IMT Institute for Advanced Studies, Lucca
Lucca, Italy

Wireless Sensing Devices: From Research to Real Applications in Logistics and Healthcare

PhD Program in CSE
XXIV Cycle

By
Stefano Abbate

2012

The dissertation of Stefano Abbate is approved.

Programme Coordinator: Prof. Rocco De Nicola, IMT Institute for Advanced Studies, Lucca

Supervisor: Prof. Marco Avvenuti, University of Pisa

Tutor: Ing. Leonardo Badia, University of Padova

The dissertation of Stefano Abbate has been reviewed by:

Prof. Janet Light, University of New Brunswick (Saint John), Canada
Ing. Alessio Vecchio, University of Pisa, Italy

IMT Institute for Advanced Studies, Lucca
2012

Persistence, guts and tenancy will always get you to the top.
- My rule of life

Table of Contents

List of Figures

List of Tables

Acronyms

ERP Enterprise Resource Plan

AD Alzheimer Disease

ADC Analog Digital Converter

ADL Activities of Daily Living

AoA Angle-of-Arrival

APIT Approximate Point in Triangle

C4ISRT Command, Control, Communications, Computing, Intelligence, Surveillance, Reconnaissance and Targeting

COTS Commercial off-the-shelf

DSRC Dedicated Short-Range Communication

ECG Electrocardiography

EEG Electroencephalography

EOG Electrooculography

GPS Global Positioning System

LOS Line-of-Sight

MAC Medium Access Control

MEMS Micro Electro-Mechanical Systems

MIMS Minimally Invasive Monitoring Sensing

ML Maximum Likelihood

MP Multi-Path

NBC Nuclear, Biological and Chemical

NCP Nodes in Close Proximity

nesC Networked Embedded System C

RF Radio Frequency

RFID Radio Frequency IDentification

RIM Resident Information Management

RSS Radio Signal Strength

RSSI Radio Signal Strength Indicator

SHIMMER Sensing Health with Intelligence Modularity, Mobility and Experimental Reusability

SOC System on Chip

SVM Support Vector Machine

TDoA Time-Difference-of-Arrival

ToA Time-of-Arrival

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

USB Universal Serial Bus

UWB Ultra Wide Band

WSN Wireless Sensor Network

Acknowledgments

I would like to express my gratitude to my Research group who gave me full support to perform this work. A special thank goes to my supervisor Prof. Marco Avvenuti, and group members Ing. Alessio Vecchio and Prof. Paolo Corsini, my reference points throughout all my Research path.

Thank you to the mathematicians of the group, Mauro Passacantando e Barbara Panicucci, from the Department of Mathematics, University of Pisa. Their consultation was important to increase the fault tolerance of the localization system. I express my appreciation also to the former Master student Ing. Guglielmo Cola for the monitoring of human movements. He is now part of the Research group as PhD student.

I thank also Prof. Janet Light, from University of New Brunswick, Canada. She worked on the monitoring of human movements and guided me in the experience of being a Canadian researcher. I am also grateful to the personnel and residents of our test site: Kings Way Care Center, New Brunswick, Canada. Finally, thank you to Arnoud Simonin and Anais Zeifer, the intern students from University of Limoges, for their help during testing and in the ergonomic development of the hat solution. Part of the monitoring work was funded by Fondazione Cassa di Risparmio di Lucca, Italy under the project "Fall Detection Alarm System for Elderly People" (FiDo).

Vita and Publications

VITA

March 10, 1983	Born in Taranto, Italy
2002-2005	Computer Engineering BSc Degree Final marks: 110/110 cum laude University of Pisa, Italy
2005-2008	Computer Engineering MSc Degree Final marks: 110/110 University of Pisa, Italy
2010-2011	Visiting PhD Researcher University of New Brunswick Saint John, NB, Canada

PUBLICATIONS AND PRESENTATIONS

- S. Abbate, M. Avvenuti, P. Corsini, B. Panicucci, M. Passacantando, and A. Vecchio (2012), An integer linear programming approach for radio-based localization of shipping containers in the presence of incomplete proximity information, IEEE Transactions on Intelligent Transportation Systems.
- S. Abbate, M. Avvenuti, and J. Light (2011), MIMS: A Minimally Invasive Monitoring Sensor Platform, IEEE Sensors Journal.

- S. Abbate, J. Light and X. Li (2011), Developing cognitive decline baseline for normal ageing from sleep-EEG monitoring using wireless neurosensor devices, The 1st IEEE International Workshop on Sensor Networks for Intelligence Gathering and Monitoring (CCECE 2011 Workshop SNIGM), Niagara Falls, Ontario, Canada.

- S. Abbate, M. Avvenuti, A. Biondi and A. Vecchio (2011), Estimation of energy consumption in wireless sensor networks using TinyOS 2.X, The 8th Annual IEEE Consumer Comm. and Networking Conference - Demos (CCNC'2011 - Demos), Las Vegas, NV, USA, 1 2011.

- S. Abbate, M. Avvenuti, G. Cola, P. Corsini, J. V. Light, and A. Vecchio (2011), Recognition of false alarms in fall detection systems, in 1st IEEE International Workshop on Consumer eHealth Platforms, Services and Applications (CCNC'2011 Workshop CeHPSA), Las Vegas, NV, USA, 1 2011, pp. 538-543.

- S. Abbate, M. Avvenuti, G. Cola, P. Corsini and A. Vecchio (2010), Patent pending: Device and procedure to detect false positives in a personal device for emergency alerts, Italy.

- S. Abbate, M. Avvenuti, P. Corsini, J. Light and A. Vecchio (2010), Monitoring of Human Movements for Fall Detection and Activities Recognition in Elderly Care Using Wireless Sensor Network: a Survey, Book chapter of Wireless Sensor Networks: Application-Centric Design, ISBN: 978-953-307-321-7, InTech.

- S. Abbate, M. Avvenuti, P. Corsini and A. Vecchio (2009), Localization of shipping containers in ports and terminals using wireless sensor networks, Proc. of 12th IEEE International Conference on Computational Science and Engineering (CSE 2009), Vancouver, Canada, IEEE Computer Society, vol.2, pp.587-592, August 2009.

Abstract

Wireless Sensor Networks have the potential to enable design of innovative applications and reinvent existing ones. To this effect, the scientific community has made many contributions. However many researchers relied on strong assumptions leading to existing gaps between theory and practice. Many of the proposed solutions can work only in a laboratory setting. This thesis provides two examples on how to bridge this gap by developing two distinct real world applications in the fields of logistics and healthcare; the first application overcomes the limits of range-free localization techniques, whereas the second application provides a solution for efficiently monitoring human movements and includes a usability study. In logistics, the problem of localization of shipping containers is phased starting from the description of currently adopted systems along with their limits. Sensing nodes are placed on containers to detect the signal strength and thus the presence, of other nodes placed on other containers. The system exploits geometrical constraints and an integer linear programming solution to localize even in presence of real faulty nodes. The sensor network's functional and their non-functional issues, such as energy consumption, scalability and fault tolerance are studied. In the healthcare application, sensors are used in a monitoring system to detect falls among elderly. The approach starts from a survey of fall detection techniques to the design of our fall-detection algorithm which reduces false positives. A conceptual, minimally invasive monitoring sensor platform and reusable architecture is designed for the deployment and testing of the algorithm. The usability and acceptability study of this application in our test sites revealed some interesting insights about human aspects and time of adaptation to the technology.

Chapter 1

Introduction

A Wireless Sensor Network (WSN) is a network of small devices with the main capability of sensing. These devices, also called *nodes*, present a typical architecture of embedded computers with a microcontroller and limited computational and memory resources. They are also equipped with a radio transceiver and a power source like a battery. Progress in wireless networking, micro-fabrication and integration using micro electro-mechanical system (MEMS) technology and embedded microprocessors has enabled the wireless sensor networks for a range of commercial and military applications, promising significant changes of the way we live, work, and interact with the physical environment [1, 2, 3]. Sensor networks enable end users to directly acquire sensor measurements and provide information that is localized in time and/or space.

With such technological advances come new challenges. Most of the proposed systems based on sensor networks work only from a theoretical point of view. This because they rely on strong assumptions which hide a number of technical issues existing in WSN. The design of real-world applications is constrained by capabilities such as limited transmission range, multiple hop paradigm, limited energy sources, data transmission capacity, lifetime, and computational power. Addressing these limits is critical towards the optimal deployment strategies of sensor networks. What is needed are novel algorithms, protocols, design methodologies and tools to support application development, signal processing, information management, and networking [4]. It is necessary, for example, to have a scalable and flexible architecture so that the network preserves his stability. A minimum number of messages needs to be exchanged among the sensor nodes, to save battery power. There is a need to cope with fault tolerance and adaptability. If a sensor node fails the system must be able to operate normally. Furthermore, appropriate design must include the usability aspect which rules the interactions with the users, the time of adaptation to a new device and its invasiveness. In summary, there is a need to overcome the issues and implement optimal protocols and algorithms. But how can this be achieved? The research study described here focuses on the design of two targeted applications and provides a walk through process, starting from the feasibility aspects to the deployment of the system. In particular, two representatives categories are analyzed, localization and monitoring, to solve two problems in the fields of the logistics and healthcare respectively. In logistics, a system for localization of shipping containers is designed and simulated, whereas in healthcare, a fall detection monitoring system is designed and prototyped. Given the different nature of the two applications, a different approach is taken.

The first application is a localization technique for shipping containers based on WSNs. The approach taken starts from an analysis of the main moving operations performed on shipping container terminals and ports, together with the emerging limits. Although today Global Positioning System (GPS) devices are used, a total automation is not reached because the procedures involve workers to ensure that a container has been placed in a specific position. As a result, in case of impossibility of placing a container in the requested position, the workers have to move it to a temporary location which,is not tracked. In the approach described here, containers are equipped with wireless sensing devices to detect the exact position containers within a stack. A basic localization algorithm, namely strawman approach, is used to locate containers [5]. However this algorithm can be applied only to an ideal environment with no interferences and all nodes working. The lack of fault tolerance is therefore compensated with an algorithm designed by modeling localization as an integer programming problem. The basic idea is that geometrical constraints are used to restrict the space of possible solutions to localize the position of containers without ambiguity. Two scenarios are considered for which the overall localization rate is increased by 4.45 and 2.4 times with respect to the strawman approach. Moreover, this approach can be easily replicated for other localization techniques based on Radio Signal Strength Indicator (RSSI) and therefore characterized by high error levels.

The second application is related to the monitoring of human movements, focusing on the automated detection of falls. It is a totally different approach with respect to localization because monitoring is strictly tight to the targeted application. For this reason an in depth analysis of the fall problem has been performed and a survey about the state-of-the-art techniques has been produced. Moreover, a definition of some common metrics has been clearly stated in order to compare different fall detection techniques. A recurring problem is finding a trade-off between the number of sensors, false positives, and usability of the system. A first system has been prototyped using a novel technique for the analysis of kinetic data that proves to be useful in distinguishing real falls from the activities of daily living. This ensured a reduction of the number of false alarms. A Minimally Invasive Monitoring Sensing (MIMS) architecture has been studied in order to provide deployment guidelines for new healthcare sensor-based systems. Lastly usability and acceptability tests have been performed to ensure that the proposed system can be actually be used by people. The main contribution of this second application are the following: an in depth description of the approach used for the analysis of the problem. The use of a novel approach to filter all the false alarms coming from activities of daily living without missing any real fall, achieving 100% sensitivity and specificity values with respect to the data we collected. The prototype proves that the filtering approach can be used also to enhance basic fall detection systems. Moreover the MIMS platform presented exposes and solves basic technical problems so that it can be easily replicated and speed-up other studies. Finally, the usability study revealed some interesting insights that might be erroneously ignored during the design phase. They are related not only to technical parts but also to human aspects (e.g. color choice) difficult to figure out a priori and that can strongly reduce time of adaptation.

To give a comprehensive understanding of what is behind the design a of a WSN-based application, the remainder of this chapter provides some basic notions about WSNs, an overview of the possible applications and the discussion about the issues and challenges. Chapter 2 describes the two main categories for which sensor networks are used: localization and monitoring. In Chapter 3 localization is used to solve a logistic problem, that is localizing shipping containers on a yard, using the novel approach that does not require a GPS. In Chapter 4 an approach for monitoring human movements is described; in particular, it provides important elements to develop a fall detection system. Chapter 5 summarizes the research work, its main contributions, and future work directions.

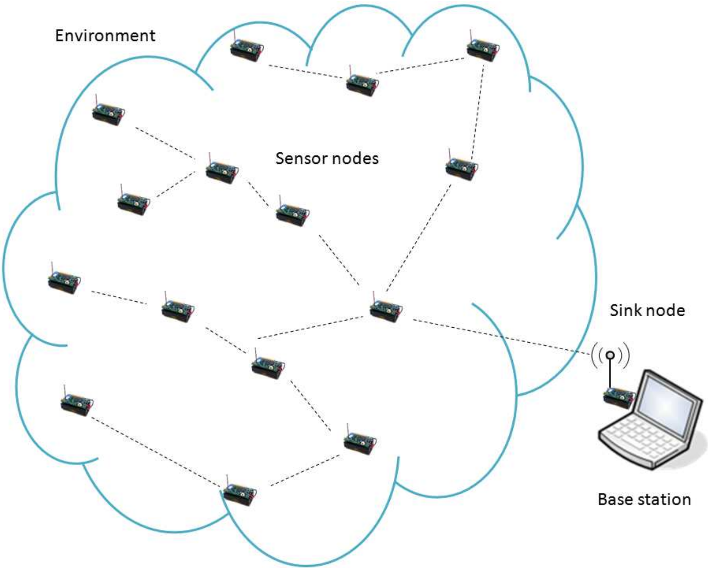


Figure 1.1: Wireless Sensor Network topology.

1.1 Background of WSN

An example of wireless sensor network topology is given in Figure 1.1. Each device is usually battery-powered and can be instrumented with one or more sensors which enable acquisition of physical data such as temperature, body acceleration and so on. The nodes are able to organize themselves in order to create an ad-hoc routing tree, whose root is represented by a *sink* node. The sink node is usually connected to a personal computer, also called *base station*, that will receive all the data sent by nodes [6]. Close nodes may organize themselves into clusters, so that sensing redundancy can be avoided and battery life prolonged. Clustering allows hierarchical structures to be built on the nodes and enables more efficient use of scarce resources, such as frequency spectrum, bandwidth, and power. Clustering also allows the health of the network to be monitored and misbehaving nodes to be identified, as some nodes in a cluster can monitor other nodes. Sometimes networks include also a mixture of nodes, including some that are more powerful or have special capabilities, such as increased communication range, GPS, and so on. Furthermore, for most advanced collaborative network functions, the sensor nodes must be placed in a known temporal and spatial frame-

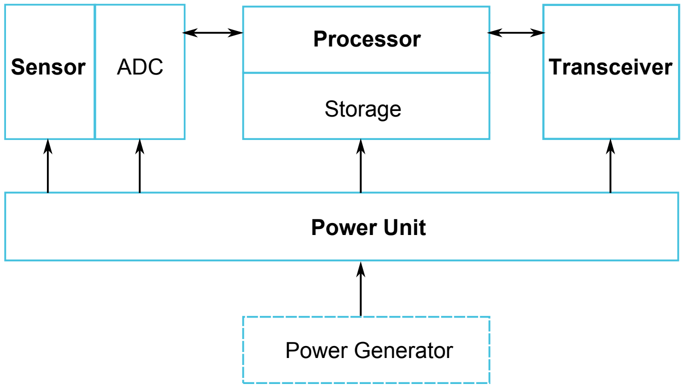


Figure 1.2: Hardware architecture of a sensor.

work. Differently from wired networks, nodes in a wireless sensor network can change the topology of the network by choosing to broadcast at less than their maximum possible power. This can be advantageous, for example, in situations where there is dense node deployment, as radio power consumes a lot of energy.

1.2 Architecture

A sensor node is made up of four basic components as shown in Figure 1.2: a sensing unit, a processing unit, a transceiver unit and a power unit. There may be also subunits such as a location finding system, a power generator and a mobilizer, according to the requirements of the application [6].

The *Sensing unit* is usually composed of two subunits: sensors and Analog Digital Converter (ADC)s. The analog signals produced by the sensors based on the observed phenomenon are converted to digital signals by the ADC, and then fed into the processing unit. The *processing unit*, which is generally associated with a small storage unit, manages the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A *transceiver unit* connects the node to the network. The *power unit* is one of the most important components of a sensor node and may be supported by a power scavenging unit such as solar cells.

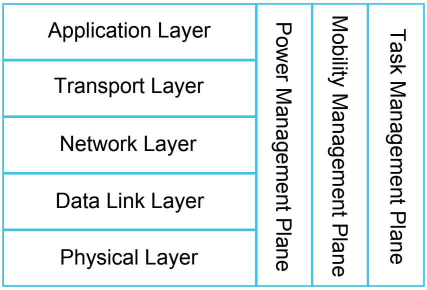


Figure 1.3: Protocol stack of a node.

Nodes are connected by a wireless medium using radio, infrared or optical media. The size of a sensor network is limited mostly by the cost of maintaining such communication links and the cost of sensor hardware. To enable global operation, these networks can be built on wireless LANs, using IEEE 802.11, Bluetooth or low-cost wireless standards such as 802.15.4/ZigBee and Ultra-Wide Band (UWB). For sensor networks, a small-sized, low-cost, ultra-low power transceiver is required because connections have a large impact on the battery lifetime. The different application requirements of sensor networks influence the choice of transmission media. For instance, marine applications may require the use of long-wavelength radiation to penetrate the water surface. Inhospitable terrain or battlefield applications require a transmission media suitable for error prone channels and environments affected by interferences.

Figure 1.3 shows the protocol stack of a sensor node. It combines power and routing awareness, integrates data with networking protocols and communicates power efficiently through the wireless medium. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer; such layers are supported by a power management plane, a mobility management plane, and a task management plane [6].

Different application layer protocols can be designed for sensor management, task assignment, data advertisement, query and dissemination, depending on the schemes used by the underlying layers. The transport layer helps to maintain the flow of data if the sensor networks application requires it. The network layer takes care of routing the data supplied by the transport layer. Since the real environment is noisy and sensor nodes can be mobile, the MAC protocol must be power aware and able to minimize collision with neighbors' broadcast. The physical layer ad-dresses the needs of a simple but robust modulation, transmission and receiving techniques.

The power, mobility, and task management planes help the sensor nodes coordinate the sensing task and lower the overall power consumption. In particular, the power management plane manages how a sensor node uses its power. For example, the sensor node may turn off its radio after receiving a message from one of its neighbors. Also, when the power level is low, the sensor node can broadcast to its neighbors that it cannot participate in routing messages, but only sensing. The mobility management plane detects and registers the movement of sensor nodes, so a route back to the user is always maintained, and the sensor nodes can keep track of which are their neighbors. This can be used, for example, to balance power and task usage. The task management plane schedules the sensing tasks given to a specific cluster. Not all sensor nodes in that cluster are required to perform the sensing task at the same time. In summary, these management planes are needed, so that sensor nodes can work together in a power efficient way, route data in a mobile sensor network, and share resources between sensor nodes.

This overview of the architecture points out that programmers have to be aware of all the protocols, in order to achieve an efficient design. For example, programmers have to be aware of the hardware characteristics and able to use special features of the processing unit and transceivers to minimize the sensor node's power consumption. This approach may push toward the design of a custom solution for different types of sensor nodes applications.

1.3 Platform and tools

A real-world sensor network application has to take into account energy, bandwidth, computation, storage, and real-time constraints. This makes sensor network application development quite different from traditional distributed system development or database programming.

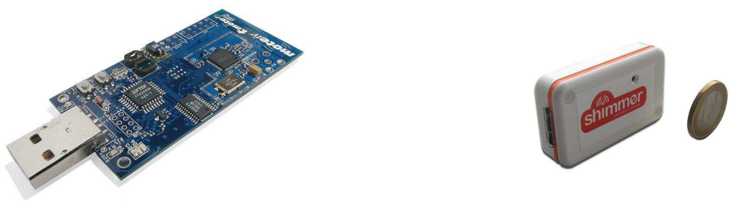
Sensor node hardware can be grouped into three categories, each of which implies a different set of trade-offs in the design choices [4]:

Augmented general-purpose computers: PC-like platforms. Compared with dedicated sensor nodes they are more power demanding. However, when power is not an issue, these platforms have the advantage of the availability of fully supported networking protocols, popular programming languages, middleware, and other off-the-shelf software.

Dedicated embedded sensor nodes: for example the Berkeley mote family [7]. These platforms typically use Commercial off-the-shelf (COTS) chip sets with emphasis on small form factor, low power processing and communication, and simple sensor interfaces. Programmers of these platforms have full access to hardware but limited software support such as TinyOS and its programming language, nesC (Networked Embedded System C).

System-on-chip (SoC) nodes: designers of these platforms try to push the hardware limits by fundamentally rethinking the hardware architecture trade-offs for a sensor node at the chip design level.

Among these hardware platforms, the Berkeley motes have gained wide popularity in the sensor network research community, due to their small form factor, open source software development, and commercial availability. The Berkeley motes are a family of embedded sensor nodes sharing roughly the same architecture. According to the specific need of the study it is possible to obtain customized hardware with reduced form factor still maintaining the same functional characteristics. Figure 1.4(a) shows Tmote-Sky, a general purpose node that is able to sense temperature, humidity and light [8], whereas Figure 1.4(b) shows SHIMMER, a smaller size version of the Tmote Sky which is more suitable to be worn by a person [9]. The SHIMMER is equipped with a tri-axial accelerometer for movement monitoring and a Secure Digital (SD) slot to locally log a large amount of data. These platforms enable addition of other sensors such as gyroscopes, in the same board.



(a) Tmote-Sky

(b) SHIMMER

Figure 1.4: Examples of nodes.

