedestrians if we consider that the linear and angular velocities are given (or calculated) from the IMU. We will denote the linear velocity at instant k by v_k , and the angular velocity by w_k [18]. Hence, we have: $u_k = \begin{bmatrix} v_k \\ w_k \end{bmatrix}$

As mentioned before, the linear velocity can be calculated as the first integral of the linear acceleration coming from the accelerometer:

$$v_{kx} = v_{kx0} + a_{kx}\Delta t \tag{5.3}$$

$$v_{ky} = v_{ky0} + a_{ky}\Delta t \tag{5.4}$$

$$v_k = \sqrt{{v_{kx}}^2 + {v_{ky}}^2} \tag{5.5}$$

The angular velocity w_k is obtained directly from the Gyroscope measurement. The velocity model equations, **when the change in angular rate is considered as part of the motion**, can be summarized by: (The mathematical derivation is given in Appendix A)

$$x_f = x_0 + \frac{v}{w}(\sin(\theta_0 + w\Delta t) - \sin(\theta_0)) \tag{5.6}$$

$$y_f = y_0 + \frac{v}{w}(\cos(\theta_0) - \cos(\theta_0 + w\Delta t)) \tag{5.7}$$

$$\theta_f = \theta_0 + w\Delta t \tag{5.8}$$

The state vector x_k of the person is considered to be the position (x, y) and the orientation (θ).

$$X_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T$$

The propagation model is given as follows: $X_{k+/k} = X_{k_k} + A_k u_k + w_k$
where $X_{k+1/k}$ is the estimate of X at instant $k+1$ given observations up to instant k, A_k is the input matrix, u_k is the input vector defined before, and w_k is the process state noise modeled as Gaussian white noise with zero mean and covariance Q_k.

$$X_{k+1/k} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \frac{v_k}{w_k}(\sin(\theta_k + w_k\Delta t) - \sin(\theta_k)) \\ \frac{v_k}{w_k}(\cos(\theta_k) - \cos(\theta_k + w_k\Delta t)) \\ w_k\Delta t \end{bmatrix} + w_k = f(X_k, u_k) + w_k$$

Since the model is nonlinear, the EIF is applied. Therefore, the Jacobian matrices $F_k = \left.\frac{\partial f}{\partial X}\right|_{X=X_{k/k}}$ and $B_k = \left.\frac{\partial f}{\partial u}\right|_{u_k}$ are computed.

$$F_k = \begin{bmatrix} 1 & 0 & \frac{v_k}{w_k}(\cos(\theta_k + w_k\Delta t) - \cos(\theta_k)) \\ 0 & 1 & \frac{v_k}{w_k}(\sin(\theta_k + w_k\Delta t) - \sin(\theta_k)) \\ 0 & 0 & \Delta t \end{bmatrix} \tag{5.9}$$

$$B_k = \begin{bmatrix} \frac{\sin(\theta_k + w_k\Delta t) - \sin(\theta_k)}{w_k} & \frac{v_k}{{w_k}^2}(\sin(\theta_k) - \sin(\theta_k + w_k\Delta t)) + \frac{v_k}{w_k}\Delta t\cos(\theta_k + w_k\Delta t) \\ \frac{\cos(\theta_k) - \cos(\theta_k + w_k\Delta t)}{w_k} & \frac{v_k}{{w_k}^2}(\cos(\theta_k + w_k\Delta t) - \cos(\theta_k)) + \frac{v_k}{w_k}\Delta t\sin(\theta_k + w_k\Delta t) \\ 0 & \Delta t \end{bmatrix} \tag{5.10}$$

The covariance matrix corresponding to this evolution model is the following:

$$P_{k+1/k} = F_k P_{k/k} {F_k}^T + B_k (Q_u)_k (B_k)^T + Q_k$$

Therefore, the informational matrix denoted $Y_{k+1/k}$ and the information vector denoted $y_{k+1/k}$ can be calculated as the following:

$$Y_{k+1/k} = {P^{-1}}_{k+1/k} \tag{5.11}$$

$$y_{k+1/k} = Y_{k+1/k} X_{k+1/k} \tag{5.12}$$

5.4 Observation Model – Correction step

The fusion of the load pressure sensors and T-sens observations with the data coming from IMU model is carried out using IF:

$$Y_{k/k} = Y_{k/k-1} + \sum_{i=1}^{N} I_i(k) \tag{5.13}$$

$$y_{k/k} = y_{k/k-1} + \sum_{i=1}^{N} i_i(k) \tag{5.14}$$

$$I_i(k) = {H_{i,k}}^T {R_i}^{-1}(k) H_{i,k} \tag{5.15}$$

$$i_i(k) = {H_{i,k}}^T {R_i}^{-1}(k)[(Z_{i,k} - \hat{Z}_{i,k}) + H_{i,k} X_{k/k-1}] \tag{5.16}$$

where $I_i(k)$ and $i_i(k)$ are the informational contributions associated with the measurement of the ith sensor, $H_{i,k}$ is the ith line of the matrix H_k, N is the number of sensors(the number of pressure sensors under the tile + the t-sens sensor). The noise associated with the sensors is assumed to be uncorrelated. And each noise is assumed to be gaussian white noise of zero mean and a covariance matrix R_i. Z matrix is composed of the pressure sensors received and the yaw angle from the t-sens sensor, $\hat{Z}$ is the estimated value of the pressure sensors and the heading angle coming from the prediction model.

$$Z = \begin{bmatrix} W & Yaw \end{bmatrix}^T = \begin{bmatrix} W_1 & W_2 & W_3 & W_4 & Yaw \end{bmatrix}^T$$

$$\hat{Z} = \begin{bmatrix} \hat{W} & \theta \end{bmatrix}^T = \begin{bmatrix} \hat{W}_1 & \hat{W}_2 & \hat{W}_3 & \hat{W}_4 & \theta \end{bmatrix}^T$$

5.4.1 W estimated Matrix

$\hat{W}$ is given by the following matrix 5.17 (The mathematical derivation is given in Appendex B), where: (x_L, y_L) are the local position of the center of pressure considering inside a specific tile

having identifier ($id = i$): $x_L = x - L[(i-1)\%n] = x - k1 \quad y_L = y - L[(i/n)] = y - k2$
where: (x, y) are the global coordinates of the person coming from the prediction state. L is the tile length , and n is the total number of tiles in a row.

$$\hat{W} = \begin{bmatrix} \hat{W}_1 \\ \hat{W}_2 \\ \hat{W}_3 \\ \hat{W}_4 \end{bmatrix} = \frac{1}{L^2} \begin{bmatrix} W \times (L - x_L)(L - y_L) \\ W \times (x_L)(L - y_L) \\ W \times (x_L)(y_L) \\ W \times (L - x_L)(y_L) \end{bmatrix} = \frac{1}{L^2} \begin{bmatrix} W \times (L - x + k_1)(L - y + k_2) \\ W \times (x - k_1)(L - y + k_2) \\ W \times (x - k_1)(y - k_2) \\ W \times (L - x + k_1)(y - k_2) \end{bmatrix} \tag{5.17}$$

5.4.2 W measured calculation

When it comes to calculate the measurement vector, the first approach is directly using the value of the 4 load sensors of the tile the person is standing on. But in most cases, the person is standing on multiple tiles at the same time. By tracking the location of the centre of pressure of the person, we can calculate the resultant force distributed on the 4 load sensors of the tile that the (x, y) coordinates of the CoP belong to.
The (x, y) coordinates of the CoP are estimated using the Barycentre equation:

$$\begin{pmatrix} x_{CoP} \\ y_{CoP} \end{pmatrix} = \frac{\sum W_i \begin{pmatrix} x_i \\ y_i \end{pmatrix}}{\sum W_i} \tag{5.18}$$

where (x_i, y_i) are the absolute coordinates of each sensor. After knowing the CoP coordinates and identifying the tile i on which it's located, **the resultant Virtual Net Force becomes the same problem of the $\hat{\mathbf{W}}$ calculation** and can be solved using the (5.17) matrix equation.

5.4.3 Observation Matrix

Because the observation model is nonlinear, its linearization around the predicted pose yields the Jacobian: $H_k = \left.\frac{\partial h}{\partial X}\right|_{X=X_{k/k-1}} = \begin{bmatrix} \frac{\partial W_1}{\partial X_k} & \frac{\partial W_2}{\partial X_k} & \frac{\partial W_3}{\partial X_k} & \frac{\partial W_4}{\partial X_k} & \frac{\partial Yaw}{\partial X_k} \end{bmatrix}$

$$H_k = \frac{1}{L^2} \begin{bmatrix} -W(L - y + k_2) & -W(L - x + k_1) & 0 \\ W(L - y + k_2) & -W(x - k_1) & 0 \\ W(y - k_2) & W(x - k_1) & 0 \\ -W(y - k_2) & W(L - x + k_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Chapter 6

Multi-person localization

Tracking multiple persons in dynamic, uncontrolled environments such as indoor places, has important applications and especially for elderly people. In a cluttered environment, the received measurements may not all arise from the real targets. Some of them may be from clutter or false alarm. As a result, there will always exist ambiguities in the association between the previous known targets and measurements. Assigning wrong measurements to tracks often results in lost tracks and track breaks [19].

For these reasons, techniques dealing with data association have received much attention in multiple target tracking (MTT) research. There are many data association techniques used in MTT systems ranging from the simpler nearest-neighbor approaches to the very complex multiple hypothesis tracker (MHT). The simpler techniques are commonly used in MTT systems, but their performance degrades in clutter. The more complex MHT provides improved performance, but it is difficult to implement, and in clutter environments a large number of hypotheses may have to be maintained, which requires extensive computational resources. Because of these difficulties, some other algorithms having smaller computational requirements were developed.

The next sections present the overall adopted procedure for multi-person tracking.

6.1 Overall Procedure

Data association is the process of associating uncertain measurements to known tracks (Fig. 6.1). This procedure is summarized in the following steps:

- Prediction: to propagate the state probability distribution function (pdf) forward in time, taking process noise into account (translate, deform, and spread the pdf).

- Gating: to determine possible matching observations.
- Data association: to determine best match.
- state update using kalman filtering.

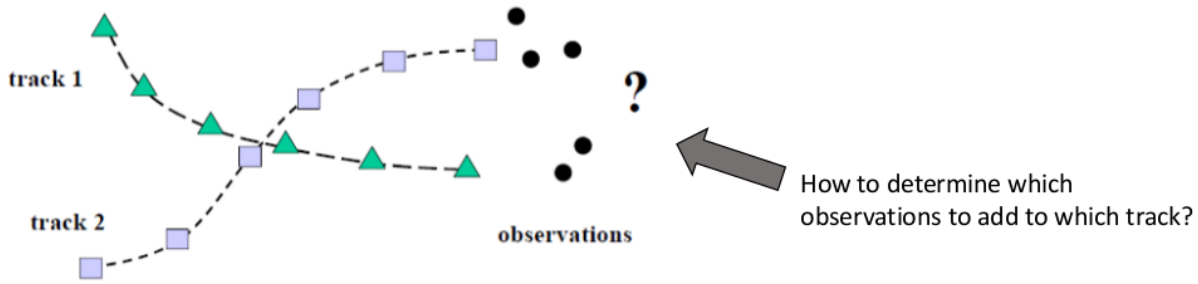


Figure 6.1: Problematic of the multi-person association.

6.2 Gating

Gating is a coarse test for eliminating unlikely observation-to-track pairing. A gate is formed around the predicted position. All measurements that satisfy the gating relationship fall within the gate and are considered for track update. The manner in which the observations are actually chosen to update the track depends on the data association method but most data association methods utilize gating in order to reduce later computation.

Figure 6.2: Gating illustration

The use of Kalman filtering (or the IF), with the associated covariance matrix, is assumed. At scan $(k-1)$, the filter evaluates the prediction $x_i(k/k-1)$ of the state vector of the ith track. The measurement at scan k is: $\boldsymbol{y_j(k) = Hx_i(k) + v(k)}$
The vector difference between measured and predicted states is: $\boldsymbol{y_{ij}(k) = y_j(k) - Hx_i(k/k-1)}$. It is defined as a residual vector with residual covariance matrix $S = HPH^T + R$, where P is the state prediction covariance matrix. Defining $\boldsymbol{{d_{ij}}^2}$ to be the norm of the residual (or innovation) vector:

$$\boldsymbol{{d_{ij}}^2 = {y_{ij}}^T S y_{ij}}$$

Define a threshold constant for gate G in a way that correlation is allowed if the following relationship is satisfied : $\boldsymbol{d_{ij}}^{\mathbf{2}} = \boldsymbol{y_{ij}}^{\boldsymbol{T}} \boldsymbol{S} \boldsymbol{y_{ij}} < \boldsymbol{G}$

6.3 Data Association

In a dense target environment additional, logic is required when an observation falls within the gates of multiple target tracks or when multiple observations fall within the gate of a target track. The optimal assignment minimizes a total distance function which is the sum of the distances for all the individual assignments.
Data association takes the output of the gating algorithm and makes final measurement-to-track associations. When a single measurement is gated to a single track, an assignment can be immediately made. However, for closely spaced targets, it is more likely that conflict situations will arise.

We assume the existence of a set of **n tracks at the time a new observation or set of observations is received**. These observations may be used for updating the existing tracks or for initiating new tracks. Suppose that **m measurements are received at time index k**. In a cluttered environment, **m does not necessarily equal n** and it may be difficult to distinguish whether a measurement originated from a target or from clutter. A validated measurement is one which is either inside or on the boundary of the validation gate of a target.

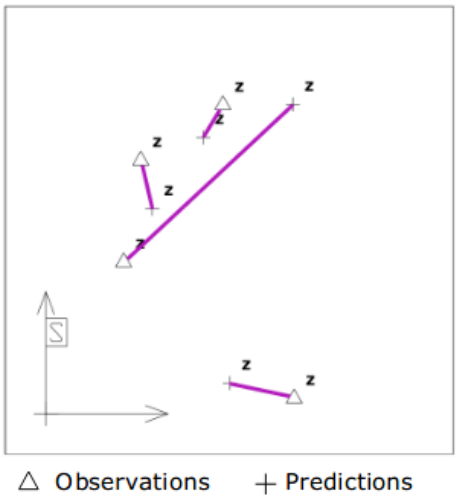
On the base of the validation test the cost matrix C for assignment problem solution is defined:

$$\boldsymbol{C_{ij}} = \begin{bmatrix} \boldsymbol{c_{11}} & \boldsymbol{c_{12}} & \boldsymbol{c_{13}} & \cdots & \boldsymbol{c_{1m}} \\ \boldsymbol{c_{21}} & \boldsymbol{c_{22}} & \boldsymbol{c_{23}} & \cdots & \boldsymbol{c_{2m}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{c_{n1}} & \boldsymbol{c_{n2}} & \boldsymbol{c_{n3}} & \cdots & \boldsymbol{c_{nm}} \end{bmatrix} \tag{6.1}$$

The elements of the cost matrix $\boldsymbol{c_{ij}} = \boldsymbol{d_{ij}}^{\mathbf{2}}$. The assignment cost matrix can be solved using different approaches, In this work we used the GNN approach that minimizes the summed total distance. In the next section, we will discuss the difference between NNSF and GNN algorithm.