Programming sensor nodes is hard if compared to normal computer systems. The resource constrained nature of these nodes gives rise to new programming models. The embedded operating system TinyOS

2.x is an application specific and event driven platform in which all resources are known statically and programs are built from a suite of reusable system components. The event-driven execution model enables fine-grained power management and flexible scheduling, suitable for wireless communication and ADC interfaces [10].

TinyOS provides libraries for network protocols, distributed services, sensor drivers, and data acquisition tools. The programming language used to build applications is nesC: a static language where there are no dynamic memory allocation, function pointers and heap space. NesC is based on few main concepts [11]:

- An application consists of one or more components, wired together to form a runnable program.
- Each component can provide interfaces to other components or use interfaces provided by other components.
- Each interface is bidirectional and can include a set of commands which are implemented by the interface provider, and a set of responses to events, that is functions implemented by the user of the interface.
- Components can be of two types: modules and configurations. Modules provide the implementations of one or more interfaces. Configurations are used to wire other components together.

The success of TinyOS and nesC is due to the fact that they are open-source and they have been published in a short time. TinyOS has been ported to over a dozen hardware platforms and numerous sensor boards. A wide community uses simulation tools based on TinyOS to develop and test algorithms and protocols, in order to increase the reliability of the applications built.

1.4 Typical applications of WSN

The majority of today's sensors rely on wired infrastructure to provide power and transfer data. The main problem with these sensors is the cost of wiring and delays in deployment. Wired sensor networks are usually deployed in two ways [6]:

- Sensors can be placed far from the actual phenomenon, e.g., a satellite probe that uses some complex techniques to distinguish targets from environmental noise.

- Multiple sensors are placed in fixed positions and with established connections. They periodically transmit the sensed data to one or more central nodes in which computation and data fusion are performed.

This makes not feasible certain applications, where mobility and rapid deployment are fundamental. Wireless sensor networks can reduce cost by replacing wired sensors in existing applications and can enable new applications. For example, a deployed wireless sensor network acts autonomously, by self-organizing and establishing a communication to the sink node: this is of advantage for the deployment in inaccessible terrains or disaster relief operations.

Different applications can vary in requirements, deployment, power supply or sensing capabilities. The sensors used can be seismic, magnetic, thermal, visual, infrared, acoustic, radar etc., according to the operational environment element to be measured [12]: temperature, humidity, vehicular movement, lightning condition, pressure, soil makeup, noise levels, the presence of certain kinds of objects, etc. The range of applications can be classified in in two categories [13] as shown in Table 1.1:

1. *Localization*: the estimation of the state of a physical entity such as a physical phenomenon or a sensor node from a set of measurements.

2. *Monitoring*: the periodic observation of the state of a physical phenomenon. It includes applications like indoor/outdoor environmental monitoring, health and wellness monitoring, power monitoring, inventory location monitoring, factory and process automation and seismic and structural monitoring.

In the literature Localization is strictly related to Tracking. Tracking produces a series of estimates over time and includes applications like tracking objects, animals, humans, and vehicles. Another classification is given by grouping a subset of the applications by sector [14]:

- Industrial, Building, Home Automation.

Sensor Network			
Localization		**Monitoring**	
Military Enemy Localization	**Habitat** Animal Localization	**Military** Security Detection	**Habitat** Animal Monitoring
Business Human Localization	**Public/Industrial** Traffic Localization Car Localization Bus Localization	**Business** Inventory Monitoring	**Public/Industrial** Structural Monitoring Factory Monitoring Inventory Monitoring Machine Monitoring Chemical Monitoring
		Health Patient Monitoring	**Environment** Environm. Monitoring

Table 1.1: Applications grouped by categories.

- Transportation & Logistics.
- Environment & Territory Monitoring.
- Healthcare and Assisted living.
- Military.
- Security.

In the following sections some examples of applications are provided in order to show the potential of a WSN.

1.4.1 Industrial, Building, Home Automation

Wireless sensors may be used to sense and diagnose manufacturing processes, appliances, factory, supply chains or the condition of industrial equipment. Chemical plants or oil refineries may have miles of pipelines that can be effectively instrumented and monitored. Using smart sensors, the condition of equipment in the field and factories can be monitored to alert for failures. The equipment to be monitored can range from turbine engines to automobiles, photocopiers, and washing machines. The industry is moving from scheduled maintenance, such as sending a car for a checkup every a fixed amount of kilometers, to maintenance based on condition indicators. The condition-based monitoring is expected to significantly reduce the cost for service and maintenance, increase machine up-time, improve customer satisfaction, and even save lives.

With emerging standards such as Dedicated Short-Range Communication (DSRC) designated for vehicle-to-vehicle communications, cars will soon be able to talk to each other and to road infrastructures. Application of these sensors can rise emergency alerts and driver safety assistance. During an emergency brake, an alert message from the braking car can be broadcast to nearby cars so that preventive measures may be taken. Telematics for the analysis on sensor data about tire pressure, speed, outside temperature, and vehicle model, can lead to other applications.

Sensors embedded in a building can drastically cut down energy costs by monitoring and regulate the temperature and lighting conditions. Sensors in a ventilation system may also be able to detect biological agents or chemical pollutants. They can be also an alternative to wired control devices such as light switches due to the high cost of wiring.

As technology advances, context-aware computing is becoming more important: sensor nodes and actuators can be placed in appliances, such as vacuum cleaners, micro-wave ovens, refrigerators, and multimedia players [15]. These sensor nodes inside the domestic devices can interact with each other and with the external network via the Internet. They allow end users to manage home devices locally and remotely more easily.

1.4.2 Transportation & Logistics

Sensors may be used to monitor and track assets such as trucks or other equipment, especially in an area without a fixed networking infrastructure. Sensors may also be used to manage assets for industries such as oil and gas, utility, and aerospace. These tracking sensors can vary from GPS-equipped locators to passive Radio Frequency IDentification (RFID) tags. The automated logging system can reduce errors in manual data entry. More importantly, businesses such as trucking, construction, and utility companies can significantly improve asset utilization using real-time information about equipment location and condition. Furthermore, the asset information can be linked with other databases such as enterprise resource planning (ERP) databases, providing decision makers a global, real-time picture in order to optimally utilize available resources. Each item in a warehouse may have a sensor node attached. The end users can find out the exact location of the item and count the number of items in the same category.

Another interesting application is Smart Transportation. The collection of real-time traffic or other information using cars equipped with wireless connections make roads safer and less congested.

1.4.3 Environment & Territory Monitoring

Sensors can be used to monitor conditions and movements of wild animals or plants in wildlife habitats where minimal disturbance to the habitats is desired. Some environmental applications of sensor networks include tracking the movements of birds, small animals, and insects.

Sensors can also monitor air quality and track environmental pollutants, wildfires, or other natural or man-made disasters. Additionally, sensors can monitor biological or chemical hazards to provide early warnings. Earthquake monitoring is another application area; seismic sensors instrumented in a building can detect the direction and magnitude of a quake and provide an assessment of the building safety. Since sensor nodes may be deployed in a forest, sensor nodes can be used for forest fire detection. Thus, they would be able to relay the exact origin of the fire to the end users before the fire becomes uncontrollable. For flood detection several types of rainfall, water level and weather sensors can be deployed. These sensors supply information to a centralized database system in a pre-defined way.

Although satellite and airborne sensors are useful in observing large biodiversity, e.g., spatial complexity of dominant plant species, they are not fine grain enough to observe small size biodiversity [16]. This can be achieved with a ground level deployment of wireless sensor nodes [17]. For precision agriculture related applications, monitoring environmental conditions that affect crops, livestock, irrigation such as the pesticides level in the drinking water, the level of soil erosion, and the level of air pollution in real-time can lead to a higher and healthy production.

Wireless sensor networks enable the interaction with objects in museums. These objects might be able to respond to touch and speech. Also, children can participate in real time cause-and-effect experiments, which can teach them about science and environment. In addition, wireless sensor networks can provide paging and localization inside the museum.

In the past few years there have been advances in the cooperation between Unmanned Aerial/Ground Vehicles (UAVs/UGVs) and ground wireless sensor-actuator networks whose nodes are carried by vehicles or people. The UAVs are now able to be coordinated for missions such as the detection and monitoring of events [18]. The use of mobile nodes can provide the ability to dynamically adapt the network to environmental events and to improve the network connectivity in case of static nodes failure. However, in many scenarios, the motion of the mobile nodes installed on ground vehicles or carried by persons is very constrained, due to the characteristics of the terrain or the dangerous conditions involved. These scenarios can be found, but not limited to, in civil security and disasters, in which the use of UAVs is very suitable. Cooperation of UAVs with the ground wireless sensor network offers many potential applications, including Space applications.

1.4.4 Healthcare and Assisted living

The physiological data collected by the sensor networks can be stored for a long period of time, and can be used for medical exploration. The installed sensor networks can also monitor and detect elder people's behavior such as a fall [19, 20]. These small sensor nodes allow the subject a greater freedom of movement and allow doctors to identify pre-defined symptoms earlier. As a result, they facilitate a higher quality of life for the subjects.

Each subject can wear one or more small and light weight sensor nodes which have a specific task. For example, one sensor node may detect the heart rate while another detects the blood pressure. Doctors may also carry a sensor node, which allows other doctors and nurses to locate them within the hospital. If sensor nodes can be attached to medications, the chance of getting and prescribing the wrong medication to subjects can be minimized. On the other hand, subjects can carry sensor nodes to identify their allergies and required medications.

1.4.5 Military

Wireless sensor networks can be an integral part of military Command, Control, Communications, Computing, Intelligence, Surveillance, Reconnaissance and Targeting (C4ISRT) systems. Wireless sensors can be rapidly deployed, either by themselves, without an established infrastructure, or working with other assets such as radar arrays and long-haul communication links. They are well suited to collect information about enemy target presence and to track their movement in a battlefield. For example, sensors can be networked to protect a perimeter of a base in a hostile environment. They may be thrown over the-hill to gather enemy troop movement data, or deployed to detect targets under foliage or other cover that render radar or satellite-based detectors less useful. In military applications, the form factor, ability to withstand shock and other impact, reliability and interoperability are among the most important characteristics.

Leaders and commanders can constantly monitor the status of friendly troops, the condition and the availability of the equipment and the ammunition in a battlefield by the use of sensor networks. Every troop, vehicle, equipment and critical ammunition can be attached with small sensors that report the status. These reports are gathered in sink nodes and sent to the troop leaders. The data can also be forwarded to the upper levels of the command hierarchy while being aggregated with the data from other units at each level.

In chemical and biological warfare, being close to ground zero is important for timely and accurate detection of the agents. Sensor networks deployed in the friendly region and used as a NBC warning system can provide the friendly forces with critical reaction time, without exposing them to toxic substances or nuclear radiations.

1.4.6 Security

An important application of sensor networks is in security monitoring and surveillance for buildings, airports, subways, or other critical infrastructure such as power and telecom grids and nuclear power plants. Sensors may also be used to improve the safety of roads by providing warnings of approaching cars at intersections; they can safeguard perimeters of critical facilities or authenticate users. Imager or video sensors can be very useful in identifying and tracking moving entities, such as people or vehicles. Sensor nodes are being deployed to detect and identify threats within a geographic region and report these threats to remote end users by the Internet for analysis [21].

1.5 Issues in WSN-based applications

The limits of the applications are strictly related to the non-functional issues of a wireless sensor network. The issues include energy consumption, fault tolerance, scalability, production costs and deployment of the nodes. In order to address these important factors, we need to act both on the hardware and software design level with protocols and algorithms. The design varies on the basis of the application to which a sensor network is intended. It has to be done according to a system performance goal, defined by parameters such as detection of false alarms or misses, classification errors, and tracking accuracy. After that, it has to be validated with an evaluation metric that, depending upon the application, includes packet loss, network dwell time, track loss, false alarm rate, processing latency, and so on.

1.5.1 Energy consumption

A wireless sensor node has limited size and can only be equipped with a limited power source (e.g., $\leq 0.5Ah, 1.2V$). In some application scenarios, replenishment of power resources might be impossible. Sometimes it is possible to extend the lifetime of the sensor networks by energy scavenging [22], which means extracting energy from the environment. Solar cells are an example for the techniques used for energy scavenging. Hence, power conservation and power management take on additional importance.

Application specific protocols can be designed by appropriately trading off other performance metrics such as delay and throughput with power efficiency [23]. Power consumption can be considered according to three domains:

- *Sensing*, for which the power varies with the nature of applications. Sporadic sensing might consume less power than constant event monitoring. Higher ambient noise levels might cause significant corruption, increase detection complexity and, therefore, the energy consumption.
- *Communication*, for which the energy consumed by a sensor node is the maximum. This involves both data transmission and reception and their costs are nearly the same for short-range communication with low radiation power. It is important that we consider not only the active power but also the start-up power consumption in the transceiver circuitry. As the transmission packet size is reduced, the start-up power consumption starts to dominate the active power consumption. As a result a large amount of power is spent in the transceiver switching activity.
- *Data processing*, for which the energy consumption is lower compared to data communication. To minimize power consumption, a sensor node must have built-in computational abilities which enable filtering, aggregating shaping of data and to send only a small and relevant quantity through the radio.

1.5.2 Fault tolerance

Some sensor nodes may fail or be blocked due to lack of power, have physical damage or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures [24]. Protocols must be designed taking into account that fault tolerance depends on the application. For example, if sensor nodes are being deployed in a battlefield for surveillance and detection, then the fault tolerance has to be high because the sensed data are critical and sensor nodes can be destroyed by hostile actions.

1.5.3 Scalability

Many sensing tasks require a sensor network system to process data cooperatively and to combine information from multiple sources. In traditional centralized sensing and signal processing systems, raw data collected by sensors are not always processed locally. From the scalability point of view, this reduces the available bandwidth. Moreover, as the number of nodes increases, every node spends almost all of its time forwarding packets of other nodes. Thus, depending on the application, it becomes critical to carefully select the sensor nodes that participate in a sensor collaboration, balancing the information contribution of each against its resource consumption.

1.5.4 Production costs

Since the sensor networks consist of a large number of sensor nodes, the cost of a single node is very important to justify the overall cost of the networks. If the cost of the network is more expensive than deploying traditional sensors, then the sensor network is not cost-justified. As a result, the cost of each sensor node has to be kept low. The cost of a Bluetooth radio, which is known to be a low-cost device, is ten times more expensive than the targeted price for a sensor node. Note that a sensor node may also be equipped with other devices and therefore the cost is a very challenging issue.

1.5.5 Deployment of WSN

A great amount of inaccessible and unattended sensor nodes, which are prone to frequent failures, make topology maintenance a challenging task. Sensor nodes are densely deployed either very close or directly inside the phenomenon to be observed. Therefore they may be working in busy intersections, inside a large machinery, at the bottom of an ocean, in a chemically contaminated field, in a battlefield, under high pressure, in harsh environments etc.

Deploying high number of nodes densely requires careful handling of topology maintenance. Sensor nodes can be either thrown in mass or placed one by one in the sensor field. They can be deployed by dropping from a plane or placed one by one by a human or a robot. The schemes for initial deployment must reduce the installation cost, eliminate the need for any pre-organization and pre-planning, increase the flexibility of arrangement, and promote self-organization and fault tolerance.

After deployment, topology changes are due to change in sensor nodes' position, reachability, available energy, malfunctioning, and task details. Device failure is a regular or common event due to energy depletion or destruction. It is also possible to have sensor networks with static or highly mobile nodes. Therefore, sensor network topologies are prone to frequent changes after deployment, also because additional sensor nodes can be re-deployed at any time to replace the malfunctioning nodes. Topology changes in presence of stringent power consumption constraints requires special routing protocols.

Chapter 2

Overview of localization and monitoring

This chapter provides an overview of localization and monitoring tasks, which are at the basis of many sensor network applications. Their description is useful to better understand the choices made in chapters 3 and 4 for the analysis, design and implementation of applications in the fields of logistics and healthcare. In logistics, localization is used in the design of a robust system able to detect the placement of shipping containers on a yard, without the use of GPS. The monitoring task is translated into the observation of human movements and detection of specific unwanted events such as falls.

2.1 Localization

Localization is one of the most challenging research activities and most of the applications have an explicit or implicit need to localize the nodes. In general, in a typical sensor network it is necessary to pinpoint the place in which each node is going to operate. This can be more or less accurate according to the definition of a precise location or a generic zone such as a room.

The localization problem is not restricted to *node localization*, but can cover also *target localization* and *location service* [25]. Target localization is the process of obtaining the coordinates of an event or a target present in the sensor network. The location of a target can be obtained with or without its cooperation with the sensor network. A location service acts as a repository that can be used to query the location of entities.

Node localization schemes can be categorized into two groups: range-based and range-free. *Range-based* methods rely on the possibility of measuring the absolute distance between nodes. Distance can be estimated through the RSSI, the Time-of-Arrival (ToA) or Angle-of-Arrival (AoA) of the communication signal from the sender to the receiver. This approach generally achieves fine grained localization and requires a sophisticated hardware since the measurements have to cope with high signal speed, Multi-Path (MP) propagation, fading, shadowing and low tolerance for clock synchronization error. Other methods such as acoustic and Ultra Wide Band (UWB) have become quite popular because of the high accuracy that can be achieved. Moreover UWB methods are more accurate and consume less energy than the acoustic methods. *Range-free* methods, in contrast, never try to estimate the point-to-point distance between nodes, but rely on high-level connectivity (proximity) information. As a result, there is no need of complex hardware and this makes them a cost-effective solution, even if the localization accuracy is coarse-grained.

In range-based techniques, the majority of existing location discovery approaches consist of two phases [26]:

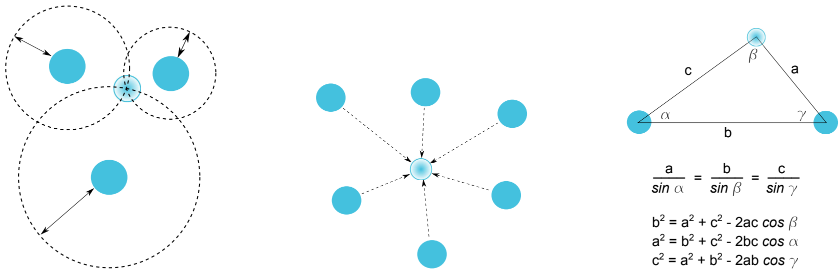
1. *Estimation* of a distance.

2. *Combining* of the distances.

The measurement estimation of the distance is given by one of the most popular methods [27]:

- *RSSI*, used to measure the power of the signal at the receiver. If transmitter power and power loss are known, it is possible to obtain a theoretic and empirical distance estimation. It is mainly used for Radio Frequency (RF) signals. More information about RSSI localization can be found in [28].

- *ToA, TDoA*, to measure the Time-of-Arrival or the Time-Difference-of-Arrival. The propagation speed can be translated into a distance and the signal can be RF, acoustic, infrared or ultrasound.



(a) Trilateration. (b) Multilateration. (c) Triangulation.

Figure 2.1: Combining the measured distances in range-based localization.

- *AoA*, to measure the Angle-of-Arrival at which signals are received and use geometrical relationships to calculate the position of the nodes.

The combination of the measured distances can be done in different ways:

- *Trilateration* locates a node by calculating the intersection of 3 circles, as shown in Figure 2.1(a).
- *Multilateration* estimates the position of a node by measuring the time difference of signal from reference points, as shown in Figure 2.1(b).
- *Triangulation* used in AoA systems for which the direction of the nodes and not the distance is estimated. In this case the position is calculated using the trigonometry laws of sines and cosines, as shown in Figure 2.1(c).

The most basic localization system using TDoA techniques is GPS deployed in 1993 and based on the NAVSTAR satellite constellation (24 satellites). So far, many systems have been developed using range-based localization techniques.

Range-free techniques are used in settings where Radio Signal Strength (RSS) and other ranging technologies cannot be used directly to estimate distances or the cost of hardware required by range-based solutions may be inappropriate. In every sensor network, each node knows what other nodes it can talk to directly, its one-hop neighbors. If the sensor nodes are densely and uniformly deployed, then hop counts to anchors can be used as a substitute to physical distance estimates. In this setting, each anchor floods the network with a broadcast message whose hop count is incremented as it is passed from node to node. The hop count in the message from an anchor that first reaches a node is the hop distance of that node to the anchor (standard graph-based breadth-first search). In order to transform hop counts into approximate distances, the system must estimate the average distance corresponding to a hop. This can be done either by using inter-anchor distances that are known in both hop and Euclidean terms [29], or by using prior information about the size of the area where the nodes are deployed and their density [30]. RSS can be used also only for distance comparisons to anchors and not for distance estimation. In the neighborhood of a node, a slight displacement that results in increased RSS from an anchors can be taken to indicate the node has moved closer to the anchor. Correspondingly, reduced RSS from a anchor can be taken to indicate the node has moved farther from the anchor. Even if sensor nodes cannot move, they can interrogate their neighbors for their RSS and thus can make inferences about relative distances to anchors.

Generally, in sensor networks, some nodes are equipped with special positioning devices which are aware of their locations. These nodes are called *beacons*, *anchors* or *landmarks*. Other nodes that do not initially know their locations are called *unknown* or *strayed*. When these systems perform localization, the unknown nodes are located using range-based or range-free based methods. An unknown can estimate its location if three or more beacons are available in its 2-D coverage. Once an unknown has estimated its position, it becomes a beacon and other unknowns can use it in their position estimations. The major challenge in localization with beacons is to make localization algorithms as robust as possible using as few beacons as possible. The resulting design consumes little energy and few radio resources.

In indoor or urban environments a larger network may be designed to operate without beacons, which is known as *beacon-free* or *anchor-free* design. Such a design determines the position of every node via local node-to-node communication. Anchor-free positioning should be a fully decentralized solution: all nodes start from a random initial coordinate assignment. Then, they cooperate with each other using only local distance estimations to figure out a coordinate assignment. The resulting coordinate assignment has both translation and orientation degrees of freedom and has to be correctly scaled. A post-process is needed to convert the translation and orientation coordinate assignment to absolute position information based on reference information acquired, for example, from GPS [30].