6.4 Nearest Neighbor Standard Filter (NNSF)

The NNSF is defined as the measurement corresponding to the smallest normalized distance squared (NDS) among the validated measurements. The algorithm of NNSF can be summarized in the following steps:



- Build the assignment matrix C.
- Iterate:
  - Find the minimum cost assignment in C.
  - Remove the row and column of that assignment.
- Check if assignment is in the validation regions.
  - Unassociated tracks can be used for track deletion.
  - Unassociated measurements can be used for track creation.

But choosing the minimum cost (nearest musearment-to-track) everytime, obviously can not solve the global assignment problem. With some probability the selected measurement is not the correct one. This can lead to overconfident covariances, filter divergence and track loss.

6.5 Global Nearest Neighbor GNN

The GNN algorithm is based on likelihood theory, and the goal is to minimize an overall distance that considers all observation-to-track pairings that satisfy the gating test. The GNN find each time k the best association between the group of all tracks-measurements by solving the problem as a linear association problem using **Munkers algorithm or K-best assignement algorithm**:

$$min \sum {d_{ij}}^2 . x_{ij} \quad x_{ij} \in \{0, 1\} \tag{6.2}$$

$$\sum_i x_{ij} = 1 \quad \sum_j x_{ij} = 1 \tag{6.3}$$

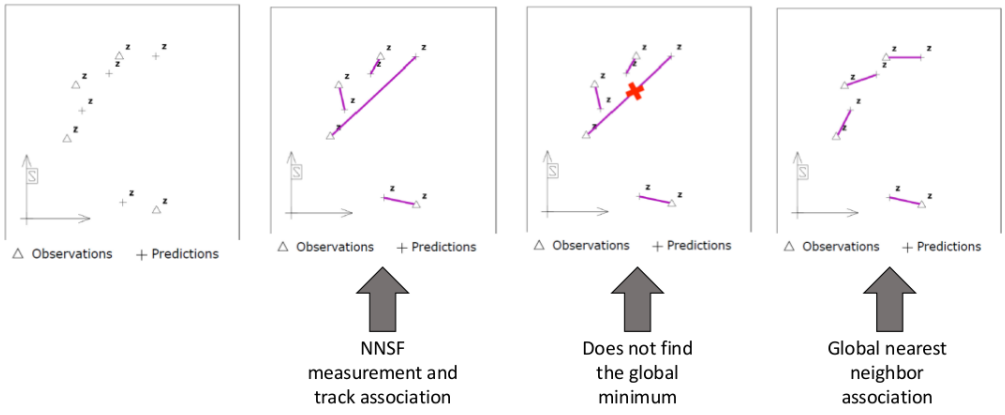


Figure 6.3: GNN Vs. NNSF

6.6 Informational Formalism for multi-tracking

In this section, we assume that we have $\boldsymbol{n}$ persons (tracks) walking together inside the apartment. The IF filter here is a generalization of the filter used in case of one person, where:

Prediction state:

$$X = \begin{bmatrix} x_1 & y_1 & \theta_1 & \dots & x_n & y_n & \theta_n \end{bmatrix}^T \quad (n \times 1) \tag{6.4}$$

$$F = \begin{bmatrix} F_1 & 0 & \dots & 0 \\ 0 & F_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & F_n \end{bmatrix} \quad (3n \times 3n) \quad B = \begin{bmatrix} B_1 & 0 & \dots & 0 \\ 0 & B_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & B_n \end{bmatrix} \quad (3n \times 2n) \tag{6.5}$$

where $F_1, F_2, \dots, F_n$ and $B_1, B_2, \dots, B_n$ are the $F(3 \times 3)$ and $B(3 \times 2)$ matrices for each person i defined in equations (5.9) and (5.10).
In the same manner we can define the covariance matrix P by the fact that there is no shared knowledge between the persons during walking [20].

$$P = \begin{bmatrix} P_1 & 0 & \dots & 0 \\ 0 & P_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & P_n \end{bmatrix} \quad (3n \times 3n) \tag{6.6}$$

The Informational matrix and vector, Y and y, are defined using the same equations in (5.11).

Correction Step: In order to correct our prediction, we have to assign for each person the best observation between all the available observations using the association algorithm defined in section (6.3).

$$Z = \begin{bmatrix} Z_1 & Z_2 & \dots & Z_n \end{bmatrix}^T \quad (5n \times 1) \tag{6.7}$$

where Z_i is the measurement (from the tiles and t-sens) associated for each person.

Distance calculation: In order to calculate the distance in the cost matrix, we have to determine the error distance between each measurement by reference to each track. The measurement lies in the observation space, so theoretically we have to transform the state information from the euclidien space (the pose of the person) to the observation space using the observation matrix. But when dealing with the data coming from the tiles, this is not always the case. If we see the equation of the $\hat{\mathrm{W}}$ we can notice that it's proportionnal with the (x, y) distance, so whenever we move away from the correct tile, the error measurement value will become greater (see (Fig. 6.4) supposing

that the correct (x, y) on the correct tile are in the middle having the values of (3.5,0.9)).

$$\hat{W} = \frac{1}{L^2} \begin{bmatrix} W \times (L - x + k_1)(L - y + k_2) \\ W \times (x - k_1)(L - y + k_2) \\ W \times (x - k_1)(y - k_2) \\ W \times (L - x + k_1)(y - k_2) \end{bmatrix} \tag{6.8}$$

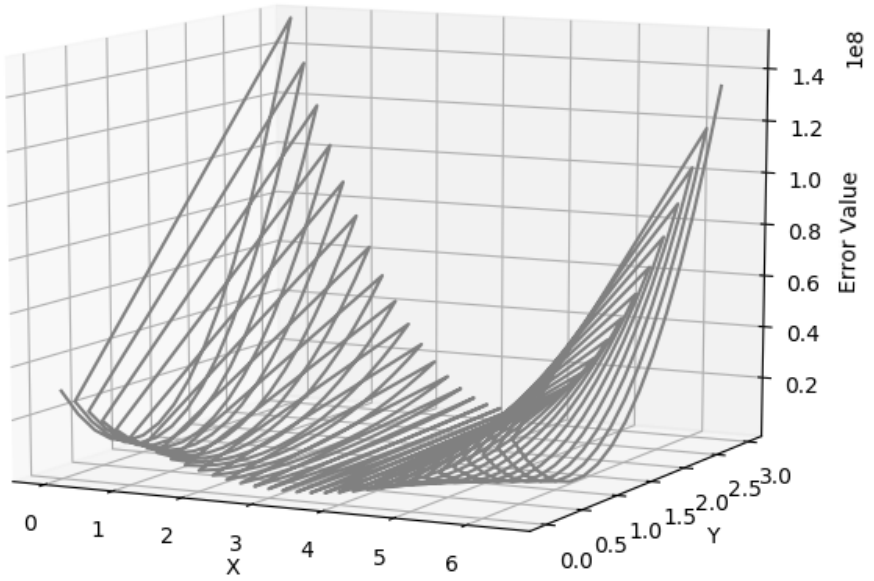


Figure 6.4: Error Function with reference to X and Y

But the problem is that the error function remains true unless the person is standing on 2 tiles at the same time. what happens here is that the weight of the person distributes on both tiles, then the effect of W factor in the equation (6.8) is no longer correct, and the euclidean distance is not proportional with the error value anymore.

The proposed solution here is to switch to the euclidean space and calculate the euclidean error function. After associating each track-measurement in this space, we can pursue normally using the informational contributions proposed before in equations (5.13).
When each person is walking on multiple tiles, we can assign multiple measurements for each track using the K-best linear assignment, and after that we can calculate the Net force using the formulas proposed before. But if two people are so close or even on the same tile (mathematically, two persons falling within the same gate), we can suppose that each person is present on only one tile (the best likely measurement). Because, if one tile from a total of 3 tiles is shared between two people we don't know which person of them is the person standing on both of the tiles.

Chapter 7

Results and Discussion

Throughout this Chapter we will discuss the data coming from the observations(tiles and T-sens), from the IMU of the watch, and the results of fusion between all of them. Then we will break down the previously done experiment into details to evaluate the model using the Force Plates.

7.1 Data Processing from tiles

In order to get the best accuracy of the model, we need to process and clean the data coming from the tiles before. For that reason many experiments have taken place.

Walking in straight line: According to the Biomechanical analysis of the human gait, we can approximate walking dynamics using the inverted pendulum formulas (section 5.1).
In the (Fig. 7.1) we can see that the result of CoP trajectory verify our assumptions.

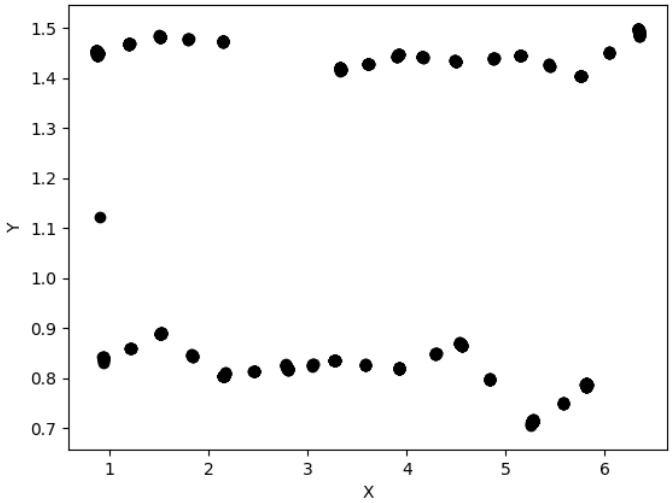


Figure 7.1: Walking in straight line and returning back.

But as we see, the CoP is passing a large distance each step (the length of the tile $L = 0.6m$), this problem is the result of not cleaning the data correctly before applying the formulas. Cleaning and calibrating the values coming from the load sensors will get us a smooth and continuous trajectory for the CoP.

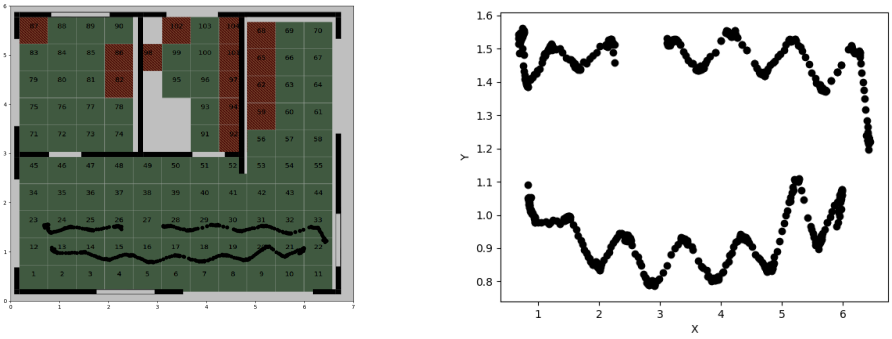


Figure 7.2: Walking in straight line and returning back after cleaning the data.

For the main experiment: the person was supposed to move in a loop shape inside the apartment wearing the sensors, in the (Fig. 7.3) we see the proposed trajectory.

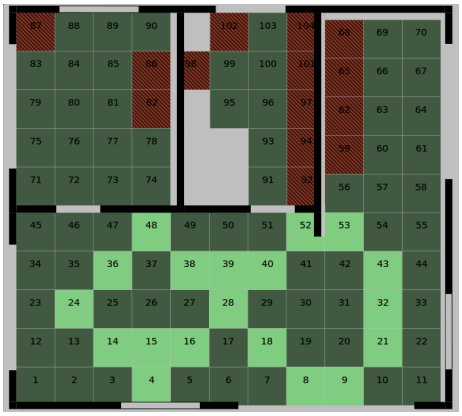


Figure 7.3: Proposed trajectory.

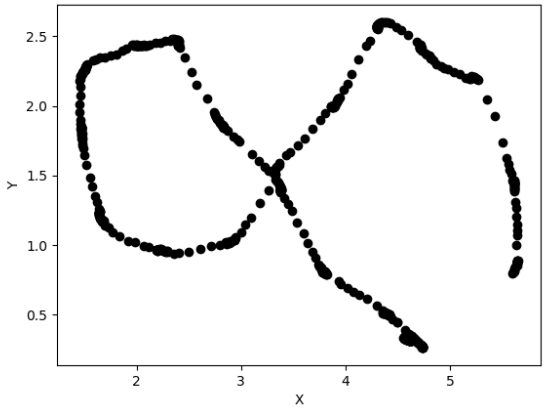


Figure 7.4: Tiles observations.

The result of the data coming from the tiles is showed in (Fig. 7.4).

7.2 Fusion of the data

When we have to fuse the data from all the sensors, noting that each one of them has its own specific frequency which differs from from the others', we come down to two approaches:

- To drop a part from the IMU data to be on the same frequency of the observation data.
- To update the prediction variables many times before the new observation arrives.

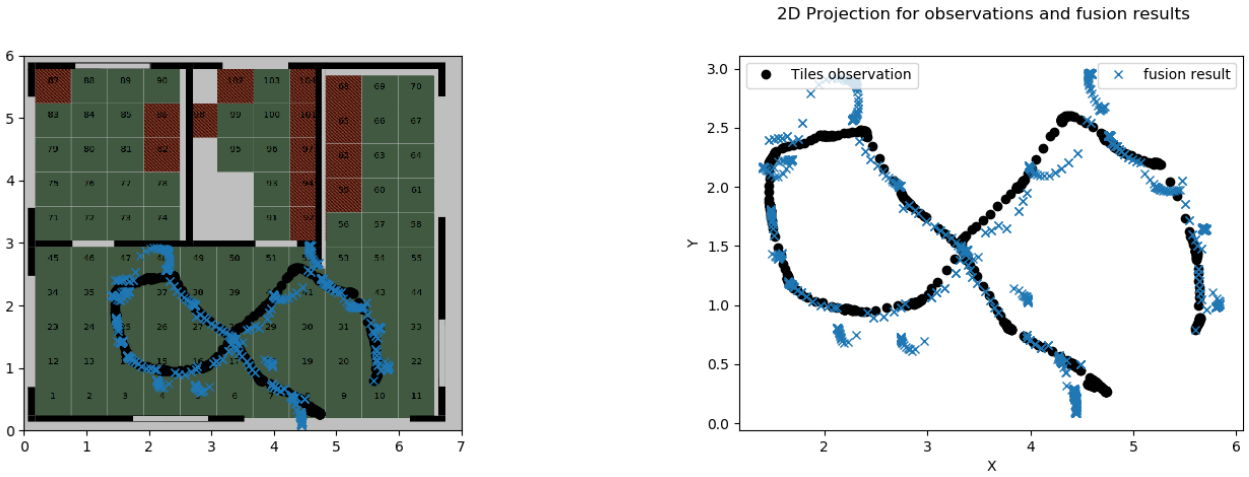


Figure 7.5: Fusion result - Approach 1.