Localization algorithms can be centralized or distributed. Both the algorithms must face the high relative costs of communication. Centralized algorithms in large networks require each sensed data to be passed over many hops to a central unit, while distributed algorithms have sensors sending messages to the first hop. The energy efficiency of centralized and distributed algorithms can be compared. In general, when the average number of hops to the central unit exceeds the necessary number of iterations, distributed algorithms will likely save communication energy costs. There may be hybrid algorithms to reduce the energy consumption. In [31], for example, the sensor network is divided into small clusters and an algorithm selects a central unit from within each cluster to estimate a map of the sensors. Then, cluster central units run a distributed algorithm to merge and optimize the local estimates.

2.1.1 Shipping containers localization

Research about smart containers gained momentum in the last years, pushed not only by recent advances in emerging sensor technologies and miniaturization, but also by governative initiatives and regulations (such as the Advanced Container Security Device program or the Marine Asset Tag Tracking program of the Department of Homeland Security of the USA [32]). In particular, radio frequency identification and wireless sensor network have been the primary technological solutions used to explore new research directions, such as enhancement of security and intrusion detection [33, 34, 35], and detection of damages to goods [36].

A paper that explicitly deals with the problem of localizing containers in a harbor is [37]. The authors describe a system, VAPS, that takes into account the physical characteristics of large objects as a way to define constraints useful for the purpose of localization: i) the metallic surface and the grid-like arrangement of containers cause a waveguide effect along some directions and a blocking effect along other directions; ii) objects are not dimensionless and cannot overlap. In VAPS, containers are equipped with two wireless devices (for the two horizontal axes), while communication between the devices located on the same container is achieved through wires. Each device is able to distinguish a (small) number of different RSS levels. Simulative results show that VAPS performs better than two competitors: an RSS-based method using a open-space propagation model and a hop-count based method. In the end, VAPS confirms the importance of the problem of container localization and the use of geometrical constraints as an effective technique over geometrically blind approaches. With respect to the technique described here, in [37] the analysis is limited to a bi-dimensional scenario and the presence of faults is not considered.

The problem of automatically identify and locate containers in the yard has been faced also in [38]. The proposed system, MOCONT, relies on GPS positioning acquired by reach-stackers that communicate the position of containers to a base station each time they are moved. The system also includes an inertial navigation system that, by using accelerometers, gyroscopes, and ground speed sensors, provides positioning information when the GPS system cannot operate (for example when the satellite signal is shielded by high container stacks). Container identification is performed through digital image analysis techniques.

Tracking of container position on a large scale can be also achieved using non-GPS technologies. In [39], the authors propose a tracking system that is based on the analysis of FM broadcast signal: a low-cost and low-power FM receiver, attached to containers, records the frequency spectrum and compare it to known data to determine the path of the container. The purpose of this approach is to overcome some of the limitations of GPS-based devices such as high power consumption, cost and the need of line-of-sight with satellites. However, the position is determined with a rather large error level (in the order of kilometers) and, in any case, its purpose is different from the approach proposed here, as it is aimed at tracking the movement of containers during ground transportation by means of trucks or trains.

The adoption of WSNs for container tracking and monitoring has been discussed also in [40]. The authors propose an architecture where sensor nodes are placed both inside and outside containers. The internal nodes are used to monitor the status of goods or to recognize some possible dangers (fire, water, etc.). Each container is also equipped with an external node, called container monitor, that is responsible of collecting the data coming from internal nodes and communicating with other monitors. Container monitors are supposed to have global connectivity, through GSM links, and be equipped with a GPS receiver. The architecture also includes the presence of a special node, with an unlimited power supply, that can be used to reduce the energy spent for communication by container monitors.

In [41] the authors designed a system to identify and localize containers. The system is based on a tablet PC equipped with a camera, a GPS unit and a digital compass. Image processing techniques are used to recognize the containers pointed by the camera, and an extended Kalman filter is used to fuse the data coming from the two sensors. The device is used by an operator that manually has to move within the yard and communicates with a database by means of a wireless connection. Thus, the purpose of the system is somehow different from the one proposed here that, as mentioned, aims at achieving automatic and and continuous monitoring of container position.

2.1.2 General constraints and challenges

Given the high complexity of localization, many researchers and scientists often make simplifying assumptions which do not enable to build a real working system. Such assumptions include: circular radio range, symmetric radio connectivity, additional hardware, no obstacles, line-of-sight, no multi-path, interference or flat terrain. Researchers have also to face the problem of finding a deployment method, a limited time for the localization algorithm to converge, presence of reference points, hardware required and energy consumption.

RSS measurements are relatively inexpensive, simple to implement in hardware but notoriously unpredictable [42]. To obtain a robust localization system, the sources of error given by RSS must be well understood. In fact, in real-world applications, RSS suffers from multi-path and path loss. In addition, RSS depends also on the calibration of both the transmitter and receiver.

Range and angle measurements used for localization are measured in a physical medium that introduces errors [42]. In general, these measurements are influenced by both time-varying errors and environment-dependent errors. Time-varying errors can be reduced by averaging multiple measurements over time. Environment dependent errors are the result of the physical arrangement of objects where the WSN is operating.

ToA is the measured time at which a RF signal first arrives at a receiver. The measured ToA is the time of transmission plus a propagation-induced time delay [42]. For acoustic propagation, 1 ms translates to 0.3 m, while for RF, 1 ns translates to 0.3m. The critical part is the ability of the receiver to accurately estimate the arrival time of the Line-of-Sight (LOS) signal. This estimation becomes more difficult in presence of additive noise and multi-path signals. A fundamental element is the clock synchronization among the sensors, so that the time delay is determined by subtracting the known transmit time from the measured ToA. In case of asynchronous sensor networks, a common practice is to use two-way ToA measurements. In this method, one sensor transmits a signal to a second sensor, which immediately replies with its own signal. At the first sensor, the measured delay between its transmission and its reception of the reply is twice the propagation delay plus a reply (known) delay internal to the second sensor. In contrast to ToA, Time-Difference-of-Arrival (TDoA) measurement does not depend on the clock bias of the transmitting sensor. It finds application in GPS and cellular localization. Under certain weak conditions, it has been shown that ToA with clock bias (unknown) is equivalent to TDoA.

AoA measurements provide localization information complementary to the above measurements [42]. It consist in calculating the direction to neighboring sensors rather than the distance to neighboring sensors. Measurements can be performed using a sensor array and array signal processing techniques at the sensor nodes. In this case, each sensor node is equipped with two or more individual sensors (microphones for acoustic signals or antennas for RF signals) whose locations with respect to an anchor node are known. The AoA is estimated from the differences in arrival times for a transmitted signal at each of the sensor array elements. Another way to perform AoA estimation includes the use of RSS ratio between two or more directional antennas located on the sensor. The antennas must be placed so that their main beams overlap and AoA is calculated from the ratio of their individual RSS values. Both AoA approaches require multiple antenna elements, which can contribute to sensor device cost and size. AoA measurements suffer from additive noise and multi-path.

2.2 Monitoring

Unlike localization, monitoring is a task strictly related to the application field. In our case, monitoring is used for monitoring of human movement analysis, in particular to detect falls among elderly. This section gives an overview of the fall problem as well as the techniques currently used and their limitations.

The problem with accidental falls among elderly people has massive social and economic impacts. Falls in elderly people are the main cause of admission and extended period of stay in a hospital. It is the sixth cause of death for people over the age of 65, the second for people between 65 and 75, and the first for people over 75. Among people affected by Alzheimer's Disease, the probability of a fall increases by a factor of three.

Elderly care can be improved by using sensors that monitor the vital signs and activities of patients, and remotely communicate this information to their doctors and caregivers. For example, sensors installed in homes can alert caregivers when a patient falls. Research teams in universities and industries are developing monitoring technologies for in-home elderly care. They make use of a network of sensors including pressure sensors on chairs, cameras, and RFID tags embedded throughout the home of the elderly people as well as in furniture and clothing, which communicate with tag readers in floor mats, shelves, and walls.

A fall can occur not only when a person is standing, but also while sitting on a chair or lying on a bed during sleep. The consequences of a fall can vary from scrapes to fractures and in some cases lead to death. Even if there are no immediate consequences, the long-wait on the floor for help increases the probability of death from the accident. This underlines the importance of real-time monitoring and detection of a fall to enable first-aid by relatives, paramedics or caregivers as soon as possible.

Monitoring the ADL is often related to the fall problem and requires a non-intrusive technology such as a wireless sensor network. An elderly with risk of fall can be instrumented with (preferably) one wireless sensing device to capture and analyze the body movements continuously, and the system triggers an alarm when a fall is detected. The small size and the light weight make the sensor network an ideal candidate to handle the fall problem.

The development of new techniques and technologies demonstrates that a major effort has been taken during the past 30 years to address this issue. However, researchers took many different approaches to solve the problem without following any standard testing guidelines. In some studies, they proposed their own guidelines.

There are three main categories of techniques based on the technology used:

- Vision-based.
- Environmental.
- Wearable.

A *Vision-based* approach uses fixed cameras that continuously record the movement of the patients. The acquired data is submitted to specific image algorithms that are able to recognize the pattern of a fall to trigger an alarm. Vision-based approaches can be classified as:

1. *Inactivity detection*, based on the idea that after a fall, the patient lies on the floor without moving.

2. *Body shape change analysis*, based on the change of posture after the fall.

3. *3D head motion analysis*, based on the monitoring the position and velocity of the head.

The main limits of this approach are the time and cost of installation, the limited space of application (only where there are the cameras) and privacy violation.

The use of *Environmental* devices is an approach based on the installation of sensors in the places to be monitored. When people interact with the environment, infrared or pressure sensors on the floor are able to detect a fall. The problem here is the presence of false-negatives, for example, a fall that occurs on a table is not detected.

Both Visual-based and Environmental device approaches require a pre-built infrastructure, and this enables their use in hospitals and houses, but it is hard to use them outdoor.

In the *Wearable* approach, one or more wearable devices are worn by the patient. They are usually equipped with movement sensors such as accelerometers and gyroscopes, whose values are transmitted via radio and analyzed. This solution offers advantages such as low installation cost (indoor and outdoor), small size and offers the possibility to also acquire physiological data (blood pressure, Electrocardiography (ECG), Electroencephalography (EEG) etc.). The wearable-device approach can be performed with wireless sensor networks.

2.2.1 Fall detection approaches

Many different approaches have been taken to solve the fall detection problem using accelerometers. The basic and trivial system uses a threshold to establish if a person falls, which is subject to many false positives. Some researchers have tried to introduce computationally-hard type of intensive algorithms but the goal has been always to find a trade-off between the system accuracy and the cost. There are two widely used evaluation parameters (described more in depth later): *Sensitivity*, that is the capacity to detect a fall, and *Specificity*, that is the capacity to avoid false positives. Intuitively it is the capacity to detect a fall only if it really occurs.

In [43] the authors used a two-level neural network algorithm to analyze the accelerations given by two sensors placed in distinct parts of the body. Such accelerations are translated into spatial coordinates and fed into the algorithm. The output of the system represents the probability that a fall is happening: if the probability is low, the system continues monitoring whereas if the probability is medium or high, the system generates an alarm unless the person presses a button.

In [44] the authors developed a system composed of a series of accelerometers, a processor and a wireless transceiver. The acquired acceleration data is constantly compared with some standard values. If there is a fall event, the processor sends an alarm signal to a remote receiver. A similar approach is given by [45] using a sensor module and an algorithm to detect posture, activity and fall. For long range communication with the base station, there are intermediate nodes that act as repeaters. The sensitivity was 93.2%.

In [46] the authors used an acoustic device on the rear side of the ear, to measure velocity and acceleration. Also [47] used a sensor on the head of the patient since it increases the accuracy of the detection.

The Inescapable Smart Impact detection System ISIS [48] used a sensor with an accelerometer and a smartphone as base station. Moving the processing to the smartphone extended the lifetime of the batteries and the usability of the sensor. They achieved 100% sensitivity with reduction in specificity.

Other methods are based on the body posture and use more than one sensor. Some researchers divided the human activities into two parts: static position and dynamic transition [49]. They used two sensors both with an accelerometer and a gyroscope, one placed on the chest and the other on the thigh. The gyroscope helped to decrease the false positives.

In [50] the authors used a sensor with two accelerometers, one orthogonal to the other and placed under the armpit. The fall is detected on the basis of the inclination of the chest and its velocity. The alarm is not raised if the patient presses a button on time, avoiding thus false alarms. An experimental evaluation showed levels of sensitivity and specificity equal to 81%. In a similar study researchers used a device with three different sensors for body posture detection, vibration detection and to measure vertical acceleration [51]. Data was processed by the base station. The sensitivity and specificity here were 85%.

Other researchers developed a real-time algorithm for automatic recognition of physical activities and their intensities [52]. They used five accelerometers placed on the wrist, the ankle, the upper arm, the upper thigh and the hip. In addition, they used a heart rate monitor placed on the chest. Trials have been conducted on 21 people for 30 different physical activities such as lying down, standing, walking, cycling, running and using the stairs. Data analyzed both in time and frequency domain were classified using the Naive Bayes classifier. Results showed an accuracy of 94.6% for a person using the training set of that person, whereas the accuracy was 56.3% using the training sets of all the other people.

Another research work exploited an accelerometer placed on the waist [53]. The device was so small that it fitted in a belt. The authors analyzed the duration, velocity, angle of a movement and its energy consumption to distinguish between activity and rest. The processing of the information was conducted by a base station. The authors used a threshold of 2.5G to detect a fall under the assumption that the subjects are not in good health and therefore unable to perform actions with acceleration above that threshold. This means that, to avoid false positives, they had to limit the type of subjects who can benefit from such a system.

In [54] the authors used a node placed on the chest featuring an accelerometer, a gyroscope, a tilt sensor, a processing unit and a Bluetooth transmitter. The accelerometer measured the kinetic force whereas the tilt sensor and the gyroscope estimated the body posture. The goal was to detect some activities of daily living and falls. The authors experimented on three people, aged over 26 years, studying the four activities: forward fall, backward fall, lateral fall and sit-stand. In this study, the system could distinguish between fall and daily activities. The accuracy of fall detection was 96.7%.

Recently, smartphones with embedded accelerometers have been used to act both as fall detector and as gateway to alert the caregivers [55, 56]. The problems associated with this approach are related to the device placement (in a fixed position or not) and to the short battery lifetime. Usually in these applications there is a trivial fall detection algorithm and to avoid false positives, the user should press a button to dismiss the alarm when there is no real fall.

2.2.2 General constraints and challenges

The review of the above proposed solutions shows some pitfalls for a real implementation. The system found more promising is the one that takes into account postures given by the accelerometers and gyroscopes to reduce false positives [49]. But the authors used two nodes and did not detect activities of daily living such a “falling” on a chair or a bed. The reported sensitivity is 92% and specificity 91%.

Hence the first challenge is to improve the performance of systems, to assist the patient only when there is a real fall. If we imagine to deploy the system in a hospital, it would be very annoying to run frequently to a patient because of false alarms.

The next challenge is to take into account the usability. The ideal system should be based on only one wearable sensor with small form factor, possibly placed in a comfortable place such as a belt. This may complicate the posture detection. Also time adaptation and invasiveness of the devices should be considered. Moreover the energy consumption must be low to extend the battery lifetime. This requires careful management of radio communications (the activity with the highest consumption of energy), flash storage and data sampling and processing. To support clinical requirements battery lifetime is a major concern: the minimum battery lifetime should be at least one day, in order to avoid stressing the caregivers with the tasks of recharging and replacing the devices, considering that longer the battery life better the continuity and the effectiveness of the system.

Chapter 3

Localization of shipping containers

In the previous chapters we have seen a complete, even though non exhaustive, overview of the wireless sensor network technology. Its inner flexibility, low cost and ease of deployment, enable the design of large scale systems, promising more efficient solutions for the existing applications and fostering the invention of new ones. Nevertheless, there are many non-functional issues that need to be addressed for every application. The issues include efficient energy management, fault tolerance, scalability, production costs, and the deployment of the nodes. In this chapter the application considered is a system for localization of shipping containers. It has been designed to work without the use of GPS and it includes the possibility to localize even in presence of faulty sensing nodes, a typical situation of a real world scenario.

Recall that range and angle measurements are not as accurate as expected due to the errors introduced by the physical medium. Unless we use an expensive hardware, the non-determinism introduced by the time and environment errors make the localization task hard to achieve. In fact, there are many techniques that depend on the working environment and time constraints. An alternative to this are range-free approaches, which never try to measure the exact distance, but rely on connectivity (proximity) information and provide a coarse-grained localization accuracy. The localization system here described uses sensor networks and can be found also in [57].

3.1 Problem description

Ports and terminals undergo a continuous increase in the level of traffic. Today the port of Singapore, one of the busiest ports in the world in terms of container handling, manages more than twenty million of shipping containers per year. For this reason, ports and terminals make use of automated container handling and transportation solutions, especially in countries with high labour costs [58]. Higher productivity has been achieved through advanced terminal layouts, more efficient IT-support and improved logistics control software systems [59]. Computers are employed to schedule and control different kinds of handling operations, such as identification [60] and tracking of containers, and their localization in the yard.

3.1.1 Shipping containers

Containers, also known as shipping containers, intermodal transport units or isotainers, are used for freight cargo transport on trucks, trains and ships. Their introduction has improved cargo shipping and has driven modifications to freight-moving standards: removable truck bodies or swap bodies have been forced into standard sizes and shapes, and their use across the globe has lessened the problems caused by incompatible rail gauge sizes of different countries.

The most widespread containers are those conforming to the ISO standard, whose measures have been accepted internationally: 8 feet (2.44 m) width, 8 feet and 6 inches (2.59 m) height, and two standard lengths of 20 and 40 feet (6.10 and 12.20 m). The standard also includes specific corners used to manage containers by means of cranes; the hardiness of corners and edges permits to arrange the containers in stacks, obtaining space benefits. Various container types are available for different needs: general purpose, high cube, temperature controlled (from -25 C to +25 C) reefer, open top, open side, and many others.

3.1.2 Container terminals and storage

Seaport container terminals can be described as open systems of material flow with two external interfaces: the quayside where loading and unloading of ships take place, and the land-side where containers are loaded and unloaded on/off trucks and trains [61]. Import as well as export containers are stored in stacks and divided into a number of blocks. This facilitates the decoupling of quayside and land-side operations.

After arrival at the terminal, the container is identified in order to obtain its major data such as content, destination, outbound vessel, shipping line. Then, it is picked up by internal transportation equipment and distributed to one of the storage blocks in the yard. Once the designated vessel is ready, the container is unloaded from the yard block and transported to the berth where quay cranes load the container onto the vessel at a pre-defined stacking position. A reverse order is followed to handle an import container.

Current computer-aided solutions include real-time assignment of transportation orders to vehicles, routing and scheduling of vehicle trips for transportation, assignment of storage slots to individual containers [62]. In general, a wireless communication system provides connectivity to vehicles and operators all over the terminal. GPS- and RFID-based solutions have been extensively adopted to achieve automatic localization and/or identification. GPS receivers are not directly installed on containers, but on top of the transport and stacking equipment. The position is measured, translated into yard coordinates and transmitted to a central system whenever a container is lifted or dropped. This way, database queries provide the geo-location of containers when needed. RFID technology enables a quick identification of containers, but it is less useful to determine their position. Also, RFID systems require a fixed or mobile infrastructure to read the tags, and the process, in many cases, includes human-driven or semi-automated operations.

3.2 Proposed approach

The localization technique described here enables the automated on-line discovery of the positions of containers in the yard. This is achieved by means of wireless sensor nodes, placed on containers, which cooperate to collect proximity information and communicate with a base station, where positions are computed.

3.2.1 Motivation

In general, shipping data about containers and their position within the yard are known in advance. However, such information is not always correct or completely up-to-date because of operational disturbances. For example, while in the yard, a container can be moved several times for content control, custom formalities, routing operations, etc. Thus, despite of currently available solutions, real-time identification and localization of containers are still error-prone activities and require human intervention to manage anomalous situations (for example, by physically searching the misplaced containers).

In these cases the RFID- and GPS-based techniques previously mentioned cannot provide a completely automated solution to the problem. Using the GPS, the position of a container can be indirectly determined from the position of the truck or quay crane when the container is lifted or dropped. But GPS receivers cannot be directly attached to containers for real-time identification and positioning, as this would prevent the receiver to be in line-of-sight with satellites when the container is not on the surface of a container stack. Instead, automatic localization in the yard can be achieved by equipping containers with smart wireless nodes and using a positioning scheme based on proximity information, as the one described here. At a glance, with the proposed approach wireless sensors detect neighbor containers and send proximity information to a base station, where it is combined with geometrical constraints to determine the relative positions of containers. In a first step, a simple algorithm places those containers whose position is not ambiguous (the strawman approach) [5]. In a second step, missing proximity information due to faulty nodes is tolerated by modeling geometrical constraints as an integer linear programming problem.

Obviously this technique, based on the idea of embedding intelligence directly on containers, involves some costs. Nevertheless, it is important to notice that: i) the cost of a single container is in the order of thousands USDs, thus adding equipment increases the total cost of only a small fraction; ii) beside localization, other orthogonal problems can be faced and solved through the use of intelligent devices, examples include security [63], supervision [64] and monitoring [65]. Moreover, from a more distant perspective, the authors believe that the Internet of Things (IoT) will become, sooner or later, a reality [66]. According to the IoT vision, the world of physical objects becomes seamlessly integrated into the information network and participates actively in business, information and social processes, where smart things communicate among themselves and with the surrounding environment. Starting from this assumption, it is possible to imagine that, in a not too distant future, also the ordinary shipping containers will be enhanced with computing and communication capabilities. Then a system like the one proposed here can be integrated with small additional costs.