The two approaches are tested, and the results will be discussed below:

For the first approach showed in (Fig. 7.5), we see that the data from the tiles and the result of fusion are close, but we witness a discontinuity in the results when we move from one tile to another. this discontinuity can appear also in the error function between the $\hat{Z}$ and Z. The (Fig. 7.6) shows the difference between one of the load sensors (W_3 and $\hat{W}_3$).

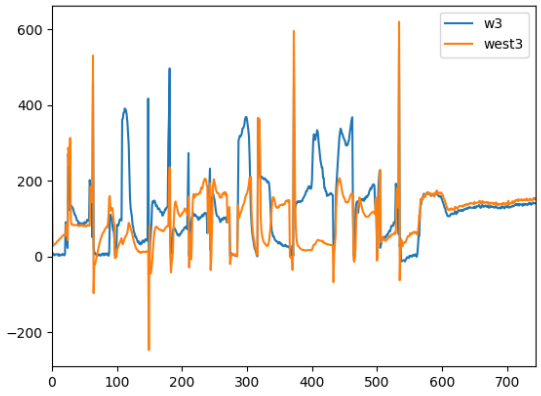


Figure 7.6: Comparison between W_3 and $\hat{W}_3$.

The reason behind the discontinuity is that when we move from one tile to another, the value of W_1 and W_3 will decrease simultaneously, and the value of W_2 and W_4 will increase simultaneously

as well, this will lead us to the discontinuity in the error function. If we keep predicting our new state with this error value the result will be reflected in a discontinuity in the calculated position.

For the second approach, the problem of simultaneous change in the value of load sensors still exists but what's different here is that when we update the prediction only on the new observation, the problem will be solved in the following observation because the error function will converge in a rapid manner. what we're proposing is to average the value of acceleration and angular velocity coming from the IMU during the time of waiting for the next observation to arrive (Fig, 7.7).

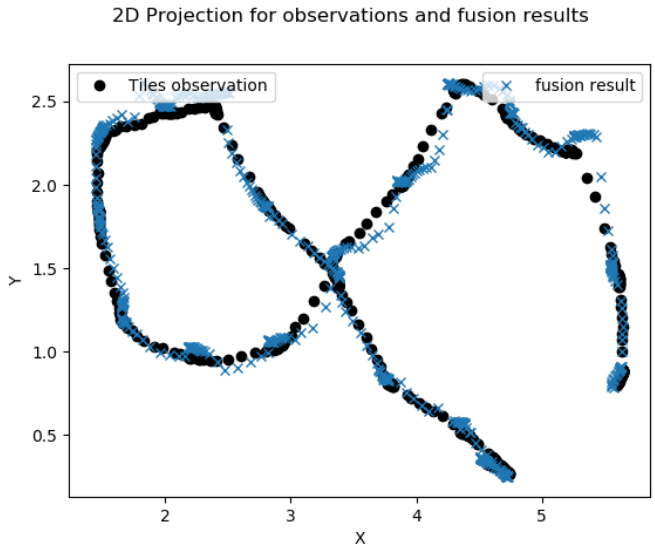


Figure 7.7: Fusion results using approach 2.

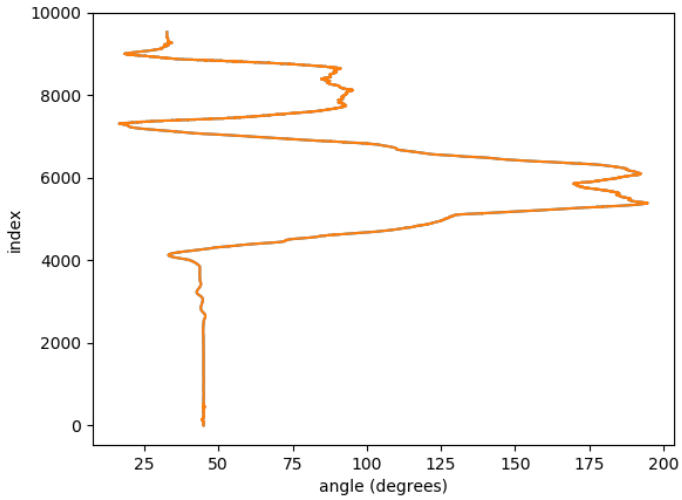


Figure 7.8: θ evolution.

7.2.1 Multi-person tracking fusion results

For the multi-person tracking, The experiment consisted of two persons walking at the same time, and a crossing occurs between the two persons on the same tile (Tile 15). The results show that the algorithm proposed was able to continue tracking the persons even after the crossing.

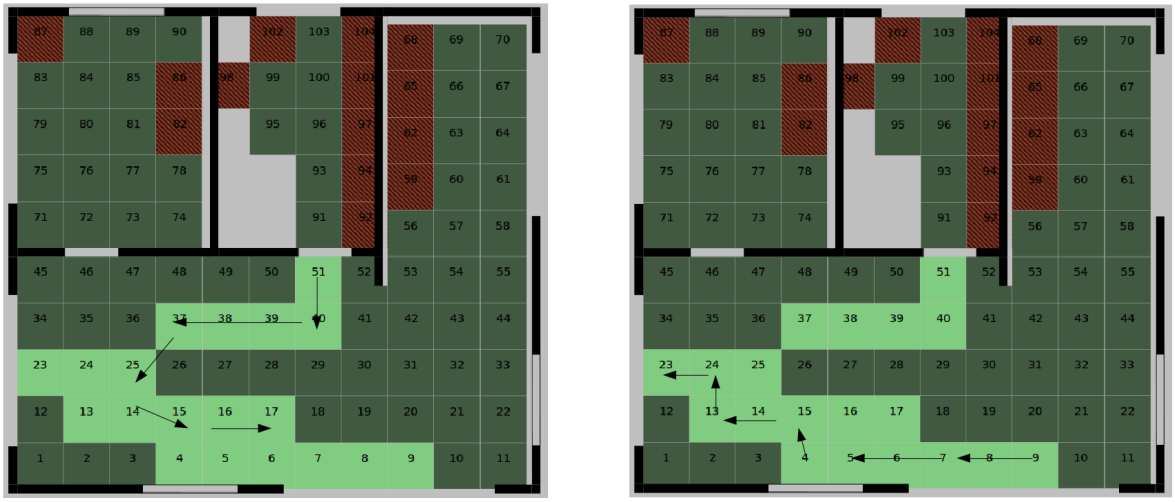


Figure 7.9: Theoretical trajectory for person one and two.