3.2.2 Assumptions and definitions

It is supposed that containers are not turned upside down, and that the long edges of containers are always parallel with each others. Containers, that are obviously parallelepipedons, are then supposed to be positioned within a three dimensional grid. These hypothesis are realistic and not over-restrictive: the placement of containers in a yard usually follows the grid model, as it maximizes the usage of the available surface. Even though the proposed approach can be extended to cope with different container types, for the sake of simplicity in this article all the containers are assumed to have the same size.

Two containers are defined as *contiguous* if they are located so that at least one edge of the first container is contiguous to an edge of the second container. Two containers are said *adjacent* if they are located face to face (i.e., four edges are contiguous). A *group* is a set of containers where every container is adjacent to, at least, another container.

More formally, given a reference system, the position of a container can be expressed through its three dimensional coordinates. Given a container A, its coordinates are indicated as $x(A)$, $y(A)$, $z(A)$. Also, since containers are placed according to a grid model, $x(A)$, $y(A)$, and $z(A)$ are three integer values that specify the element of the grid where A is placed. Without loss of generality, it can be supposed that the coordinates of the containers are non-negative, thus $x(A) \in \{0, \ldots, M_x - 1\}$, $y(A) \in \{0, \ldots, M_y - 1\}$, and $z(A) \in \{0, \ldots, M_z - 1\}$ with M_x, M_y, M_z positive values.

According to this notation, two containers A and B are contiguous if

$$\begin{cases} |x(A) - x(B)| \leq 1, \\ |y(A) - y(B)| \leq 1, \\ |z(A) - z(B)| \leq 1, \\ |x(A) - x(B)| + |y(A) - y(B)| + |z(A) - z(B)| \leq 2, \end{cases}$$

and they are adjacent if

$$|x(A) - x(B)| + |y(A) - y(B)| + |z(A) - z(B)| = 1$$

Given that a container cannot be turned upside down and that containers are placed according to a grid model where elements are parallelepipedons, a container can be oriented only in two ways. Thus, the orientation of a container A can be expressed as $o(A)$, where $o(A) \in \{0, 1\}$.

If the containers are organized in the yard as separated groups, the localization procedure described in the following can be applied separately to each group.

3.2.3 Container equipment

Every container is equipped with wireless nodes that $i)$ detect the presence of nodes belonging to other containers; $ii)$ calculate their relation of proximity on the base of measures of the Received Signal Strength (RSS) of the wireless communication channel.

As known, localization techniques based on RSS are characterized by limited accuracy, in particular when used in the presence of obstacles [67, 68]. To overcome this limitation, the system has been conceived according to the following guidelines:

- the proximity relation between two nodes is modeled as a value in a binary domain: *close/far*;
- the proximity relation between nodes belonging to different containers can be used to infer contiguity and/or adjacency conditions between such containers.

An obvious placement strategy can be based on the use of six wireless nodes placed at the center of the container faces. In this way, proximity between two nodes can be easily translated into a condition of adjacency between the corresponding faces, and thus between containers.

Unfortunately, an initial set of experiments showed that wireless communication between nodes of the same container was prevented by its metallic nature. Thus, in order to guarantee a line of sight between nodes belonging to the same container, and considering that there is always a small amount of space between adjacent containers, the nodes have been moved to the edges.

Figure 3.1 shows the placement of the nodes on a container. Four sensors are placed on the edges of the upper face, whereas two other sensors

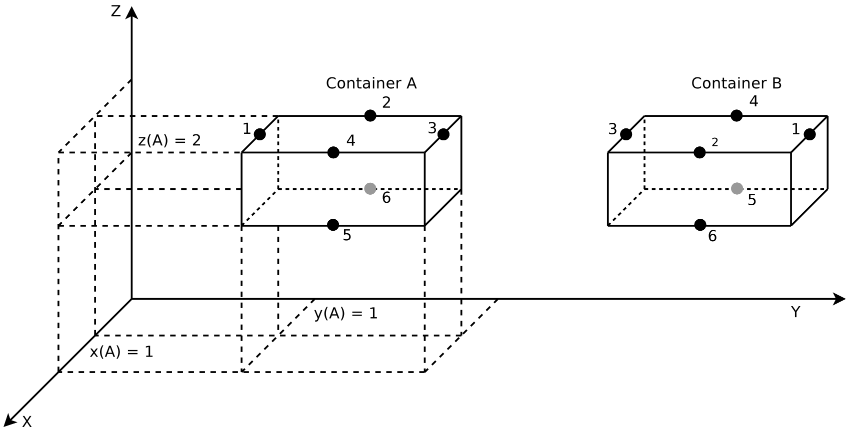


Figure 3.1: Position of nodes, reference system, and orientation.

are placed on the edges of the bottom face of the container. The unsymmetrical distribution of nodes with respect to the horizontal plane is justified by the fact that containers cannot be turned upside down. Every node is identified by a unique node identifier, called *NodeId*. The NodeId is composed of two parts: the *ContainerId*, that identifies the container a node belongs to, and the *EdgeId*, that identifies the edge where the node is placed.

Figure 3.1, besides the EdgeId of nodes, also shows the direction of the axes of the reference system and two containers with different orientations ($o(A) = 1$ and $o(B) = 0$).

With this second placement strategy, proximity relations between nodes can still be used to derive contiguity information about the edges of containers, and in turn adjacency between containers. To this purpose, two nodes are considered close if they belong to contiguous edges, far in all other cases.

3.2.4 Detection of proximity information between nodes

Given two containers *A* and *B*, there are, in theory, 36 proximity relations between the six nodes of *A* and the six nodes of *B*. Actually, because of the storage rules previously mentioned (grid model where long edges are always parallel), and because of the position of nodes on the edges of

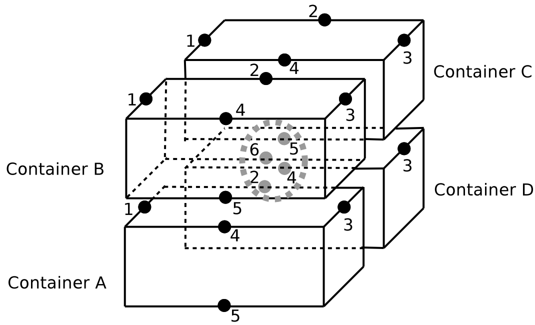


Figure 3.2: Every node can be close to up to three other nodes.

containers, the number of significant proximity relations is equal to 20. For example, node 1 of container A cannot be close to nodes 2, 4, 5, 6 of container B, as it would violate the parallelism of long edges of the two containers. Similarly, node 3 of container A cannot be close to nodes 2, 4, 5, 6 of container B, and nodes 2, 4, 5, 6 of container A cannot be close to nodes 1 and 3 of container B. In other words, the grid model defines compatibility rules between edges and, in turn, between nodes.

Every node maintains a table that contains the NodeId of nodes in close proximity (NCP table). Because of the storage rules and grid model, every node can be in close proximity with up to three other nodes, as shown in Figure 3.2. Thus, the NCP table can contain a maximum of three entries. Also note that, from the nodes that are close, every node is from a different container.

To determine its proximity with respect to other nodes, every node periodically broadcasts a beacon. The beacon contains the NodeId of the sender. The emission of the beacons is performed with the same period T for all the nodes in the network, but there is no synchronization among different nodes (T is equal to $30s$ in the implemented prototype). At the same time, every node listens to the radio channel for the possible reception of beacons generated by other nodes. On receiving a beacon, the NodeId of the sender is analyzed, and it is checked if the EdgeId of the sender belongs to the set of edges compatible with the edge of the receiver (i.e., the proximity relation between the sender and the receiver is significant). If the edges are not compatible, the beacon is not further analyzed. For example, if a node with EdgeId equal to 1 receives a beacon from a node with EdgeId equal to 1 or 3, then the beacon is further analyzed as described in the following, otherwise the beacon is discarded.

If the edges are compatible, the receiver node compares the RSS of the received beacon against a fixed threshold (experiments in a real setting have been carried out to tune this threshold to reflect an approximate distance of $1m$). If the RSS is greater than the threshold, the receiver inserts the NodeId of the sender node within its NCP table.

To improve the stability of the system and make it tolerant to possible packet losses, a node is removed from the NCP table only if no beacons are received from that node during a time longer than kT, where k is a configurable parameter (set to 3 in the implemented prototype). Because of the movement of containers, it may happen that a node receives the beacons from its new neighbors while its NCP table still contains the entries of its old neighbors (that are removed after kT time). To manage this situation the following policy is adopted: In case a beacon coming from a new node is received, let e be the EdgeId of the new node and let be S the subset of entries of the NCP table of the receiver that are compatible, in terms of edges, with e; if S already contains the maximum amount of entries (i.e. 3 if $e \in \{2, 4, 5, 6\}$ or 1 if $e \in \{1, 3\}$), the oldest entry in S is discarded and an entry containing the information coming from the new node is added to the table. In other words, with scope limited to sets of compatible edges, recent information is preferred to old data.

3.2.5 Collection and storage of proximity information

Each node must send the content of its NCP table to the base station. This operation, called *collection*, runs with period hT, where h is a configurable parameter (equal to k in the implemented prototype). Transfer of data to the base station is achieved through the standard Collection Tree Protocol (CTP), provided by the TinyOS operating system (TinyOS has been used as the base platform for the implementation of the system, as described in Section 3.7). In CTP, transfer of data is performed through a multi-hop routing tree that converges at the base station. Every node takes part in the forwarding activity and routing is based on a shortest-path algorithm (together with mechanisms that take into account the quality of a link). The tree is built and maintained independently with respect to the operations aimed at the detection of proximity information, and its topology does not depend on the position of containers (Figure 3.3). If a node gets broken, that node will be excluded from the routing tree and it will not participate in the forwarding activity (CTP detects broken links and selects another node as the next hop). This is not a problem until the number of broken nodes gets so high that

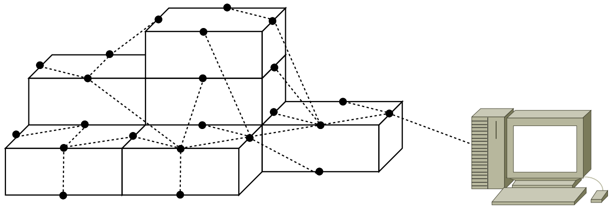


Figure 3.3: The routing tree is independent from the position of containers.

large parts of the network become unreachable.

It is important to notice that the focus here concerns the localization mechanisms and that transfer of information from nodes to the base station can be achieved through standard routing protocols. Thus, any routing protocol that is able to provide connectivity to the base station and that is able to dynamically re-arrange the routes in case of faults could be suitable (literature about routing protocols for wireless sensor networks is quite abundant, see [69] for a survey).

Since the NCP table has a maximum of three entries, it is possible to insert all its data in a packet (NCP packet) of fixed size and format. In more detail, the packet contains the NodeId of the sender and the NodeIds of all the nodes in its close proximity. If the number of nodes in close proximity is lower than three, the remaining fields are set to zero.

Table 3.1 shows the content of a NCP packet that represents the relations of close proximity of node 2 of container A with nodes 6, 5, 4 of containers B, C, D respectively, as also depicted in Figure 3.2 (NodeIds are expressed in the form ContainerId, EdgeId).

$A, 2$	$B, 6$	$C, 5$	$D, 4$

Table 3.1: Example of a NCP packet.

3.2.6 Representation of proximity information at the base station

The base station maintains a set $\mathcal{C}$ that includes the ContainerIds of all known containers. The set is initially empty and it is managed as follows: i) each time a NCP packet is received, all the NodeIds contained in the packet are extracted; $ii)$ the ContainerIds of the extracted NodeIds are added to $\mathcal{C}$; $iii)$ if a ContainerId is already included in $\mathcal{C}$, the corresponding element is refreshed; $iv)$ elements of $\mathcal{C}$ are removed from the set when not refreshed for a time equal to z times kT (with z set to 3 in the implemented prototype).

Let the relation of close proximity between node *i* of container A and node *j* of container B be represented as $R_{A,B}(i,j)$. The base station maintains also a set $\mathcal{R}$ of relations of close proximity, initially empty, and managed as follows: $i)$ each time a NCP packet is received, the relations of close proximity derived from the packet content are added to $\mathcal{R}$; $ii)$ if a relation of close proximity is already included in $\mathcal{R}$, the corresponding element is refreshed; $iii)$ elements of $\mathcal{R}$ are removed from the set when not refreshed for a time equal to z times kT.

Given the symmetrical nature of proximity relations, every time $R_{A,B}(i,j)$ is inserted into $\mathcal{R}$, then $R_{B,A}(j,i)$ is inserted as well. This helps to make the system more resilient to the possible loss of packets during the collection phase.

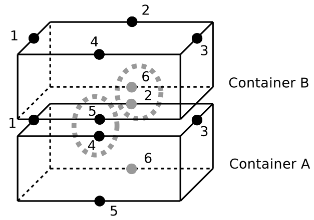
3.2.7 Inferring the position of containers from the relations of close proximity

As mentioned, two containers are adjacent if they are located face to face. Thus given two adjacent containers A and B, the latter can be adjacent to one of the six faces of A (and viceversa). Since every container can have two different orientations and considering that the possible positions of B with respect to A are six, it is possible to distinguish a total of twelve forms of adjacency: six if B has the same orientation of A and six if they have opposite orientation.

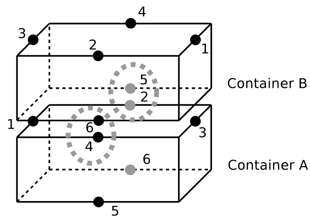
Figure 3.4 shows two adjacent containers when they have the same and opposite orientation. In the first case, the relations of close proximity that are generated are $R_{A,B}(4,5)$, $R_{A,B}(2,6)$, $R_{B,A}(5,4)$, and $R_{B,A}(6,2)$, while in the second case are $R_{A,B}(4,6)$, $R_{A,B}(2,5)$, $R_{B,A}(6,4)$, and $R_{B,A}(5,2)$.

Thus, if the coordinates and the orientation of a container A are known, it is possible to infer the coordinates and the orientation of a container B by using the relations of close proximity included in $\mathcal{R}$. Table 3.2 contains twelve rules that can be used to compute the coordinates and the orientation of B with respect to A.

A rule can be applied if the correspondent condition is true. Figure 3.5



(a) Same orientation



(b) Opposite orientation

Figure 3.4: Containers with the same and opposite orientation.

#	Condition	Orientation of B	Coordinates of B (equal to A if not specified)
1	$R_{A,B}(1,3) \in \mathcal{R}$	$o(B) = o(A)$	$y(B) = y(A) - 2\,o(A) + 1$
2	$R_{A,B}(1,1) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$y(B) = y(A) - 2\,o(A) + 1$
3	$R_{A,B}(3,1) \in \mathcal{R}$	$o(B) = o(A)$	$y(B) = y(A) + 2\,o(A) - 1$
4	$R_{A,B}(3,3) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$y(B) = y(A) + 2\,o(A) - 1$
5	$R_{A,B}(2,4) \in \mathcal{R} \vee R_{A,B}(6,5) \in \mathcal{R}$	$o(B) = o(A)$	$x(B) = x(A) - 2\,o(A) + 1$
6	$R_{A,B}(2,2) \in \mathcal{R} \vee R_{A,B}(6,6) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$x(B) = x(A) - 2\,o(A) + 1$
7	$R_{A,B}(4,2) \in \mathcal{R} \vee R_{A,B}(5,6) \in \mathcal{R}$	$o(B) = o(A)$	$x(B) = x(A) + 2\,o(A) - 1$
8	$R_{A,B}(4,4) \in \mathcal{R} \vee R_{A,B}(5,5) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$x(B) = x(A) + 2\,o(A) - 1$
9	$R_{A,B}(2,6) \in \mathcal{R} \wedge R_{A,B}(4,5) \in \mathcal{R}$	$o(B) = o(A)$	$z(B) = z(A) + 1$
10	$R_{A,B}(2,5) \in \mathcal{R} \wedge R_{A,B}(4,6) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$z(B) = z(A) + 1$
11	$R_{A,B}(5,4) \in \mathcal{R} \wedge R_{A,B}(6,2) \in \mathcal{R}$	$o(B) = o(A)$	$z(B) = z(A) - 1$
12	$R_{A,B}(5,2) \in \mathcal{R} \wedge R_{A,B}(6,4) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$z(B) = z(A) - 1$

Table 3.2: Coordinates and orientation of a container B with respect to a container A.

shows a case where, since $R_{A,B}(1,3) \in \mathcal{R}$, rule 1 can be used to infer the coordinates and orientation of B with respect to A. Container B has the same x and z coordinates of A and same orientation, whereas the coordinates $y(A)$ and $y(B)$ are related as follows: if $o(A) = 1$ then $y(B) = y(A) - 1$, otherwise $y(B) = y(A) + 1$ (Figure 3.5 depicts the case where $o(A) = 1$). These relations can be merged in the equation $y(B) = y(A) - 2\,o(A) + 1$.

Figure 3.6 shows another example where rule 5 can be used. This rule can be applied if at least one relation of proximity out of two (logic *OR*) is in $\mathcal{R}$ ($R_{A,B}(2,4)$ or $R_{A,B}(6,5)$). Container B has the same y and z coordinates of A and the same orientation, whereas $x(B) = x(A) - 2\,o(A) + 1$. In this case the system exhibits a basic form of fault tolerance: even if one of the two relations of proximity is not present in $\mathcal{R}$, because of faulting nodes or packet losses, it is still possible to determine the position of a container B. Similar considerations can be drawn about rules 6, 7, and 8.

In other cases, a single relation of proximity is not sufficient to resolve the possible ambiguities and to infer the position and orientation of B with respect to A. Figure 3.7 shows one of such cases: if $R_{A,B}(2,6) \in$

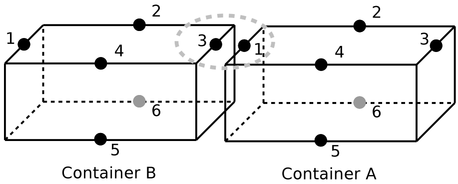


Figure 3.5: Rule 1 can be applied to determine the position of B if the position of A is known.

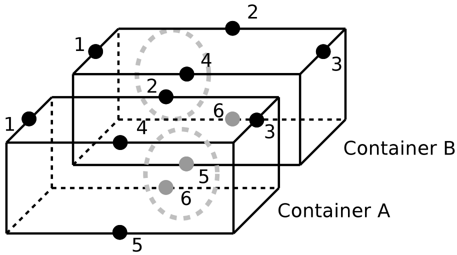
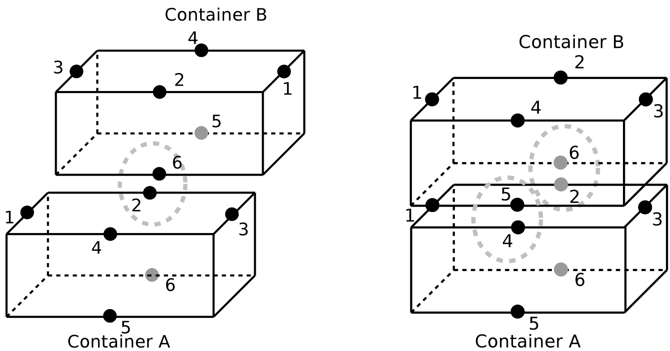


Figure 3.6: One of the two relations of close proximity is sufficient to determine the position and orientation of B with respect to A.



(a) (b)

Figure 3.7: A single relation of close proximity is not sufficient to infer the position and orientation of B with respect to A.

$\mathcal{R}$ then both the solutions depicted in Figure 3.7 are possible. Only the presence in $\mathcal{R}$ of $R_{A,B}(4,5)$ (or its symmetrical $R_{B,A}(5,4)$) can resolve the ambiguity and, in that case, the position of B can be computed through rule 9. This also explains the presence of the logic *AND* in rule 9. Similar considerations can be made for rules 10, 11, and 12.

3.3 The strawman approach

This section describes a simple approach for the localization of containers (the strawman approach) [5]. Then, in the following sections, some problems of the strawman approach are discussed and solved through Integer Linear Programming (ILP) techniques.

The localization procedure begins its execution with a container with known position and orientation. Then it finds all the adjacent containers by examining the content of $\mathcal{R}$ and uses the rules shown in Table 3.2 to compute the position of adjacent containers and their orientation. The same operations are repeated for all the containers whose position and orientation has been determined, until all the containers have been localized.

More in detail, for every container the system must store its identifier, its coordinates and its orientation. Thus, for every container A it is possible to define a tuple

$$\tau(A) := \{A, x(A), y(A), z(A), o(A)\}$$

that contains such information.

Besides $\mathcal{C}$ and $\mathcal{R}$ previously introduced, the localization procedure makes use of the following sets:

- $\mathcal{K}$: the set of known containers at the time the localization procedure is executed. It is initialized with the value of $\mathcal{C}$.
- $\mathcal{P}$: the set of tuples of containers whose position and orientation have been computed by the algorithm.
- $\mathcal{D}$: the set of tuples of containers whose neighbors still have to be discovered.

Initially, $\mathcal{P}$ and $\mathcal{D}$ are empty. The algorithm starts performing the following preliminary operations: it inserts in both $\mathcal{P}$ and $\mathcal{D}$ the tuple of a container I with known position and orientation. Then, it periodically

I is a container with known position and orientation
$\mathcal{K} \leftarrow \mathcal{C} - \{I\}$
$\mathcal{D} \leftarrow \{\tau(I)\}$
$\mathcal{P} \leftarrow \{\tau(I)\}$
for all $\tau(A) \in \mathcal{D}$ **do**
  $\mathcal{D} \leftarrow \mathcal{D} - \{\tau(A)\}$
  for all $B \in \mathcal{K}$ **do**
    for $i = 1$ to 12 **do**
      if Rule i can be applied **then**
        Compute coordinates and orientation of B
        $\mathcal{P} \leftarrow \mathcal{P} \cup \{\tau(B)\}$
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{\tau(B)\}$
        $\mathcal{K} \leftarrow \mathcal{K} - \{B\}$
        break
      end if
    end for
  end for
end for
return $\mathcal{P}, \mathcal{K}$

Figure 3.8: Pseudo-code of the localization procedure in the strawman approach.

executes the following actions: it extracts the tuple of a container A from $\mathcal{D}$ and for each container B that is contiguous to A (i.e, $R_{A,B}(i,j) \in \mathcal{R}$, for some (i,j)), it verifies if B is adjacent to A. In such case it computes the coordinates and orientation of B using the rules previously defined, $\tau(B)$ is updated and inserted into both $\mathcal{D}$ and $\mathcal{P}$ whereas B is removed from $\mathcal{K}$. If B is not adjacent to A, it is simply ignored. Once all the containers that are contiguous to A have been checked, the algorithm starts again by extracting another element of $\mathcal{D}$ and performs the same operations. The algorithm stops when $\mathcal{D} = \emptyset$ and returns the sets $\mathcal{P}$ and $\mathcal{K}$: $\mathcal{P}$ contains the identifier, position and orientation of all localized containers, whereas $\mathcal{K}$ contains the identifier of all known containers whose position and orientation cannot be computed. If there is no missing information about the set of adjacent containers under observation, because of faults or lost packets, then all the containers are localized. The pseudo-code of the algorithm is shown in Figure 3.8.