The result of the fusion and association is shown in the (Fig. 7.10):

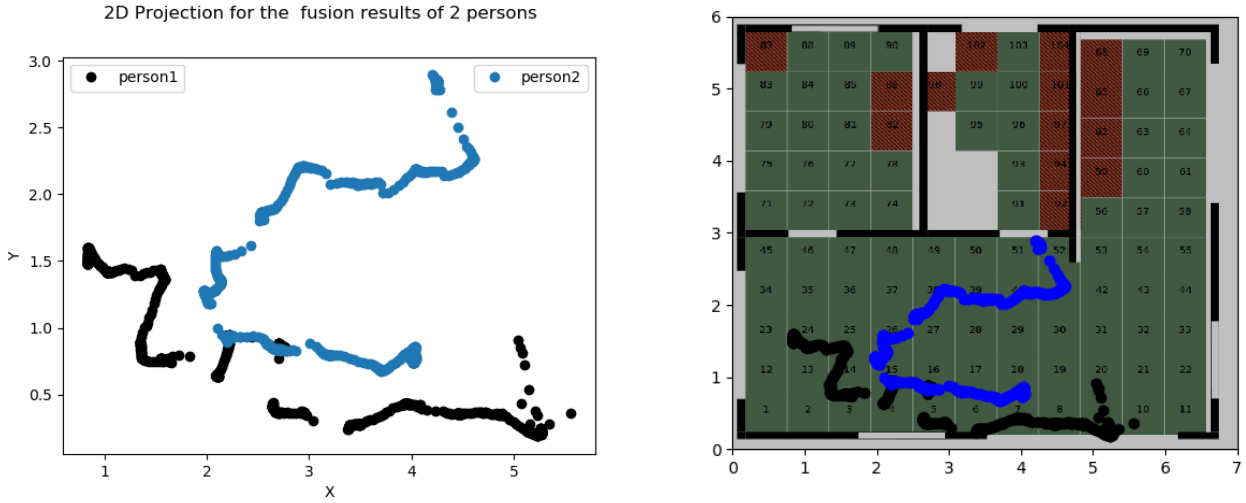


Figure 7.10: Result for multi-person fusion

7.3 Ground Truth - accuracy evaluation

To evaluate the work, an experiment was done using Force plates to compare the real Data with the data collected after the fusion. The Force Plates can measure the position of the centre of pressure in a highly accurate way.

In the experiment, 4 force plates are used, on which the user was walking forward and returning back to the initial position.

The Force Plates are synchronized with a Qualisys System, and they give the value of Forces and Moments applied on each one of them with a frequency of 100 Hz.

7.3.1 CoP computation

Let us consider a point of the sole (generally the normal projection of the ankle) and the unit normal vector directed outwards from the support surface. [21].

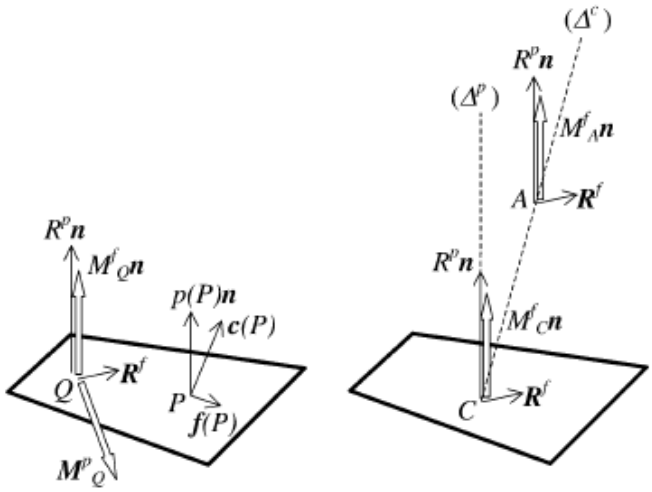


Figure 7.11: Contact forces and moments acting on the sole.

Knowing the expression of the pressure forces on the point O, in other words knowing $R^p, M_O{}^p$, the problem consists of determining the position C of the CoP. By definition, C is the point where the moment of the pressure forces vanishes. Therefore,

$$M_O{}^p = OC \times R^p \tag{7.1}$$

$$OC = \frac{n \times M_O{}^p}{R^p} \tag{7.2}$$

It is straightforward to establish that OC can be formulated as the function:

$$OC = \frac{n \times {M_O}^c}{R^c \cdot n} \tag{7.3}$$

7.3.2 Ground truth results

After calculating the CoP from the Forces Plates, Tiles, and fusion off all the sensors, the results are shown in the (Fig. 7.12).

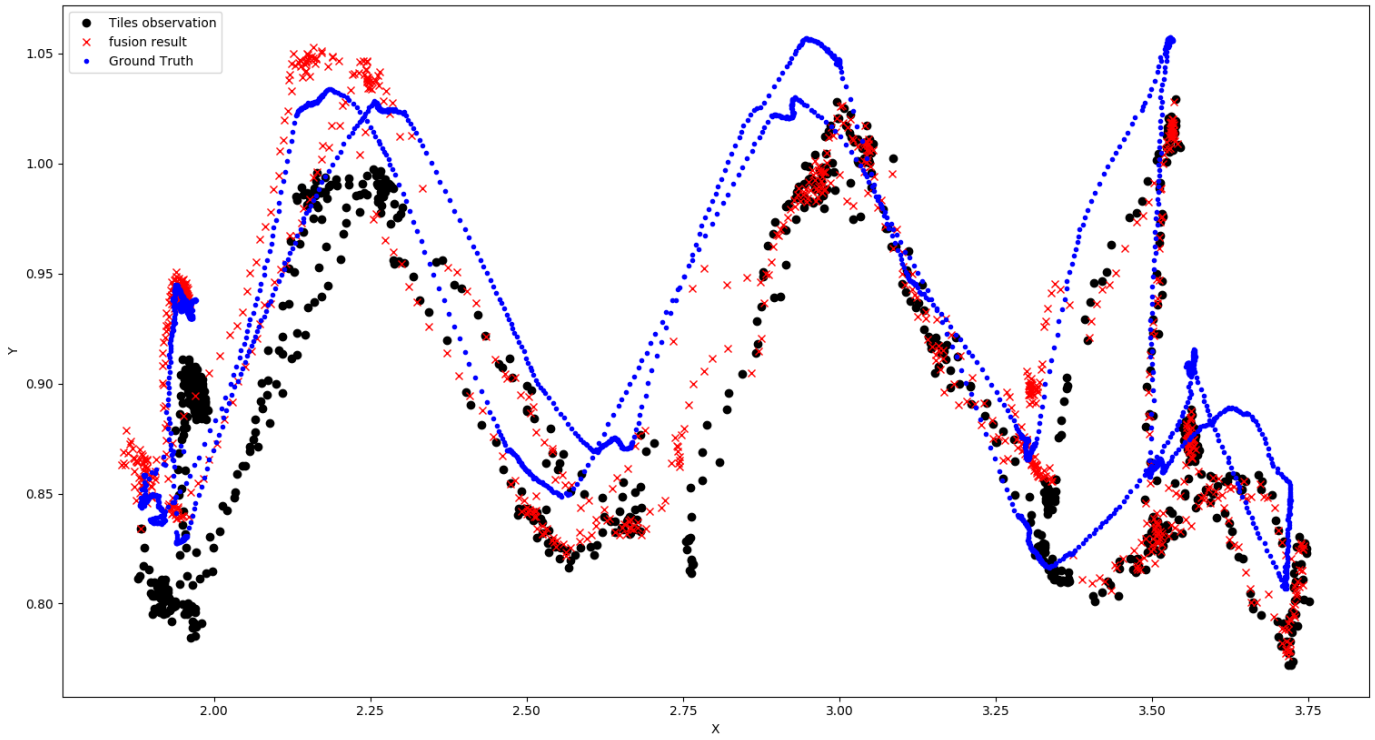


Figure 7.12: CoP plot from the tiles, the force plates, and the fusion result.

To make sure our resultant are true, we calculated the Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}} \tag{7.4}$$

RMSE value using only the data coming from the tiles = 0.066079126816207 [m]

RMSE value after the fusion of all sensors = 0.054104138942056894 [m]

The RMSE values are very close to zero, and the result after the fusion is better than the result from only the data coming from tiles.