3.4 Localization by means of Integer Linear Programming

The strawman approach does not guarantee the localization of all containers in case of node faults or lost packets (in general, in case of incomplete proximity information in $\mathcal{R}$). However, the redundancy of contiguity information of a group of containers can be used to tolerate faults and localize a larger number of containers. This can be done by modeling localization as an ILP problem. The solutions of the ILP problem provide a tuple for each container which is compatible with the tuples of all the other containers and with the proximity information.

3.4.1 Variables and geometrical constraints of the ILP problem

The variables of the ILP problem are the coordinates $x(A)$, $y(A)$, $z(A)$, and the orientation $o(A)$ of each container $A \in \mathcal{C}$. In addition to the previous variables, a new variable $p(\tau(A))$ is introduced for each tuple $\tau(A)$:

$$p(\tau(A)) = \begin{cases} 1 & \text{if the coordinates and the orientation} \\ & \text{ of } A \text{ are those contained in } \tau(A) \\ 0 & \text{otherwise.} \end{cases}$$

The variables x, y, z, o, p are linked by geometrical and operational constraints as follows:

$$\sum_{i=0}^{M_x-1} \sum_{j=0}^{M_y-1} \sum_{k=0}^{M_z-1} \sum_{l=0}^{1} i\, p(A, i, j, k, l) = x(A) \qquad \forall\, A \in \mathcal{C},$$

$$\sum_{i=0}^{M_x-1} \sum_{j=0}^{M_y-1} \sum_{k=0}^{M_z-1} \sum_{l=0}^{1} j\, p(A, i, j, k, l) = y(A) \qquad \forall\, A \in \mathcal{C},$$

$$\sum_{i=0}^{M_x-1} \sum_{j=0}^{M_y-1} \sum_{k=0}^{M_z-1} \sum_{l=0}^{1} k\, p(A, i, j, k, l) = z(A) \qquad \forall\, A \in \mathcal{C},$$

$$\sum_{i=0}^{M_x-1}\sum_{j=0}^{M_y-1}\sum_{k=0}^{M_z-1}\sum_{l=0}^{1} l\, p(A,i,j,k,l) = o(A) \qquad \forall\, A \in \mathcal{C}.$$

Moreover, a unique tuple is associated to each container, therefore:

$$\sum_{i=0}^{M_x-1}\sum_{j=0}^{M_y-1}\sum_{k=0}^{M_z-1}\sum_{l=0}^{1} p(A,i,j,k,l) = 1 \qquad \forall\, A \in \mathcal{C},$$

and, at the same time, each location of the yard can host at most one container:

$$\sum_{A\in\mathcal{C}}\sum_{l=0}^{1} p(A,i,j,k,l) \leq 1 \qquad \forall\, (i,j,k).$$

Reduction of problem size. The size of the ILP problem can be reduced by using, as pre-processing phase, the procedure described as the straw-man approach. In this way, it is possible to localize the subset of containers whose position and orientation can be unambiguously determined (coordinates and orientation are contained in the tuples of the $\mathcal{P}$ set). For every container A such that $\tau(A) \in \mathcal{P}$, the corresponding coordinates and orientation are considered as known values.

3.4.2 Additional constraints derived from the relations of close proximity

Additional constraints to the ILP problem can be inferred from $\mathcal{R}$. Rules 9–12 of Table 3.2 express the position of B with respect to A, on the base of their adjacency relation. However, such rules can be re-written to express the position of B with respect to A on the base of their contiguity relation. For example, rule 9 of Table 3.2 can be split into rules 9a and 9b of Table 3.3, where the proximity relations are taken into account separately. Orientation is not considered in Table 3.3 as it cannot be uniquely determined. Rule 9a provides only partial information about the position of B:

$$\begin{cases} x(B) = x(A) - o(A) + o(B) \\ y(B) = y(A) \\ z(B) = z(A) + 1 \end{cases}$$

since $x(B)$ is a function of the orientation of B, that is unknown and can-

#	Condition	Coordinates of B (equal to A if not specified)
9a	$R_{A,B}(2,6) \in \mathcal{R}$	$x(B) = x(A) - o(A) + o(B)$ $z(B) = z(A) + 1$
9b	$R_{A,B}(4,5) \in \mathcal{R}$	$x(B) = x(A) + o(A) - o(B)$ $z(B) = z(A) + 1$
10a	$R_{A,B}(2,5) \in \mathcal{R}$	$x(B) = x(A) - o(A) - o(B) + 1$ $z(B) = z(A) + 1$
10b	$R_{A,B}(4,6) \in \mathcal{R}$	$x(B) = x(A) + o(A) + o(B) - 1$ $z(B) = z(A) + 1$
11a	$R_{A,B}(5,4) \in \mathcal{R}$	$x(B) = x(A) + o(A) - o(B)$ $z(B) = z(A) - 1$
11b	$R_{A,B}(6,2) \in \mathcal{R}$	$x(B) = x(A) - o(A) + o(B)$ $z(B) = z(A) - 1$
12a	$R_{A,B}(5,2) \in \mathcal{R}$	$x(B) = x(A) + o(A) + o(B) - 1$ $z(B) = z(A) - 1$
12b	$R_{A,B}(6,4) \in \mathcal{R}$	$x(B) = x(A) - o(A) - o(B) + 1$ $z(B) = z(A) - 1$

Table 3.3: Contiguity rules for a container B with respect to a container A.

not be established through the single proximity relation $R_{A,B}(2,6)$. Obviously, if also rule 9b can be applied, the resulting system of equations can be solved and the result is the one expressed by rule 9.

Similar considerations can be made about rules 10, 11, and 12, that can be replaced by the couples of rules {10a, 10b}, {11a, 11b} and {12a, 12b}.

3.5 ILP-based localization

The localization procedure of the strawman approach returns two sets: $\mathcal{P}$, that contains the tuples of all the containers that have been successfully localized, and $\mathcal{K}$, that contains the identifiers of known containers whose position is still undetermined. If $\mathcal{K}$ is empty, all the containers have already been localized and the algorithm terminates. When the set $\mathcal{K}$ is nonempty, a larger number of containers can be localized by applying an algorithm which iteratively solves an ILP problem with the constraints described in Section 3.4 and with a suitable objective function.

At the beginning, the algorithm finds any solution satisfying the constraints described in Section 3.4. Such solution gives a tuple (i.e. position and orientation) for each container of set $\mathcal{K}$. However, such a solution is not sufficient to localize the containers of $\mathcal{K}$, because there could be different solutions satisfying the constraints. On the other hand, we accept as localized a container only if it has a unique possible position and orientation. Thus, we have to verify if the obtained tuples are unique. This can be done by solving a suitable ILP problem with the same set of constraints. An objective function is added to such model whose aim is to maximize the number of containers which could have a different tuple from the initial one. If such a number is equal to zero, all the containers are localized. Otherwise, the set of containers can be divided into two subsets. Some of them keep the same tuple as in the initial solution and therefore we have to further verify the uniqueness of their tuple. Others change their tuple and therefore correspond to containers which can not be localized. The process is iterated. In the following the ILP-based algorithm reported in Figure 3.9 is described.

The content of $\mathcal{P}$ is used to assign a value to the variables corresponding to the containers already localized, and a feasible layout for all the containers is computed (see line 3). In other words, the algorithm finds suitable initial values $\bar{p}$, for the variables p, so that the constraints given by $\mathcal{P}$, the proximity relations contained in $\mathcal{R}$, and the geometrical constraints are satisfied. Moreover for each container $A \in \mathcal{C}$ we denote by $\bar{\tau}(A)$ the tuple such that $\bar{p}(\bar{\tau}(A)) = 1$. Then, an auxiliary set $\mathcal{Q}$ is initialized with the value of $\mathcal{K}$ (line 4). The set $\mathcal{Q}$ represents the set of containers for which the uniqueness of their tuples must be verified. Among all the feasible layouts of containers, the ILP-based algorithm looks for the one where the number of containers whose tuple is equal to the initial one is minimum (which corresponds to maximize the number of containers which have a different tuple from the initial one). In other words, the following ILP problem is solved:

$$\min \sum_{A \in \mathcal{Q}} p(\bar{\tau}(A)),$$

subject to the constraints given by $\mathcal{P}$, the geometrical constraints, and the relations of close proximity (line 6). The value of the objective function represents the total number of containers which have the same tuple as the initial solution. If this minimum value is equal to the cardinality $|\mathcal{Q}|$ of the set $\mathcal{Q}$, then no container in $\mathcal{Q}$ can have a tuple that is different from the initial one, i.e. all the containers in $\mathcal{Q}$ are unambiguously localized (lines 7-8). Otherwise, each container A having a tuple $\tau_{opt}(A)$ that is different from the initial one $\bar{\tau}(A)$ cannot be localized; hence the set $\mathcal{Q}$ is updated by removing such containers (lines 10-14) and the ILP problem is solved again. When the procedure completes, the set $\mathcal{Q}$ contains all the containers which can be localized and their tuples are the ones specified by the initial solution; the containers in $\mathcal{Q}$ are removed from $\mathcal{K}$ and their tuples are inserted in $\mathcal{P}$ (lines 17-20). The ILP-based algorithm returns the set $\mathcal{P}$ of tuples of all the containers which can be localized, and the set $\mathcal{K}$ which contains the containers which cannot be localized given the

Consider $\mathcal{P}$ and $\mathcal{K}$ returned by the strawman approach.
if $\mathcal{K} \neq \emptyset$ **then**
  Find $\bar{p}$ satisfying the constraints given by $\mathcal{P}$, geometrical constraints, and $\mathcal{R}$.
  Let $\bar{\tau}(A),\ A \in \mathcal{C}$ be the tuple such that $\bar{p}(\bar{\tau}(A)) = 1$
  $\mathcal{Q} \leftarrow \mathcal{K}$
  while $\mathcal{Q} \neq \emptyset$ **do**
    Find p_{opt} that solves the following ILP problem:

$$v = \min \sum_{A \in \mathcal{Q}} p(\bar{\tau}(A))$$

    subject to the constraints given by $\mathcal{P}$, geometrical constraints, and $\mathcal{R}$.
    Let $\{\tau_{opt}(A),\ A \in \mathcal{C}\}$ be the set of tuples such that $p_{opt}(\tau_{opt}(A)) = 1$
    if $v = |\mathcal{Q}|$ **then**
      break
    else
      for all $A \in \mathcal{Q}$ **do**
        if $\tau_{opt}(A) \neq \bar{\tau}(A)$ **then**
          $\mathcal{Q} \leftarrow \mathcal{Q} - \{A\}$
        end if
      end for
    end if
  end while
  for all $A \in \mathcal{Q}$ **do**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{\bar{\tau}(A)\}$
    $\mathcal{K} \leftarrow \mathcal{K} - \{A\}$
  end for
end if
return $\mathcal{P}, \mathcal{K}$

Figure 3.9: Pseudo-code of the ILP-based algorithm.

partial information (line 22). It is worth noting that the procedure always provides the tuples of all containers which can be unambiguously localized based on the available information.

As an example of the proposed methodology, let six containers be placed as depicted in Figure 3.10(a), while Figure 3.10(b) shows the X-Z plane and the working nodes. Figure 3.10(c), for the sake of clarity, represents the same information in a different way: containers are depicted as circles and the relations of close proximity are depicted as dashed lines.

It is assumed that A is a container with known position and orientation. The localization procedure of the strawman approach is not able to localize other containers, thus produces only the following result:

$$\mathcal{P} = \{(A, 0, 0, 0, 1)\}.$$

Containers in the set $\mathcal{K} = \{B, C, D, E, F\}$ are not localized because no one of the rules of Table 3.2 can be applied. Subsequently, the ILP-based algorithm starts with a feasible layout $\bar{\tau}$ of the containers as the one shown in the figure and sets $\mathcal{Q} = \mathcal{K}$. Then, the ILP problem is solved and the solution is $v = 3 < |\mathcal{Q}|$ with

$$\tau_{opt}(B) = \{(B, 2, 0, 0, 0)\} \neq \bar{\tau}(B)$$

$$\tau_{opt}(C) = \{(C, 1, 0, 0, 0)\} \neq \bar{\tau}(C)$$

$$\tau_{opt}(D) = \bar{\tau}(D)$$

$$\tau_{opt}(E) = \bar{\tau}(E)$$

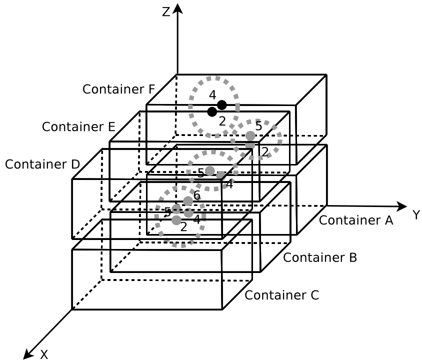
$$\tau_{opt}(F) = \bar{\tau}(F).$$

The ILP problem is solved again with $\mathcal{Q} = \{D, E, F\}$ and in this case $v = 3 = |\mathcal{Q}|$, so the algorithm terminates. The result of the algorithm is

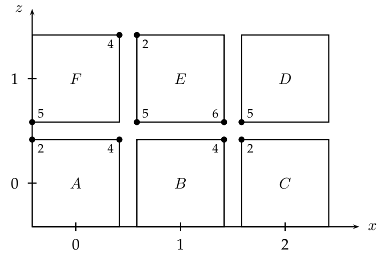
$$\begin{aligned}\mathcal{P} = \{&(A, 0, 0, 0, 1), (D, 2, 0, 1, 1),\\ &(E, 1, 0, 1, 1), (F, 0, 0, 1, 1)\},\end{aligned}$$

$$\mathcal{K} = \{B, C\}.$$

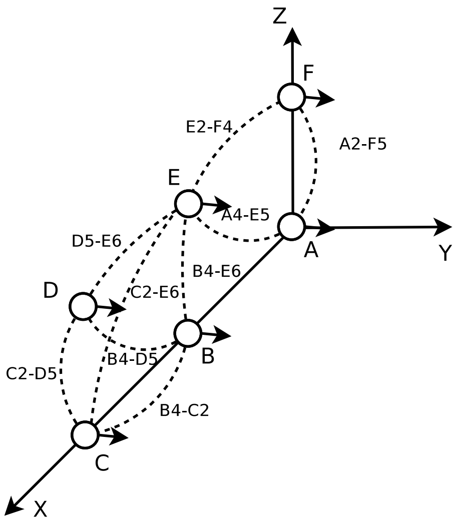
In the end, the ILP-based algorithm increases the number of localized containers as it is able to determine also the position of containers D, E, F. However, it is not able to localize B and C because of the limited number of relations of close proximity (both $\{(B, 1, 0, 0, 1), (C, 2, 0, 0, 1)\}$ and $\{(B, 2, 0, 0, 0), (C, 1, 0, 0, 0)\}$ are possible solutions).



(a) Six containers



(b) X-Z plane



(c) Schematic representation: containers are depicted as circles with an arrow that specifies their orientation, relations of close proximity are depicted as dashed lines

Figure 3.10: Example of ILP-based localization.

3.6 Simulation and results

The effectiveness of the ILP-based algorithm has been evaluated by means of simulations. The numerical results have been obtained applying the

Table 3.4: Average percentage of localized containers (first scenario).

(a) Strawman approach

	faulty nodes (%)					
group size	15	20	25	30	35	40
4x4x4	99.7	98.9	85.8	64.4	41.3	29.7
5x5x5	99.8	89.2	96.6	81.2	57.0	21.6
10x2x6	99.6	98.7	96.3	38.3	40.3	6.9
2x10x6	89.1	87.3	65.5	33.8	6.8	8.4
10x1x6	97.7	94.5	46.7	28.2	10.0	8.0
1x10x6	68.5	24.2	12.0	6.7	3.5	2.8

(b) ILP-based algorithm

	faulty nodes (%)					
group size	15	20	25	30	35	40
4x4x4	100.0	100.0	99.7	98.9	97.8	90.3
5x5x5	100.0	100.0	99.4	98.1	95.5	95.1
10x2x6	100.0	99.8	99.7	98.5	95.8	91.6
2x10x6	100.0	99.3	99.0	98.1	95.1	85.9
10x1x6	100.0	99.3	98.0	93.5	88.8	80.5
1x10x6	100.0	99.3	95.7	89.3	54.2	52.5

previously described model and considering different levels of node faults. The ILP model has been implemented using the AMPL 8.1 modeling language [70] and solved using the CPLEX 9.1.0 solver [71] on a normal PC. The CPLEX solver has also been used to compute the initial solution ($\bar{p}$ and $\bar{\tau}$). The experiments have been carried out considering two different scenarios.

In the first scenario, the containers have been disposed in a group with a box-like shape, with different size: $4 \times 4 \times 4$, $5 \times 5 \times 5$, $10 \times 2 \times 6$, $2 \times 10 \times 6$, $10 \times 1 \times 6$, and $1 \times 10 \times 6$. Moreover, the percentage of faulty nodes have been varied between 15% and 40%. For each disposition and for each percentage of faults, the algorithm has been run on ten different instances randomly generated. The results of the tests are shown on Table 3.4. The first column reports the size of the group, the remaining columns report the average percentage of localized containers depending on the percentage of faulty nodes.

In the second scenario, the space that contains the containers is not completely full. More precisely, the group has been organized as follows: a part of the inner volume with size $I_x \times I_y \times I_z$ is assumed to be completely full of containers, then the remaining part of the volume up to size $V_x \times V_y \times V_z$, contains a number of containers placed randomly. To test the algorithm in this scenario, different configurations of $V_x \times V_y \times V_z$, have been considered: $5 \times 5 \times 5$ (with inner volume $3 \times 3 \times 3$), $10 \times 2 \times 6$ (with inner volume $5 \times 2 \times 3$), $10 \times 2 \times 6$ (with inner volume $10 \times 2 \times 2$), $2 \times 10 \times 6$ (with inner volume $2 \times 5 \times 3$), and $2 \times 10 \times 6$ (with inner volume $2 \times 10 \times 2$). The total number of containers in the group has been randomly varied to be in the interval $125 - 150\%$ of the number of containers contained in the inner volume. Also in this case, for each configuration and for each percentage of faults, the algorithm has been executed on ten different instances randomly generated. The results of the tests are shown on Table 3.5.

Table 3.5: Average percentage of localized containers (second scenario).

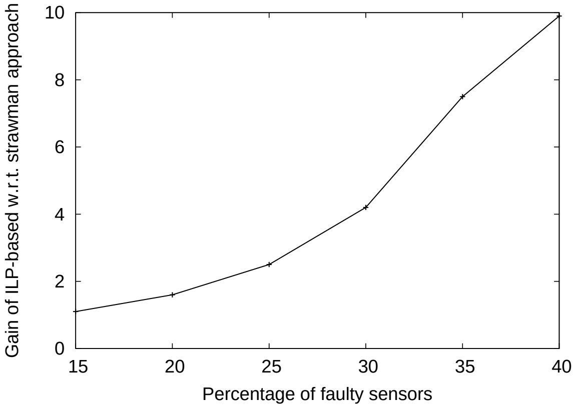
(a) Strawman approach

		faulty nodes (%)					
inner vol.	outer vol.	15	20	25	30	35	40
3x3x3	5x5x5	96.2	75.7	81.9	68.6	27.0	18.8
5x2x3	10x2x6	84.2	91.3	69.4	59.4	41.3	12.6
10x2x2	10x2x6	89.5	75.6	73.2	54.3	21.5	13.6
2x5x3	2x10x6	73.3	74.1	52.8	42.5	15.6	8.4
2x10x2	2x10x6	91.8	84.0	49.8	17.3	21.0	7.7

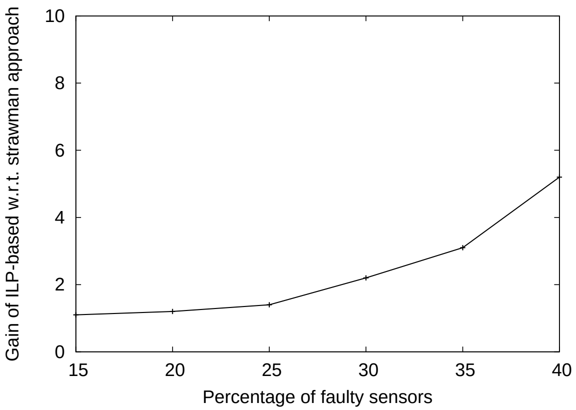
(b) ILP-based algorithm

		faulty nodes (%)					
inner vol.	outer vol.	15	20	25	30	35	40
3x3x3	5x5x5	99.1	97.2	92.9	92.4	88.6	64.1
5x2x3	10x2x6	95.6	95.4	92.6	91.4	80.6	81.0
10x2x2	10x2x6	96.8	95.3	88.5	79.7	74.0	60.4
2x5x3	2x10x6	97.2	96.2	81.3	77.5	55.6	56.4
2x10x2	2x10x6	96.2	89.7	90.2	86.9	67.5	37.9

It is worth noting that, for a given percentage of faults, the average number of localized containers decreases when the configuration is less compact. This is reasonable because if the containers are less compactly disposed, the redundancy of relations of close proximity that are correctly detected is reduced as well. For each test the run time was of about few seconds. With respect to the strawman approach, the ILP-based algorithm increases the number of localized container up to 18.5 times in the first scenario and up to 6.7 times in the second scenario.