Conclusion

Having a system that localizes and tracks people in their home with a high accuracy using only the tiles and wearable sensors can be very useful, especially for elderly people.

This internship presents a technique for the detecting and tracking of one or multiple persons inside the INRIA smart apartment.
The proposed tracking algorithm shows the ability of Bayesian Filters (Kalman filter and Informational filter) to compensate the noisy measurements, using the sensor fusion techniques.

A data association technique was developed, taking the prediction state coming from the filter as input to gate the most likely measurements, and associate between the track and measurements using the global nearest neighbour linear assignment, and finally update our beliefs using the correction state of kalman filters.

The obtained results are very promising. And a Proof of concept was done to evaluate the tracking process.

However, certain limitations can be pointed out and can be used as a basis for future works.

First of all, the gating threshold was chosen heuristically. However, choosing the threshold is a critical step in making the right decision in the association technique. Hence, developing an adaptive threshold for gating procedure will be very useful for the stability of tracking.

In some cases, the sensors give faulty measurements that can lead to track loss. A proposed solution is to integrate a layer of diagnosis for detection and identification of sensor faults to ensure a safe and reliable system of functioning using informational based approaches (using KL divergence for example for information measures), and an adaptive threshold will improve the localization integrity of the overall system.

Appendices

Appendix A

Mathematical Derivation for Velocity Model

We will now derive the algorithms for motion-model-velocity. Let $u_t = \begin{pmatrix} v & w \end{pmatrix}^T$ denote the control at time t. If both velocities are kept at a fixed value for the entire time interval $(t-1, t]$, the robot moves on a circle with radius:

$$r = |\frac{v}{w}| \tag{A.1}$$

This follows from the general relationship between the translational and rotational velocities v and w for an arbitrary object moving on a circular trajectory with radius r:

$$v = w \cdot r \tag{A.2}$$

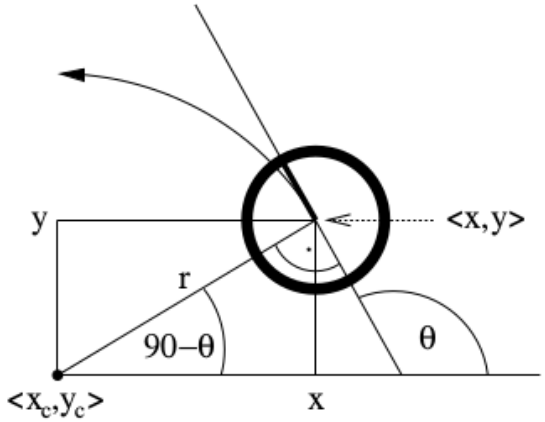


Figure A.1: Motion carried out by a robot moving with constant velocities v and w.

Equation A.1 encompasses the case where the robot does not turn at all (i.e., $w = 0$), in which case the robot moves on a straight line. A straight line corresponds to a circle with infinite radius, hence we note that r may be infinite.

Let $x_{t-1} = \begin{pmatrix} x & y & \theta \end{pmatrix}^T$ be the initial pose of the robot, and suppose we keep the velocity constant at $\begin{pmatrix} v & w \end{pmatrix}^T$ for some time Δt. As one easily shows, the center of the circle is at:

$$x_c = x - \frac{v}{w}\sin(\theta) \tag{A.3}$$

$$y_c = y + \frac{v}{w}\cos(\theta) \tag{A.4}$$

The variables $\begin{pmatrix} x_c & y_c \end{pmatrix}^T$ denote this coordinate. After Δt time of motion, our ideal robot will be at $\begin{pmatrix} x' & y' & \theta' \end{pmatrix}^T$ with:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x_c + \frac{v}{w}\sin(\theta + w\Delta t) \\ y_c - \frac{v}{w}\cos(\theta + w\Delta t) \\ \theta + w\Delta t \end{pmatrix} \tag{A.5}$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{w}\sin(\theta) + \frac{v}{w}\sin(\theta + w\Delta t) \\ \frac{v}{w}\cos(\theta) - \frac{v}{w}\cos(\theta + w\Delta t) \\ w\Delta t \end{pmatrix} \tag{A.6}$$

The derivation of this expression follows from simple trigonometry: After Δt units of time, the robot has progressed $v \cdot \Delta t$ along the circle, which caused it's heading direction to turn by $w \cdot \Delta t$. At the same time, its x and y coordinate is given by the intersection of the circle about $\begin{pmatrix} x_c & y_c \end{pmatrix}^T$, and the ray starting at $\begin{pmatrix} x_c & y_c \end{pmatrix}^T$ at the angle perpendicular to $w \cdot \Delta t$. Of course, real robots cannot jump from one velocity to another, and keep velocity constant in each time interval. To compute the kinematics with non-constant velocities, it is therefore common practice to use small values for Δt, and to approximate the actual velocity by a constant within each time interval. The (approximate) final pose is then obtained by concatenating the corresponding cyclic trajectories using the mathematical equations just stated.

Appendix B

Load Pressure $\hat{W}$ Calculation

For a person having a weight W and a position coordinates $\begin{pmatrix} x_L & y_L \end{pmatrix}$ in a tile plan of dimension L (see Fig.), the estimated load pressures amount of the ith load pressure sensor can be calculated using the Newton's laws assuming the following criteria:

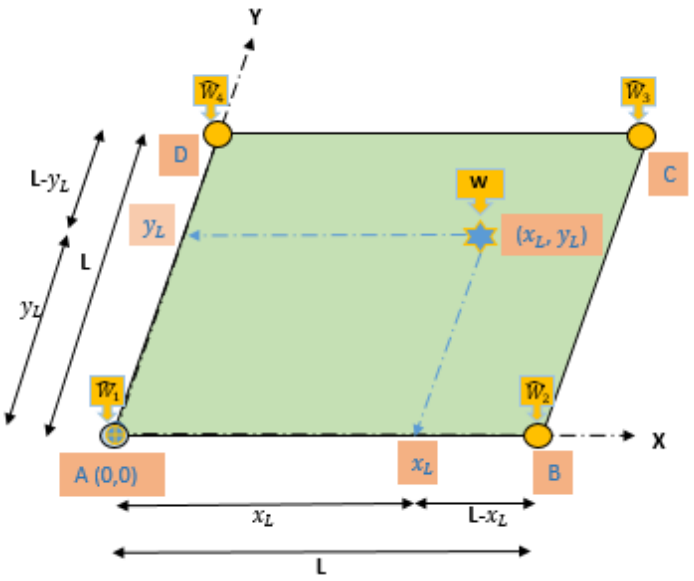


Figure B.1: A person having a weight of W and a position coordinates (x_l,y_l) in a tile plan of dimension L.

- The only points of contact between the tiles surface and the ground are the sensors,
- The ground and the sensors are considered infinitely rigid. Thus, the surface is not supposed to be deformed,
- The weight $W = mg$ is normal; where m is the mass, and g is the gravitational field strength

(about $9.81m/s^2$ on Earth); and the vertical forces "$\hat{W}_i$" applied are perpendicular to its plane,

- The system is considered stable vertically and horizontally (using of pinned or fixed supports), then it becomes isostatic.