(a) First scenario



(b) Second scenario

Figure 3.11: Average gain in localization using the ILP-based approach w.r.t. the strawman approach.

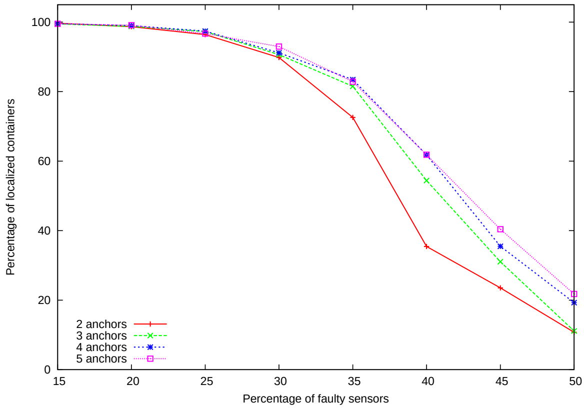
Figure 3.11 shows a summary of the results for the two scenarios. The curves depict the ratio between the number of containers localized using the ILP-based approach and the number of containers localized using the strawman approach, averaged over all the different configurations. In other words, Figure 3.11(a) and 3.11(b) show the average gain obtained through the ILP-based approach with respect to the strawman approach (a value equal to 1 means that there is no gain). In both scenarios the gain is small when the percentage of faults is small (in such situations, also the strawman approach is able to localize almost all the containers), but it becomes very relevant when the number of faults increases. This is particularly evident for the first scenario where containers are disposed in a more compact way. In few cases, even if the gain with respect to the strawman approach is large, the performance of the ILP-based approach could be considered not very satisfactory because the absolute percentage of localized containers is not close to 100%. Nevertheless, it is important to notice that the ILP-based algorithm provides the optimal result, compatibly with the set of geometrical constraints previously introduced (better solutions can be achieved only changing the set of constraints, e.g. increasing the number of nodes attached to a container).

3.6.1 Varying the number of containers with known position

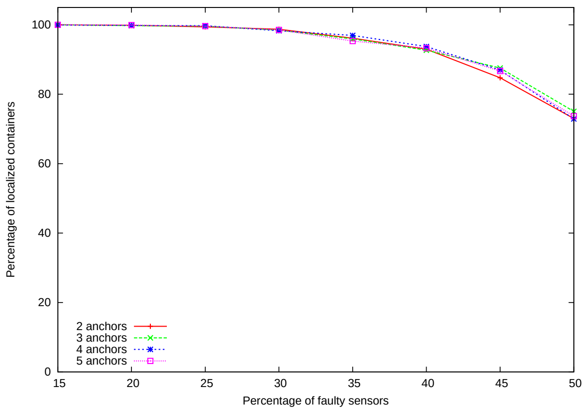
Further experiments have been performed to study the effects of varying the number of containers with known position. Such containers will be called *anchor containers* from now on. Experiments have been carried out on a subset of the configurations previously presented where the number of anchor container has been varied between 2 and 5. Considering that increasing the number of anchor containers allows better localization, the percentage of faults has been pushed up to 50%. The results are the average values obtained for ten random instances, in terms of container placement and position of anchors.

In the first scenario the containers were placed in a box-like shape of size $5 \times 5 \times 5$ and $10 \times 2 \times 6$. Figures 3.12 and 3.13 show the fault tolerance for the strawman approach (on the left) and the ILP-based algorithm (on the right). Note that in both figures the strawman approach is influenced by the number of anchor containers. This because an increase of the number of anchor containers contributes to fill in the missing information and therefore localization improves. On the other hand, the ILP-based algorithm is able to localize a greater number of containers and the number of anchor containers does not affect significantly the localization procedure. This is due to the fact that in this scenario the containers are compactly disposed and the geometrical constraints suffice for the requirements of the ILP-based algorithm.

In the second scenario two configurations have been considered: $5 \times 5 \times 5$ (with inner volume $3 \times 3 \times 3$) and $10 \times 2 \times 6$ (with inner volume $10 \times 2 \times 2$). Figures 3.14 and 3.15 show the two configurations for the strawman approach (on the left) and the ILP-based algorithm (on the



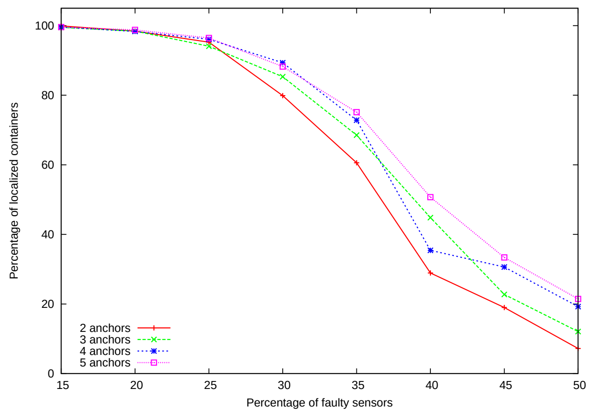
(a) Strawman approach



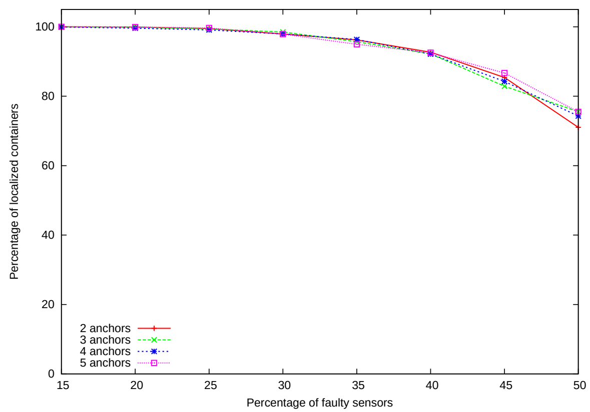
(b) ILP-based algorithm

Figure 3.12: Comparison between the strawman approach and ILP-based algorithm (first scenario, $5 \times 5 \times 5$).

right). Both suffer a decrease of localization performance with respect to the first scenario. This is caused by the sparse placement of the containers that is translated in a lower number of adjacencies. This is more evident, for example, in a very sparse configuration such as $10 \times 2 \times 6$. Note also that the lack of information of the second scenario is counterbalanced by the increasing number of anchor containers which, this time, influences also the performance of the ILP-based algorithm.



(a) Strawman approach

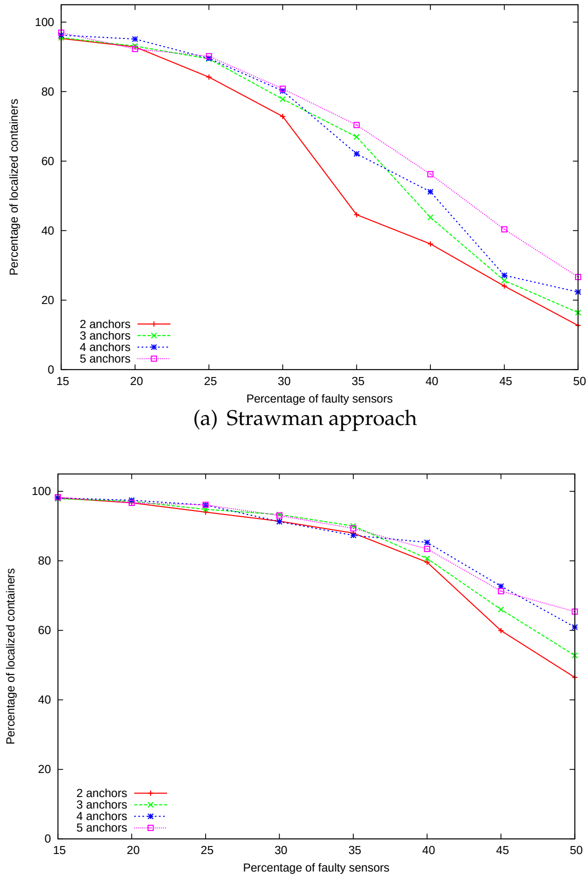


(b) ILP-based algorithm

Figure 3.13: Comparison between the strawman approach and ILP-based algorithm (first scenario, $10 \times 2 \times 6$).

3.7 Implementation and prototyping

A completely working prototype of the system has been built using the *Tmote Sky* nodes commonly available for the realization of wireless sensor networks. A WSN is a wireless network composed of a large number of distributed autonomous sensors capable not only of measuring real world phenomena but also filtering, sharing, combining and aggregating such readings [6]. Each node of the network is equipped with a radio

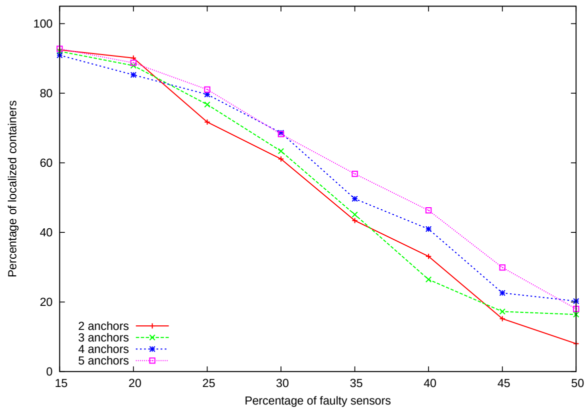


(a) Strawman approach

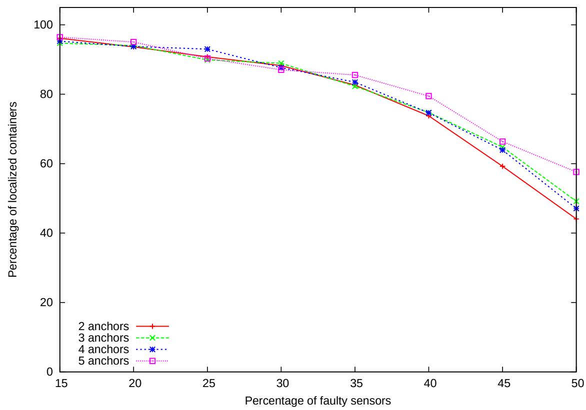
(b) ILP-based algorithm

Figure 3.14: Comparison between the strawman approach and ILP-based algorithm (second scenario, $5 \times 5 \times 5$ with inner volume $3 \times 3 \times 3$).

transceiver, a microcontroller, one or more sensing devices, and is powered by batteries. The nodes organize themselves in a wireless ad-hoc network: each node supports a multi-hop routing algorithm that allows it to forward data packets to a sink node directly connected to a base station. All these features eased the production of the prototype. Considering the aim of the localization system, the sensing features of the devices have not been used. However, it should be noted that the localization system can be easily extended, with the proper sensing equipment, to



(a) Strawman approach



(b) ILP-based algorithm

Figure 3.15: Comparison between the strawman approach and ILP-based algorithm (second scenario, $10 \times 2 \times 6$ with inner volume $10 \times 2 \times 2$).

automate some existing procedures. For example, containers could be sensed to measure the level of CO_2 (to detect hidden people), or to continuously monitor the temperature of refrigerators. The application that is executed on the nodes is written in nesC [11], while the operating system is TinyOS [7].

The software executed on the base station, which processes the data coming from the WSN and which determines the position of containers, has been implemented in Java. To verify its outcome, the program has

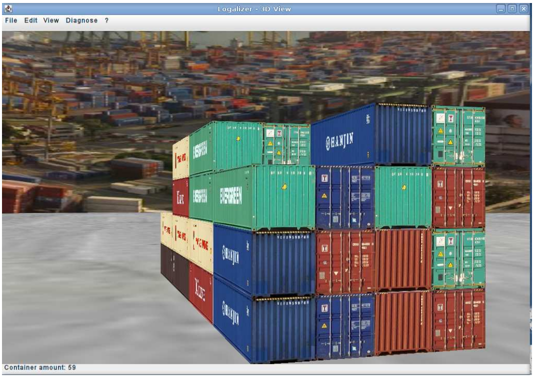


Figure 3.16: The graphical user interface.

been interfaced with a GUI that provides a visual representation of the yard. The GUI interface (shown in Figure 3.16) allows the user to easily locate the position of each container. It is possible to interact with the interface to manipulate the view (rotation, zoom in and out, change from textured mode to wire-frame and vice versa). The user can also select one of the containers with the mouse pointer to retrieve its specific information, or he can search for a given container by specifying its ID. The selected container is then displayed in texture mode, while the other containers are switched in wire-frame mode (this is useful to find containers that are completely hidden by others). The user can also move within the virtual environment.

3.8 Conclusion

In this chapter a non-conventional system for the localization of containers in the yard of ports and terminals has been presented. This localization solution represents an alternative or an integration with respect to the currently available systems, that are based on GPS and RFID technologies. In particular, the use of a wireless sensor network overcomes some problems that may arise from the use of these two technologies: first the need to guarantee a line of sight towards satellites, that limits the use of GPS systems only to outdoor environments or to the containers positioned on the surface of a stack, second the need of an infrastructure and an explicit action for reading RFID tags. Moreover, for both technologies real-time localization of containers is not possible.

The proposed localization system is characterized by high scalability. In fact, when the number of container increases the amount of signaling traffic generated by a single container is not subject to changes, since the maximum number of adjacent container is still equal to six. Obviously, because of data collection, the traffic injected into the network increases linearly with the number of nodes, since they all produce their NCP packets. However, it is important to consider that the movement rate of containers is generally low, and that it is not needed to have an extremely fast reaction time. Thus, the system can be easily tuned to tolerate the size of stacks that are typically found in real scenarios.

However, besides the practical relevance of the implemented system, the main contributions of this work are the following. First, the use of geometrical constraints as a way to reduce the space of possible solutions of a localization problem. As known, localization techniques based on the strength of the received signal are characterized by high error levels. The discretization of node positions makes the localization process simple and scarcely sensible to RSS errors. Second, the idea of modeling the localization problem as a ILP problem where the geometrical constraints can be easily represented and managed. The resulting ILP problem can be solved by using standard procedures and proves to be resilient to a large number of faults: in the two considered scenarios, the overall localization rate is increased by 4.45 and 2.4 times with respect to the strawman approach (average values over all configurations and percentages of faulting nodes). It also reasonable to believe that these techniques can be successfully applied to other localization scenarios characterized by geometrical constraints or to make existing techniques more tolerant to faults.

Chapter 4

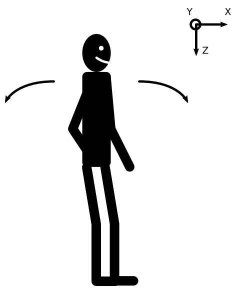
Monitoring of human movements

Monitoring in the field of healthcare can be performed using different algorithms and hardware platforms. The design of monitoring systems should consider the context in which monitoring is performed, the mobility of the subject, the invasiveness of the system and its usability. The focus of this chapter is on the design and implementation of a fall detection system using wireless sensor networks. To this end, many steps need to be performed and are detailed in the following sections. The study starts from a survey of the most relevant parameters, data filtering techniques and testing approaches from the related works done so far [72]. State-of-the-art fall detection techniques were surveyed, highlighting the differences in their effectiveness at fall detection. A system to detect falls with low false positives has been designed, prototyped and tested [73]. The work done has been then generalized by creating a MIMS platform, to support multiple sensors and interoperability with the existing infrastructure. The chapter ends with a description of a testing done to assess the usability and acceptability of the sensors by elderly affected by Alzheimer Disease (AD) and living in a nursing home. The tests showed interesting insights and some lessons were learned from the real world experience.

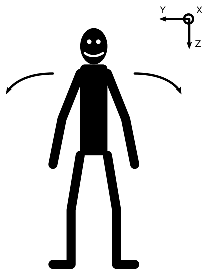
4.1 Basic definitions

A *fall* can be defined in different ways. A a suitable definition of a fall is "Unintentionally coming to the ground or some lower level and other than as a consequence of sustaining a violent blow, loss of consciousness, sudden onset of paralysis as in stroke or an epileptic seizure." [74]. It is always possible to easily re-adapt this definition to address the specific goals a researcher wants to pursue.

In terms of human anatomy, a fall usually occurs along one of two planes, called *sagittal* and *coronal* planes. Figure 4.1(a) shows the sagittal plane, that is an X-Z imaginary plane that travels vertically from the top to the bottom of the body, dividing it into left and right portions. In this case a fall along the sagittal plane can occur forward or backward. Figure 4.1(b) shows the coronal Y-Z plane, which divides the body into dorsal and ventral (back and front) portions. The coronal plane is orthogonal to the sagittal plane and is therefore considered for lateral falls (right or left). Note that if the person is standing without moving, that is, he or she is in a *static* position, the fall occurs following in the down direction. The sense of x, y and z are usually chosen in order to have positive z-values of the acceleration component when the body is falling.



(a) Along sagittal plane

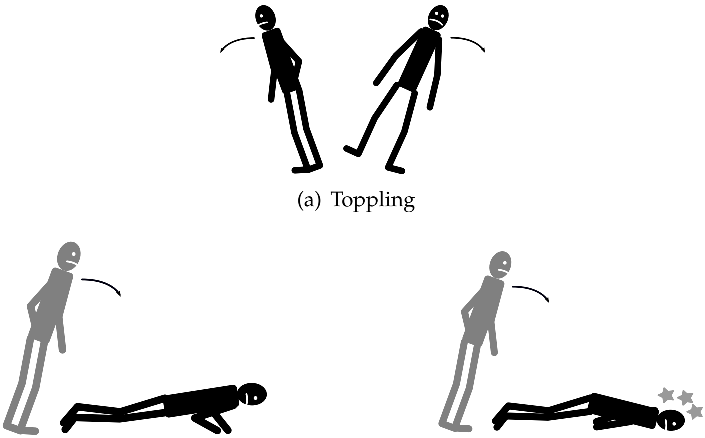


(b) Along coronal plane

Figure 4.1: Fall directions.

Toppling simply refers to a loss in balance. Figure 4.2(a) shows the body from a kinematic point of view. When the vertical line through the center of gravity lies outside the base of support the body starts *toppling*. If there is no reaction to this loss of balance, the body falls on the ground [75].

Let us now consider the fall of a body from a stationary position at height $h = H$. Initially the body has a potential energy mgh which is transformed into kinetic energy during the fall with the highest value just before the impact on the floor ($h = 0$). During the impact the energy is totally absorbed by the body and, after the impact, both potential and kinetic energy are equal to zero. If the person is conscious the energy can be absorbed by the his muscles, for example, using the arms (see Figure 4.2(b)), whereas if the person is unconscious it can lead to sever injuries (see Figure 4.2(c)).



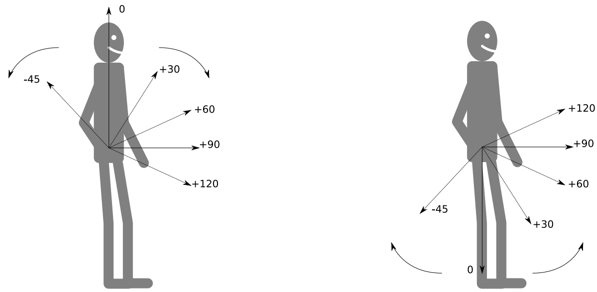
(a) Toppling

(b) Conscious fall

(c) Unconscious fall

Figure 4.2: Kinematic analysis of a fall.

Strictly related to a fall is the *posture*, a configuration of the human body that is assumed intentionally or habitually. Some examples are standing, sitting, bending and lying. A posture can be determined by monitoring the tilt transition of the trunk and legs, the angular coordinates of which are shown in Figure 4.3(a) and Figure 4.3(b) [49, 76]. The ability to detect a posture helps to determine if there has been a fall.



(a) Trunk (b) Legs

Figure 4.3: Angular coordinates.

4.2 How, where and why people fall

Among elderly people that live at home, almost half of the falls take place near or inside the house [77, 78]. Usually women fall in the kitchen whereas men fall in the garden [79]. The rate of falls increases significantly among elderly people living in nursing homes: at least 40% of the patients fell twice or more within 6 months. This rate is five times more with respect to the rate of fall when people live at home. This may be due to people having to aquaint themselves with the new living enviroment and its obstacles.

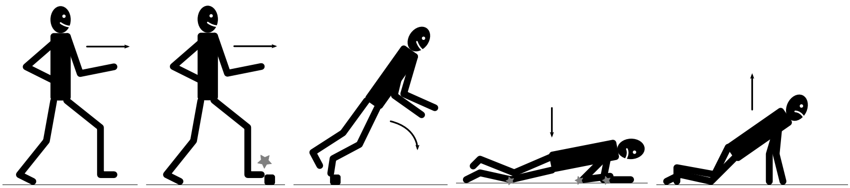
4.2.1 Anatomy of a fall

A fall is generally the consequence of a normal activity of daily living and is triggered by a hard-predictable event such as tripping over, slipping or loss of balance. Once the fall and thus the impact on the floor occur, the subject usually lies down for some seconds or even hours and then tries to recover by himself or with the help of someone else. Just before the impact, the body of the subject is in a free-fall, its acceleration is the same as the gravitational acceleration. Thus, it is possible to distinguish five phases as depicted in Figure 4.4:

1. Activity of Daily Living

2. Hard-predictable event

3. Free-fall
4. Impact
5. Recovery (optional)



(a) Activity of Daily Living (b) Hard predictable event (c) Free-fall (d) Impact (e) Recovery

Figure 4.4: Anatomy of a fall.

Note that there are activities of daily living that can be wrongly detected as falls, e.g. “falling” on a chair.

4.2.2 Physical causes

The factors that lead to most of the falls in people over 65 are to stumble on obstacles or steps and to slip on a smooth surface. The fall is usually caused by loss of balance due to dizziness. Approximately 14% of people do not know why they fall and a smaller number of people state that the fall is due to the fragility of the lower limbs [79].

Further researchers determined that traditional fall prevention measures such as bed rails can make the fall worse [80].

4.2.3 Activities

Most of the falls happen during the ADL that involve a small loss of balance such as standing or walking. Fewer falls happen during daily activities that involve a more significant movement such as sitting on a chair or climbing the stairs. Conversely, activities usually defined “dangerous”, such as jogging or physical exercises are less likely to increase the probability of a fall [81]. There are more falls during the day than during the night [77].

4.2.4 Consequences

Accidental falls are the main cause of admission in a hospital and the sixth cause of death for people over 65. For people aged between 65 and 75 accidental falls are the second cause of death and the first cause in those over 75 [82].

Physical damage

Scratches and bruises are the soft injures due to a fall [82]. In the worst cases the injuries are concentrated on the lower part of the body, mainly on the hip. On the upper part of the body the head and the trunk injuries are the most frequent. About 66% of admissions to an hospital are due to at least one fracture. The fracture of elbow and forearm are more frequent but hip fracture is the most difficult to recover from. Such a fracture in fact requires a long recovery period and involves the loss of independence and mobility.

Sometimes, when a person falls and is not able to stand up by himself, he lies down on the floor for long time. This leads to additional health problems such as hypothermia, confusion, complications and in extreme cases can cause death [83].

Psychological damage

A fall also involves hidden damages that affect the self-confidence of a person [83]. Common consequences are fear, loss of independence, limited capabilities, low self-esteem and generally, a lower quality of life.

Economic damage

The direct costs associated with falls are due to the medical examinations, hospital recoveries, rehabilitation treatments, tools of aid (such as wheelchairs, canes etc.) and caregivers service cost [84].

Indirect costs concern the death of patients and their consequences. Recent studies have determined that in the year 2000 alone fall-related expenses was above 19 billion dollars and it is estimated to reach 54.9 billion in 2020. This shows that year by year, health costs due to the falls are increasing dramatically [85].

4.2.5 Fall risk factors

A person can be more or less prone to fall, depending on a number of risk factors and hence a classification based on only age as a parameter is not enough. In fact, medical studies have determined a set of so called *risk factors*:

- Intrinsic:

  Age (over 65)

  Low mobility and bone fragility

  Poor balance

  Chronic disease

  Cognitive and dementia problems

  Parkinson disease

  Sight problems

  Use of drugs that affect the mind

  Incorrect lifestyle (inactivity, use of alcohol, obesity)

  Previous falls

- Extrinsic:

  Individual (incorrect use of shoes and clothes)

  Drugs cocktail

- Internal Environment:

  Slipping floors

  Stairs

  Need to reach high objects

- External Environment:

  Damaged roads

  Crowded places

  Dangerous steps

  Poor lighting

There is a clear correlation between the above list and the probability of fall. The number of people that fall are as follows [81]:

- 8% of people without any of risk factors
- 27% of people with only one risk factor
- 78% of people with four or more risk factors

The history of the falls is also important since people who have already fallen two times are more at risk to fall again. This can be due to psychological (fear, shame, loss of self-esteem), and/or physical (injuries, lack of exercise) reasons.

4.2.6 Typical fall scenarios

The most important scenarios of falls are described by [19] in detail:

- **Fall from standing**
  1. It lasts from 1 to 2 seconds.
  2. In the beginning the person is standing. At the end the head is stuck on the floor for a certain amount of time.
  3. A person falls along one direction and the head and the center of mass move along a plane.
  4. The height of the head varies from the height while standing and the height of the floor.
  5. During the fall the head is in free-fall.
  6. After the fall the head lays in a virtual circle that is centered in the position of the feet before the fall and has radius the height of the person.
- **Fall from chair**
  1. It lasts from 1 to 3 seconds.
  2. In the beginning the height of the head varies from the height of the chair to the height of the floor.
  3. During the fall the head is in free-fall.
  4. After the fall the body is near the chair.
- **Fall from bed**
  1. It lasts from 1 to 3 seconds.

2. In the beginning the person is lying.
3. The height of the body varies from the height of the bed to the height of the floor.
4. During the fall the head is in free-fall.
5. After the fall the body is near the bed.

With the description of the main falls it is possible to simplify the complexity of a fall. This enables in turn to focus on the resolution of the detection fall problem, rather than on the reconstruction of a detailed scenario. The simplified and theoretical description often reflects the practical sequence of a fall.

4.3 Performance parameters and scenarios

A real working fall detection system requires to be sufficiently accurate in order to be effective and alleviate the work of the caregivers. The quality of the system is given by three indexes that have been proposed based on the four possible situations shown in Table 4.1:

	A fall occurs	A fall does not occur
A fall is detected	*True Positive* (TP)	*False Positive* (FP)
A fall is not detected	*False Negative* (FN)	*True Negative* (TN)

Table 4.1: Possible outputs of a fall detection system.

- *Sensitivity* is the capacity to detect a fall. It is given by the ratio between the number of detected falls and the total falls that occurred:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.1}$$

- *Specificity* is the capacity to avoid false positives. Intuitively it is the capacity to detect a fall only if it really occurs:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4.2}$$

- *Accuracy* is the ability to distinguish and detect both fall (TP) and non-fall movement (TN):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \tag{4.3}$$

Where *P* and *N* are, respectively, the number of falls performed and the number of non-falls performed.

Accuracy (Equation 4.3) is a global index whereas sensitivity and specificity (Equations 4.1 and 4.2) enable a better understanding of the some limits of a system.

A fall exhibits high acceleration or angular velocity which are not normally achievable during the ADL. If we use a fixed low threshold to detect a fall, the sensitivity is 100% but the specificity is low because there are fall-like movements like sitting quickly on a chair, a bed or a sofa which might involve accelerations above that threshold.

The logged data is sometimes pre-processed by applying some filters: a low-pass filter is used to perform posture analysis and a high-pass filter is applied to execute motion analysis. However, this processing is not mandatory and it strongly depends on the fall detection algorithm.

The calibration of the sensors is sometimes neglected or not mentioned in research studies, but it is an important element that ensures a stable behavior of the system over time.

Amplitude parameters are useful during specific phases of the fall [86, 87, 55]. The *Total Sum Vector* given in Equation 4.4 is used to establish the start of a fall:

$$\text{SV}_{\text{TOT}}(t) = \sqrt{(A_x)^2 + (A_y)^2 + (A_z)^2} \tag{4.4}$$

where A_x, A_y, A_z are the gravitational accelerations along the x, y, z-axis.

The *Dynamic Sum Vector* is obtained using the Total Sum Vector formula applied to accelerations that are filtered with a high-pass filter taking into account fast movements.

The *MaxMin Sum Vector* given in Equation 4.5 is used to detect fast changes in the acceleration signal, which are the differences between the maximum and minimum acceleration values in a fixed-time ($\Delta t = t_1 - t_0$) sliding window for each axis.

$$\mathrm{SV}_{\mathrm{MaxMin}}(\Delta t) = \max_{t_0 \le i \le t_1} \mathrm{SV}_{\mathrm{TOT}}(i) - \min_{t_0 \le j \le t_1} \mathrm{SV}_{\mathrm{TOT}}(j) \tag{4.5}$$

Vertical acceleration given in Equation 4.6 is calculated considering the sum vectors $\mathrm{SV}_{\mathrm{TOT}}(t)$ and $\mathrm{SV}_{\mathrm{D}}(t)$ and the gravitational acceleration G.

$$Z_2 = \frac{\mathrm{SV}^2_{\mathrm{TOT}}(t) - \mathrm{SV}^2_{\mathrm{D}}(t) - G^2}{2G} \tag{4.6}$$

4.3.1 Standard trial scenarios and characteristics

Researcher should agree on a common set of trials in order to test and compare different fall detection systems. In Table 4.2 we propose a set of actions for which a fall detection system should always detect a fall. In Table 4.3 we propose a set of fall-like activities of daily living that can lead the system to output false positives. In addition to performing tests on all the listed 36 actions, each research group can combine them in sequential protocols, called *circuits* (e.g. sitting, standing, walking, falling).

#	Name	Symbol	Direction	Description
1	Front-lying	FLY	Forward	From vertical going forward to the floor
2	Front-protecting-lying	FPLY	Forward	From vertical going forward to the floor with arm protection
3	Front-knees	FKN	Forward	From vertical going down on the knees
4	Front-knees-lying	FKLY	Forward	From vertical going down on the knees and then lying on the floor
5	Front-right	FR	Forward	From vertical going down on the floor, ending in right lateral position
6	Front-left	FL	Forward	From vertical going down on the floor, ending in left lateral position
7	Front-quick-recovery	FQR	Forward	From vertical going on the floor and quick recovery
8	Front-slow-recovery	FSR	Forward	From vertical going on the floor and slow recovery
9	Back-sitting	BS	Backward	From vertical going on the floor, ending sitting
10	Back-lying	BLY	Backward	From vertical going on the floor, ending lying
11	Back-right	BR	Backward	From vertical going on the floor, ending lying in right lateral position
12	Back-left	BL	Backward	From vertical going on the floor, ending lying in left lateral position
13	Right-sideway	RS	Right	From vertical going on the floor, ending lying
14	Right-recovery	RR	Right	From vertical going on the floor with subsequent recovery
15	Left-sideway	LS	Left	From vertical going on the floor, ending lying
16	Left-recovery	LR	Left	From vertical going on the floor with subsequent recovery
17	Syncope	SYD	Down	From standing going on the floor following a vertical trajectory
18	Syncope-wall	SYW	Down	From standing going down slowly slipping on a wall
19	Podium	POD	Down	From vertical standing on a podium going on the floor
20	Rolling-out-bed	ROBE	Lateral	From lying, rolling out of bed and going on the floor

Table 4.2: Actions to be detected as falls.

Participant characteristics

Different people have different physical characteristics and therefore it is extremely important to specify, for each trial, the following five parame-

#	Name	Symbol	Direction	Description
21	Lying-bed	LYBE	Lateral	From vertical lying on the bed
22	Rising-bed	RIBE	Lateral	From lying to sitting
23	Sit-bed	SIBE	Backward	From vertical sitting with a certain acceleration on a bed (soft surface)
24	Sit-chair	SCH	Backward	From vertical sitting with a certain acceleration on a chair (hard surface)
25	Sit-sofa	SSO	Backward	From vertical sitting with a certain acceleration on a sofa (soft surface)
26	Sit-air	SAI	Backward	From vertical sitting in the air exploiting the muscles of legs
27	Walking	WAF	Forward	Walking
28	Jogging	JOF	Forward	Running
29	Walking	WAB	Backward	Walking
30	Bending	BEX	Forward	Bending of about X degrees (0-90)
31	Bending-pick-up	BEP	Forward	Bending to pick up an object on the floor
32	Stumble	STU	Forward	Stumbling with recovery
33	Limp	LIM	Forward	Walking with a limp
34	Squatting-down	SQD	Down	Going down, then up
35	Trip-over	TRO	Forward	Bending while walking and than continue walking
36	Coughing-sneezing	COSN	-	-

Table 4.3: Activities that must not be detected as falls.

ters:

- Gender
- Age
- Weight
- Height
- Body Mass Index [1]

Hardware characteristics

Variation among the technology of the nodes depends on their level of the development and manufacturing cost. It is therefore important to define some basic characteristics for the hardware used in trials:

- Model
- Sampling frequency
- Update rate
- Movement detection delay time
- Range of measurement

[1]Body mass index (BMI) is a measure of body fat based on height and weight that applies to adult men and women.

- Size
- Weight
- Wired/wireless communication protocol

4.4 The basic system

Figure 4.5 shows the general architecture for a human movement monitoring system based on a wireless sensor network. One or more sensing nodes are used to collect raw data. Analysis of the data can be performed on the node or on the base station by a more powerful device such as a smartphone or a laptop. The wireless connectivity standard between the nodes (e.g. ZigBee) can be different from the one that connects the sink node with the base station (e.g. Bluetooth). The base station in turn acts as a gateway to communicate with the caregivers through wireless and/or wired data connection (e.g. Internet or other mobile phones).

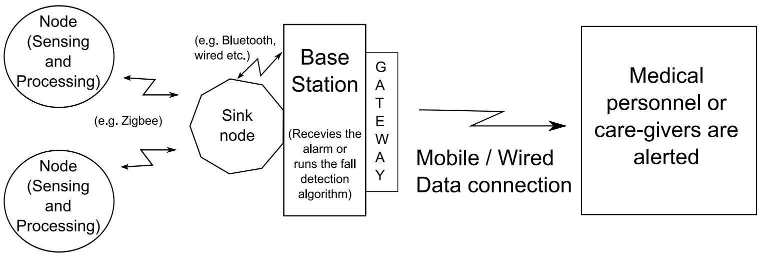


Figure 4.5: General system architecture.

4.4.1 Node sensors and position

A node for kinematic monitoring is typically instrumented with the following sensors:

- Accelerometer, to measure the acceleration.
- Gyroscope, to measure the angular velocity.

In particular, the gyroscope requires more energy than the accelerometer. If we connect the acceleration of the movements with the position of the node worn by the patient, it would be possible to detect the posture of a person.

The placement of one or more nodes on the body is the key to differentiate the influences of various fall detection algorithms. It is not possible to neglect the usability aspect, since it strongly affects the effectiveness of the system. A node placed on the head gives an excellent impact detection capability, but more hardware efforts are required to ensure its usability for wearing the node continuously. The wrist is not recommended to be a good position, since it is subject to many high acceleration movements that would increase the number of false positives. The placement at the waist is more acceptable from the user point of the view, since this option fits well in a belt and it is closer to the center of gravity of the body. There are many other node locations selected by researchers, such as the armpit, the thigh or the trunk, quoting their own advantages and disadvantages as explained later. Sometimes the nodes are inserted in clothes, for example jackets, or in accessories such as watches or necklaces.

Our basic system was based on a single waist-mounted sensor, extended with a set of techniques able to filter the false alarms and increase its accuracy.

Even if the analysis of acceleration along the three different axes would provide more detailed information, the basic system uses only the magnitude of the acceleration vector. This choice is because the device is not completely integral with respect to the patient's body, and its orientation may change both at the time when it is put on or as a consequence of movements.

The basic system detects a fall-like event when the following conditions are satisfied: $i)$ the magnitude of acceleration is greater than $3\,g$; $ii)$ the peak of acceleration magnitude is followed by a "static interval", which is a period of at least $1200\,ms$ in which there are no peaks exceeding the threshold.

The $3\,g$ value has been chosen so that, according to results obtained in [88] and in our collected data, the risk of false negatives is avoided, thus achieving a 100% degree of sensitivity. There are several fall-like ADLs that reach this threshold, thus the specificity achieved is inevitably low. The static interval is used to understand when the previous event is finished, both when it is a fall-like ADL or when it is a real fall. After the detection of a fall-like event, the system tries to understand if the event is the consequence of an ADL or has been caused by a real fall. In the second case the system alerts the caregiver.

4.4.2 Recognition of false alarms

The main contribution of our approach is the definition of a set of techniques able to filter false alarms without using posture information and thus making possible the adoption of fall detection systems based on a single accelerometer [73].

False alarms are recognized on the basis of peculiar patterns of the acceleration data. In a preliminary phase, we performed experiments to collect information about different types of real falls and common ADLs. In particular, we gathered the acceleration data of about 32 falls and 68 executions of different ADLs (the details of the data collection process, including the list of the different types of falls and ADLs, are presented in Section 4.4.3).

Activities of daily living that may cause false alarms

The analysis of the characteristics of ADLs and falls is fundamental for the development of filtering methods able to isolate false alarms from actual falls. The following ones are some classes of ADLs that can be confused with real falls and could generate false alarms:

A) sitting/lying quickly on soft/elastic surfaces (such as a bed or a sofa);

B) sitting quickly on medium/hard surfaces (such as a chair);

C) jumping on the ground.

The reason is that they present at least an acceleration magnitude peak which in some cases can be greater than $3\,g$, which is the threshold used to detect a fall. Fortunately, each previously listed class of fall-like ADLs presents at least a feature that can be used to distinguish it from a real fall:

A) the fall-like ADLs happen on soft/elastic surfaces, thus they are characterized by smooth acceleration peaks;

B) it is distinguished by low/medium kinetic energy, which is quickly absorbed with a single sharp peak;

C) it has a typical acceleration shape, due to push, free fall and landing phases.

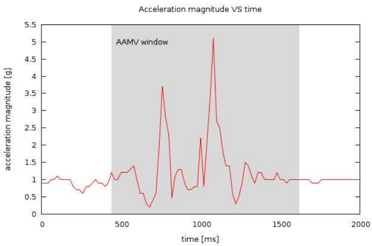
Recognition of ADLs belonging to classes A and B

Falls are characterized by a violent impact on hard surfaces causing sharp peaks in the graph of acceleration magnitude. In general, the graph of acceleration magnitude contains several peaks (even if not all of them are greater than $3\,g$) because of the following reasons: different parts of the body touch the ground at different times; the relatively high kinetic energy causes a sort of "rebound" effect on the body or parts of it. A sharp peak is characterized by quick variations of acceleration magnitude from a sample to the next. Differently, class A ADLs (sitting/lying on soft/elastic surfaces) present smooth peaks, since the kinetic energy is gracefully dissipated after impact, while class B ADLs (sitting on hard surfaces) generally determine a single sharp peak followed by quick stabilization. Therefore, there are relatively quick and numerous acceleration variations in falls, slow variations in class A ADLs and few variations in class B ADLs.

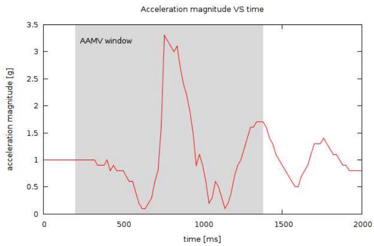
In order to extract the features previously described from raw acceleration data, we defined a measure, the *Average Acceleration Magnitude Variation* index, defined as follows:

$$AAMV = \sum_{i \in W} \frac{|acc_{i+1} - acc_i|}{number\ of\ samples\ in\ W} \tag{4.7}$$

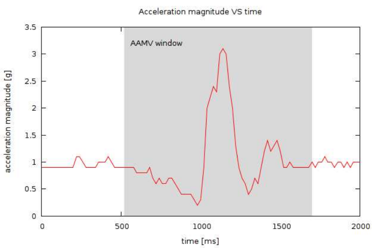
The value is computed in a time window (W) of proper size which includes the $3\,g$ peak. Through an empirical evaluation we found that the AAMV window which provides the best results, according to our dataset, is the one that starts $640\,ms$ before the last $3\,g$ peak and ends $540\,ms$ after the peak. The value of AAMV is directly proportional not only to how quickly the acceleration magnitude changes, but also to the number of peaks present in the considered window. As a consequence, we expect to find greater AAMV values for real falls with respect to those obtained for ADLs belonging to classes A and B. Figures 4.6(a), 4.6(b), 4.6(c), and 4.6(d) show the typical acceleration graphs of examples of real falls and ADLs belonging to classes A and B. The AAMV window is also shown. We performed a binary classification of data and, as expected, the AAMV values obtained for ADLs belonging to class A and B are lower than those obtained for real falls. This difference



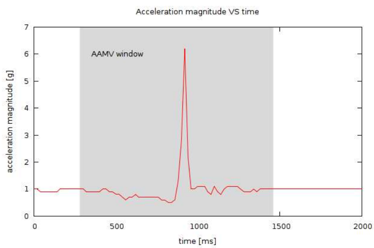
(a) Real fall



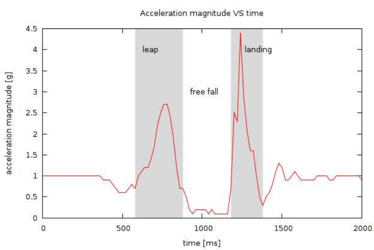
(b) Class A (sitting/lying quickly on a elastic surface)



(c) Class A (sitting/lying quickly on a soft surface)



(d) Class B (sitting quickly on a medium/hard surface)



(e) Class C (jumping)

Figure 4.6: Acceleration for real falls and ADLs belonging to classes A, B, and C.

is clearly shown in Figure 4.7. By comparing the AAMV of the potential fall with a threshold (approximately $0.27\,g$) it is possible to classify the event as a real fall, or as an ADL belonging to class A or B. In the latter case, even if the acceleration magnitude exceeds the $3\,g$ threshold, the fall-like event is filtered and an alarm is not raised, thus increasing the specificity of the system.

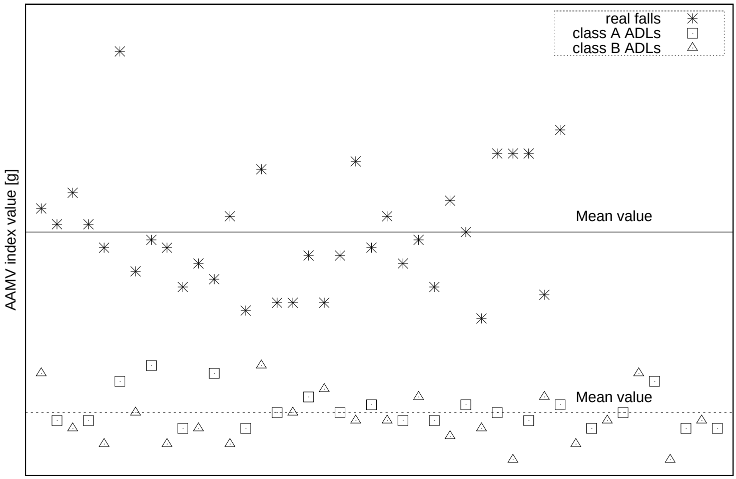


Figure 4.7: Values of the AAMV index for real falls and ADLs belonging to classes A and B.