For a two-dimensional body and based on Newton's laws of motion, the equilibrium equations available are:

- $\sum \vec{F} = 0$: The vectorial sum of the forces acting on the body equals zero.
  $\sum \vec{F} = 0 \rightarrow W - W_1 - W_2 - W_3 - W_4 = 0$
  $W = W_1 - W_2 - W_3 - W_4$

- $\sum \vec{M}_A = 0$: The sum of the moments (about an arbitrary point) of all forces equals zero.
  $\sum \vec{M}_A = 0 \rightarrow -W_1 \times 0 + (W_1 + W_4) \times y_L - W_4 \times L = 0$
  $\rightarrow W_4 = \dfrac{W(L - x_L)y_L}{L^2}$
  $\rightarrow W_1 = \dfrac{W(L - x_L)}{L} - W_4$
  $\rightarrow W_1 = \dfrac{W(L - x_L)(L - y_L)}{L^2}$
  $\sum \vec{M}_B = 0 \rightarrow -W_2 \times 0 + (W_2 + W_3) \times y_L - W_3 \times L = 0$
  $\rightarrow W_3 = \dfrac{W x_L y_L}{L^2}$
  $\rightarrow W_2 = \dfrac{W x_L}{L} - W_3$
  $\rightarrow W_2 = \dfrac{W x_L (L - y_L)}{L^2}$
  As the tile is square shaped and the only points of contact between the tiles surface and the ground are the load sensors (A, B, C, and D), thus the weight W can be divided into two parts such that: $(W_1 + W_4)$ on [AD] and $(W_2 + W_3)$ on [BC].

  Here:

$$W_1 + W_4 = \frac{W(L - x_L)}{L} \tag{B.1}$$

$$W_2 + W_3 = \frac{W(x_L)}{L} \tag{B.2}$$

Thus:

$$\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} = \begin{bmatrix} \frac{W \times (L-x_L)(L-y_L)}{L^2} \\ \frac{W \times (x_L)(L-y_L)}{L^2} \\ \frac{W \times (x_L)(y_L)}{L^2} \\ \frac{W \times (L-x_L)(y_L)}{L^2} \end{bmatrix} \tag{B.3}$$

Since the platform has a rectangular form with (nxm) tiles, we can calculate the position (x, y) using the following equation:

$$x_L = x - L[(i-1)\%n] = x - k_1 \tag{B.4}$$

$$y_L = y - L[(i)divn] = y - k_2 \tag{B.5}$$

where % operation (modulo) is the remainder of the Euclidean division, div operation is the Euclidean division, i is the identifier for the tile, x_L and y_L are the local position coordinates (the origin is at the bottom left of a current tile having the identifier equal to i), and x and y are the position coordinates of a person in the global frame where the origin $(0, 0)$ is at the bottom left corner of the cartography of the platform.

Therefore:

$$\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} = \begin{bmatrix} \frac{W \times (L-x+k_1)(L-y+k_2)}{L^2} \\ \frac{W \times (x-k_1)(L-y+k_2)}{L^2} \\ \frac{W \times (x-k_1)(y-k_2)}{L^2} \\ \frac{W \times (L-x+k_1)(y-k_2)}{L^2} \end{bmatrix} \tag{B.6}$$

References

[1] Population ageing in EuropeFacts, implications and policies. Luxembourg: Publications Office of the European Union, 2014

[2] An Aging Nation: The Older Population in the United States. By Jennifer M. Ortman, Victoria A. Velkoff, and Howard Hogan. Issued May 2014

[3] Mautz, R. Indoor Positioning Technologies. Ph.D. Thesis, ETH Zürich, Zürich, Switzerland, February 2012.

[4] A Review of Wearable Technologies for Elderly Care that Can Accurately Track Indoor Position, Recognize Physical Activities and Monitor Vital Signs in Real Time. By Zhihua Wang, Zhaochu Yang, and Tao Dong.

[5] N.Ravi, P.Shankar, A.Frankel, A.Elgammal, and L.Iftode. "Indoor localization using camera phones." In Mobile Computing Systems and Applications, 2006. WMCSA'06. Proceedings. 7th IEEE Workshop on, pp. 49-49. IEEE, 2006.

[6] Fuqiang Gu, Milad Ramezani, Kourosh Khoshelham. "Fast and Reliable WiFi Fingerprint Collection for Indoor Localization." University of Melbourne, Conference: International Conference on Urban Intelligence and Applications (ICUIA 2019)At: Wuhan, China

[7] Qigao Fan, Hai Zhang, Peng Pan, Xiangpeng Zhuang, Jie Jia, Pengsong Zhang, Zhengqing Zhao, Gaowen Zhu and Yuanyuan Tang. "Improved Pedestrian Dead Reckoning Based on a Robust Adaptive Kalman Filter for Indoor Inertial Location System." Open access journal on the science and technology of sensors and biosensors.

[8] Maxime Rio, Francis Colas, Mihai Andries and Franǫis Charpillet. "Probabilistic sensor data processing for robot localization on load-sensing floors." IEEE International Conference on Robotics and Automation (ICRA), May 2016, Stockholm, Sweden. hal-01274696

[9] Mohamad Daher , Joelle Al Hage, Maan El Badaoui El Najjar, Ahmad Diab, Mohamad Ali Khalil, and Franǫis Charpillet. "Toward High Integrity Personal Localization System Based on Informational Formalism." IEEE transactions on instrumentation and measurement.

[10] Elmenreich, W. (2002). Sensor Fusion in Time-Triggered Systems, PhD Thesis (Vienna, Austria: Vienna University of Technology. p. 173.

[11] R. R. Murphy. Biological and Cognitive Foundations of IntelligentSensorFusion. IEEE Transactions on Systems, Man and Cybernet-ics, Jan. 1996.

[12] K. E. Foote and D. J. Huebner. Error, Accuracy, and Precision.Technical report, The Geographer's Craft Project, Department of Geography, University of Texas at Austin, 1995.

[13] Gabriel A. Terejanu. "Discrete Kalman Filter Tutorial." Department of Computer Science and Engineering University at Buffalo, Buffalo, NY 14260terejanu@buffalo.edu

[14] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in Proceedings of the IEEE, vol. 92, no. 3, pp. 401-422, March 2004.

[15] Philippe Sardain and Guy Bessonnet. "Forces Acting on a Biped Robot. Center of Pressure—Zero Moment Point." IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans (Volume: 34 , Issue: 5 , Sept. 2004)

[16] Takeshi Yamaguchi, Ryosuke Okamoto, Kazuo Hokkirigawa, and Kei Masani. "Misalignment of the Desired and Measured Center of Pressure Describes Falls Caused by Slip during Turning." May 2018, Journal of Biomechanics 74.

[17] Manon Kok, Jeroen D. Hol, and Thomas B. Schon (2017), "Using Inertial Sensors for Position and Orientation Estimation", Foundations and Trends in Signal Processing: Vol. 11: No. 1-2, pp 1-153.

[18] "Probabilistic Robotics" Book by Dieter Fox, Sebastian Thrun, and Wolfram Burgard, Chap5.3 Robot Motion, Velocity motion model.

[19] Konstantinova, Pavlina and Udvarev, Alexander and Semerdjiev, Tzvetan. (2003). "A study of a target tracking algorithm using global nearest neighbor approach." 10.1145/973620.973668.

[20] Joelle Al Hage, Maan E. El Najjar, Denis Pomorski. "Multi-sensor fusion approach with fault detection and exclusion based on the Kullback-Leibler Divergence: Application on collaborative multi-robot system." Information Fusion, Volume 37, 2017.

[21] Philippe Sardain and Guy Bessonnet. "Forces Acting on a Biped Robot. Center of Pressure–Zero Moment Point." EEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans (Volume: 34 , Issue: 5 , Sept. 2004).