Recognition of ADLs belonging to class C

The filtering method based on the AAMV index cannot be applied successfully to class C fall-like ADLs. The reason of this can be explained considering Figure 4.6(e), which shows a typical acceleration magnitude graph obtained performing a small jump on the floor. There are two relevant peaks: the first is produced when the user leaps, the second (about $4.5\,g$) is higher and sharper and corresponds to the landing on the floor. These variations determine AAMV values for jumps that are comparable to those obtained for real falls.

Jumping consists of three phases: leap, free fall, landing. As shown in Figure 4.6(e), each phase can be easily identified in the acceleration magnitude graph. This peculiar shape represents the feature that can be used to filter this class of false alarms. Recognition of a jump can be performed through the following procedure:

1. Verify the presence of a peak associated with the leap.
2. Find two instants
   (a) *landing start*: $80\,ms$ before the last acceleration magnitude peak

greater than $3\,g$. This is a empirical estimation of when landing begins.

(b) *leap end*: found searching backwards in time from $20\,ms$ before landing start until measured acceleration magnitude is greater than or equal to $1\,g$. This is a simple estimation of when free fall begins.

3. Use these instants to find two quantities:

   (a) *Free Fall Interval* (FFI): as the difference between landing start and leap end.

   (b) *Free Fall Average Acceleration Magnitude* (FFAAM): average acceleration magnitude in the free fall interval.

We noticed that real falls have a lower FFI or a higher FFAAM value with respect to the values obtained for jumps. Thus, we defined two conditions useful to recognize an event as a class C ADLs:

- $FFI > 100ms$
- $FFAAM < 0.5g$

An event is classified as a class C ADL only if both tests are passed. In such case, an alarm is not raised.

4.4.3 Collection of data and implementation

In this section we describe the acquisition of data related to real falls and different ADLs. We also provide some details about the implementation of the system.

Data collection

Data acquisition is the first step in every fall detection study and it is a time-consuming process. Accelerations measured during tests are fundamental to understand the features that can be used to isolate falls from harmless actions like sitting or lying. Unfortunately, previous studies generally describe the performed tests and the obtained results, but the acceleration data is usually not made available.

User ID	Sex	Age	Height cm	Weight kg
U1	male	24	175	62
U2	male	37	177	81
U3	male	26	178	75
U4	male	64	175	91

Table 4.4: Users involved in the data collection.

Action	Short form
Jumping	JUM
Lying quickly on bed	LYBE
Lying quickly on a mat	LYMA
Lying quickly on sofa	LYSO
Parkinsonian gait	PGA
Jogging (forward)	JOF
Sitting quickly on armchair	SAR
Sitting quickly on chair	SCH
Sitting quickly on sofa	SSO

Table 4.5: List of fall-like ADLs.

Our experiments involved four male subjects. They have been engaged into a battery of tests designed to collect data about the most common fall-like ADL and falls. Recorded ADLs always present at least an acceleration magnitude peak greater than $3\,g$, which is followed by a static interval lasting at least $1200\,ms$. These are the kinds of ADLs that would produce a false alarm in the basic fall detection system. Skate pads have been used to avoid injuries to knees, elbows and wrists, since landing always took place on hard surfaces. This also ensured a realistic execution of falls as it removed the fear of hitting the ground (and thus reduced the effects of those semi-unconscious actions aimed at self-protection in planned falls). Table 4.4 shows the profiles of the volunteers who have been involved in the collection of data. Table 4.5 describes the list of fall-like ADLs, while Table 4.6 shows the list of real falls. In fact, different types of falls could be defined, each characterized by a peculiar way of landing on the floor, or by the action performed before losing balance. Finally, Tables 4.7 and 4.8 respectively show the number of ADLs and the number of falls performed by each volunteer.

Action	**Short form**
Rolling out of bed	ROBE
Fall almost vertically down from standing (syncope)	SYD
Fall after parkinsonian gait	FPG
Fall forward landing on hands first	FPLY
Fall after a small jump	FJU
Fall forward landing on knees first	FKLY
Fall while running	FRU
Fall from sitting	FSI

Table 4.6: Types of falls performed.

	LYBE	**LYMA**	**LYSO**	**SAR**	**SCH**	**SSO**	**JUM**	**PGA**	**JOF**
U1	6	2	4	4	6	2	14	5	5
U2	0	0	0	0	3	0	0	0	0
U3	0	2	0	0	3	0	0	0	0
U4	3	0	0	3	3	3	0	0	0

Table 4.7: Fall-like ADLs performed by each user.

4.4.4 Falls study database

Data acquisition is probably the most difficult and time-consuming portion in a fall-detection study. In the best case, log files of fall trials contain raw accelerations measured during the simulation of an action (fall or ADL). If other researchers want to access and use such raw accelerations, it is necessary to provide an accurate description of the trials. Moreover, previous studies generally describe the tests performed and the results obtained, but the acceleration data is usually not publicly made available. This points out the need for a database with a standard structure to store all the logs. Such a database is intended to be available to the scientific community and has two main advantages: on one hand the possibility of storing and sharing data coming from sensors following a standard format; on the other hand, the availability of raw sensed data before, during, and after a fall or an activity of daily living that enables the researchers to test and validate fall detection algorithms using the same test-beds.

A trial or *experiment* is described in terms of the *action* performed, the *configuration* used for the wearable device and the *user's profile*. Human actions under study are all characterized by the following aspects: i) *posture*: users have a particular body orientation before and after the action

	FPLY	FKLY	ROBE	SYD	FSI	FJU	FPG	FRU
U1	5	2	3	3	3	2	2	2
U2	2	2	0	2	0	0	0	0
U3	0	3	0	1	0	0	0	0

Table 4.8: Falls performed by each volunteer.

is performed; ii) *surface*: user's body is supported by a particular kind of surface before and after the action is performed. A configuration establishes a particular way to sense kinematic data, and it can be described in terms of the following: i) *position*: the device is worn at some body position; ii) *device used*: the type of sensor node adopted for the collection of data. The Entity-Relationship model depicted in Figure 4.8 is derived from the previous considerations.

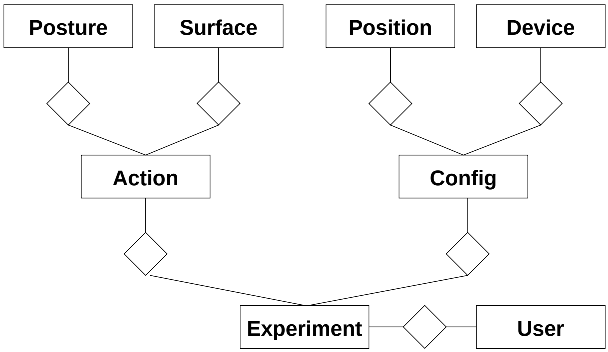


Figure 4.8: Database Entity-Relationship diagram.

A possible structure of the table is the following:

```
Postures (ID, posture)

Surfaces (ID, surface)

Action (ID, starting_posture, starting_surface, ending_posture,
         ending_surface, description)

Position (ID, position)

Device (ID, manufacturer, model, description, characteristics)

Configuration (ID, record_content, Mote, scale_G, sample_frequency,
               Body_position, x_direction, y_direction, z_direction)

Users (ID, age, gender, height_cm, weight_kg, body_mass_index)

Experiments (ID, Configuration, Action, User, content)
```

Note that we decided to collect, represent, and store extra information, such as the posture of the user before and after a potential fall, the separate acceleration values and acceleration magnitude as-well. This has been done to foster the reuse of the collected data and to enable the evaluation of future techniques on the same sets of data. A database interface is available online [89].

4.4.5 Implementation details

We used a SHIMMER mote as the wearable device [9], which incorporates a triaxial accelerometer, a microcontroller and a radio transceiver. The output of the accelerometer is sampled by the microcontroller at a $50\,Hz$ frequency. There are two ways of implementing the filtering technique. The first consists of using the embedded intelligence of the wearable device. The main advantage of this approach is that several useless transmissions to the base station can be avoided by filtering the false alarms directly on the remote device, enhancing the lifetime of batteries. The second consists in running the algorithm on the base station, after the acceleration samples of the event have been received via radio. The advantage of this approach is the abundance of computing resources on the base station. Since the extraction of AAMV, FFI, and FFAAM values from acceleration data is not computationally intensive, we implemented the filtering techniques directly on the wearable device in order to communicate with the base station only when an alarm occurs. Software has been developed using the TinyOS/nesC platform.

4.4.6 Results

The collected data has been used to evaluate the performance of our techniques for the recognition of false alarms. As previously mentioned in Section 4.4.2, in the data we collected, the set of AAMV values of real falls does not overlap with the set of AAMV values of ADLs belonging to classes A and B. This allowed us to filter all the false alarms coming from such activities without missing any real fall. Similarly, all the ADLs belonging to class C of our collected data satisfy the conditions described in Section 4.4.2, while the real falls fail both of them. Henceforth, these filtering methods achieve 100% sensitivity and specificity values with respect to the data we collected.

Despite the relatively small data set, the prototype proves that the idea of filtering ADLs on the basis of peculiar features of the acceleration data can be used to enhance significantly the specificity of a basic fall detection system. We applied this idea to a system based on just one accelerometer placed at patient's waist. However, the same idea could be adapted to improve the accuracy of systems based on two or more devices, or placed at a different position of the patient's body, after proper analysis of the acceleration data and the extraction of new peculiar features. It is also important to notice that the enhancements in terms of detection accuracy have been achieved without compromising the usability of the system.

4.5 A Minimally Invasive Monitoring Sensing platform

Continuous healthcare patient monitoring requires real-time processing of signals over a long period of time, varying from few hours to many days, depending on the patient's medical condition and the symptoms being monitored. For example, in fall monitoring, movement-related information is required to distinguish normal ADL from abnormal. It requires continuous monitoring during daytime, in which a person is engaged in many day to day activities. Monitoring of sleep-related fall and other disorders such as apnea, requires data for eight hours during the night, in which a person is often found less active. With the study of sizeable data collected over the past from passive monitoring, researchers are now interested in studying physiological signals for early detection, prevention, and prediction of anomalous events.

The passive monitoring systems consist of simple off-the-shelf cameras and on/off sensors placed on doors, toilets, beds, chairs in the individual's living area, to monitor his/her activities. In addition, the system provides reminders for medication and similar instructions. When people interact with the environment, infrared or pressure sensors on the floor or bed are triggered and the system is able to recognize an abnormal activity, such as a fall.

There are many limitations with this approach: it requires installation time; has limited coverage capability (only where there are sensors) and privacy violations. Both the camera and the environmental sensor devices require pre-built infrastructures which enable their use only in hospitals and houses, but not in the outdoor environment. A large amount of redundant data is collected and often incidents are missed. For example, it is not possible to detect a fall occurring outside a floor sensor.

The use of wearable and active sensors provides better monitoring ability [90]. They enable mobility and therefore ensure unobstructive ubiquitous and continuous monitoring. With built-in sensors such as accelerometers, gyroscopes, biosensors etc. movement and physiological signals are transmitted via wireless links and analyzed immediately. This approach offers additional advantages such as low installation cost (indoor and outdoor) and small form factor. Moreover, a sensor in a wearable device can pre-process and filter redundant data, so as to provide useful information only. This reduces the traffic in the wireless network and extends battery life of the sensors.

This section introduces MIMS, a platform for building a comprehensive and customizable health monitoring system [91]. The core component of this platform is called *Virtual Hub*, which acts as a gateway for communication of captured data from the wearable device and at the same time enables coordination of signal processing and fusion of sensor data from different vendors including those from passive monitoring systems. The Virtual Hub is hardware independent, and it can be executed independently on a personal computer or a smartphone, and in a distributed web environment as well.

In the recent years, passive monitoring solutions have penetrated into health monitoring systems in homes, assisted living environments, and nursing homes. They provide timely interventions in case of emergency [92, 93]. However, they face many unresolved problems such as lack of intelligence to support proactive care by assessing the individual's physiological conditions. Monitoring platforms have been also narrowly focused on using a specific set of sensors, vendor specific software and protocols. They do not address the problem of interoperability among heterogeneous sensors using different software and protocols. The MIMS proposed here provides in-the-gap solution for the present and the future research in active and passive physiological continuous monitoring.

4.5.1 Platform architecture

The MIMS is a flexible and scalable platform for minimally invasive health monitoring. As shown in Figure 4.9 a person wears one or more wireless sensor devices (SHIMMER [9] and Enobio [94] in our fall study) based

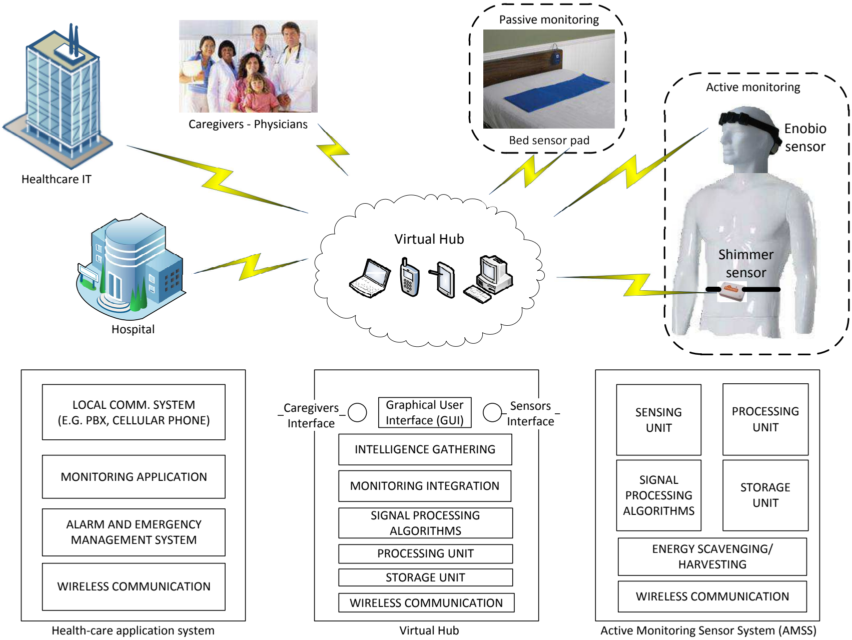


Figure 4.9: MIMS architecture.

on the monitoring application.

The *Active Monitoring Sensor System (AMSS)* shown in figure is composed of standard off the shelf units but with limited resources for sensing, processing, storing and communicating. They are usually equipped with a rechargeable battery with limited lifetime. Hence, an *Energy scavenging and harvesting* module is added to extend the use of the device without being recharged frequently. A *Signal processing algorithms* module contains the algorithms to enable pre-processing, analysis and data filtering: the role of this unit is to efficiently manage and consider only the relevant data coming from the *Sensing unit*. It also interacts with the *Storage unit* to retrieve or permanently store the data. Some operations such as the Fast Fourier Transform (FFT), demand higher computational power and cannot be performed with on-board *Processing unit*. For this reason the Signal processing algorithms unit can delegate all or part of its processing operations to the *Virtual Hub*.

The Virtual Hub can reside on a desktop computer, a laptop, a smartphone, a PDA or a distributed system. It represents not only a gateway between the monitored person and the caregivers, but also a shared point between the active and passive sensing devices. Through the *Sensors interface*, the Virtual Hub becomes the first communication hop for the sensor devices. On the other hand, with the *Caregivers interface*, the Virtual Hub can connect to the *Health-care application systems* currently used by hospitals, caregivers or those provided by Healthcare industry. In the Virtual Hub, a *Monitoring integration* module is used to aggregate the active and passive monitoring data, enabling the system to perform correlation analysis. The aggregated data and the data that cannot be processed on the sensor device, are fed into local *Signal processing algorithms* module which uses powerful computational resources. An *Intelligence gathering* module is used in the event classification algorithms using machine learning techniques such as State Vector Machine (SVM), Bayes networks, reinforced learning etc. The *Graphical User Interface* (GUI) allows the patients to interact with the sensor devices and perform self-diagnosis and check-up. The use of the GUI is made optional, especially for people who are not able to interact with the device.

The Virtual Hub easily adapts to deployment of monitoring systems to different care environments. The *Health-care application system* on the caregivers devices or the institutions infrastructure uses standard communication protocols which are included in the Virtual Hub. The Virtual Hub can be connected and integrated to the existing *Local communication system* of the caregivers. This enables the possibility for a caregiver, not only to receive and manage an alarm through the *Alarm and Emergency Management system*, but also to remotely monitor and perform online queries to retrieve the health condition of a patient from a *Monitoring application*.

The *Wireless communication* module is common to the three systems and relies on protocols such as IEEE 802.15.4, Zigbee, Bluetooth or those provided by telecommunication networks. Such a unit can be readily integrated in the system or plugged in (e.g. via a USB port).

The *Signal processing algorithm* module provides information to the alarm units which will trigger the corresponding alarms to the caregivers. MIMS also provides the possibility of easy add or remove sensors with a process called *association* in a way similar to the Bluetooth standard.

4.5.2 Application case studies

In this section, we present our sleep-related fall monitoring study among elderly. In our study, we used SHIMMER node for monitoring body movements and Enobio for monitoring sleep and brain activity during sleep.

Institutional monitoring

In institutions such as geriatric care clinics and nursing homes, there are communication and alerting systems in place which allow the caregivers to assist the patients. Hence, a patient usually rings a bell in order to request for assistance.

The passive monitoring devices like sensor pads embedded in beds can detect if a person is lying on a bed by measuring his body weight force applied on it. Sensor pads detect falls from a bed when the force is zero. Due to their simple nature, such devices are subject to many false positives. In fact, during night, a person can wake up from bed intentionally to go to the bathroom or accidentally sit hard on a bed. Moreover, the number of false positives increases especially if the patient is under specific medications.

To solve this problem, most of the institutions invest a lot of money to build a monitoring infrastructure with cameras which limit the monitoring of the subject's movements within the building. The MIMS platform is flexible and provides easy integration to existing systems along with all the advantages given by the use of wearable sensors. For example, a person can wake up in the morning and wear the SHIMMER node placed in a belt. In this case, the fall detection system will be able to monitor the movements for fall during the day. Before going to bed, the person can remove the belt and put on the Enobio sensor worn as a nightcap for sleep monitoring. In cases where the subjects have high risk of fall during night, they can wear both sensors with minimum invasiveness, due to the light weight and small size of the sensors. When a sensor device is not used, it can be recharged. When an abnormal event is detected, the MIMS will ensure that the alert will reach the caregivers quickly and they can use their own devices to query the health status of the patient anytime if they are warned of any potential emergency incidents. This way MIMS provides timely proactive monitoring.

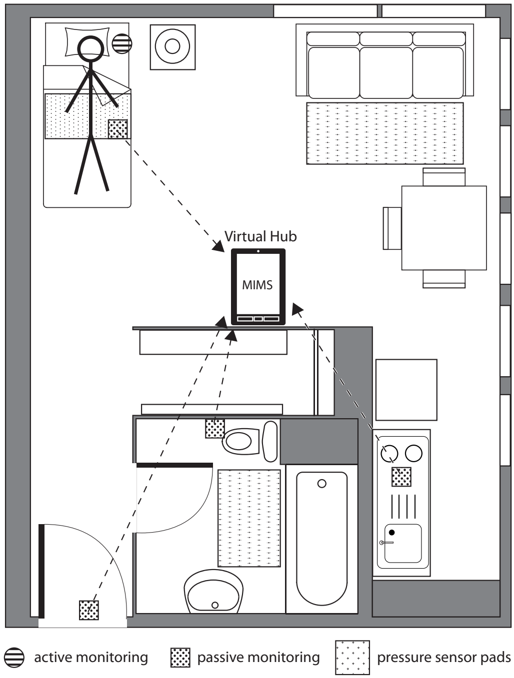


Figure 4.10: Example of layout for home monitoring.

Home monitoring

Home monitoring of an older adult is an alternative to more expensive hospitalization. The probability to fall at home is 60%, whereas only 10% of falls occur in nursing homes or other institutions [95]. This highlights how important it is to have fall prevention and detection system at home and MIMS enables easy installation in homes.

In fact, opposed to an institution which has its own communication system, the home monitoring system requires a gateway (Virtual Hub) with a specific communication unit support (e.g. cellular network, Internet) to connect the house with the external world [96].

Figure 4.10 shows an example of layout for home monitoring. Depending on the house structure and size, it may be necessary to extend the wireless coverage of the sensors to reach the Virtual Hub. This is true even if the Virtual Hub resides in a Smartphone since people usually move within the house without carrying it. Wireless repeaters spread inside the house (for example one per room) are able to extend the coverage of the network using ad-hoc or multi-hop paradigm.

Similar to the institutional monitoring scenario, MIMS integrates the passive monitoring sensors (if any in the house) with wearable sensors. The 24X7 monitoring would enable alerting caregivers and relatives of any abnormal health condition and emergency events. Moreover, if the wearable sensors are able to measure physiological characteristics such as heart-rate or blood pressure, a periodical check can be performed by a doctor from a remote location. During critical conditions, it can be translated into frequent checks and consultations before allowing the medical condition to progress.

4.5.3 Active monitoring

Though a well established correlation exists between the signals from the portions of the brain that control body movements to physiological conditions, using these correlations to develop proactive, preventative solutions are being explored in [91]. We derive from the body movements of the elderly and the multi-threshold measures of the brain-activity (alpha, beta, theta waveforms), an indicative factor to identify potential emergency incidents such as falls. SHIMMER and Enobio sensor devices are the two Active Monitoring Sensor Systems used and we refer to them as AMSS1 and AMSS2 respectively.

The AMMS1 system, represented by the SHIMMER device, monitors the human movements to detect fall and fall-like events. The prototype has been demonstrated to doctors, caregivers and industry. A set of tests for fall-like tests, such as sitting quickly on a chair, lying on a bed etc. and fall tests were performed using protections for the knees and the elbows [73]. A survey was conducted to study the different aspects of the system and we received positive feedback from doctors and caregivers. The patient has no interaction with the device worn on a belt, whereas the caregivers would receive alerts if a fall happens. We were conducting further study to enable the MIMS platform to manage more than one sensor and to integrate the AMMS1 with existing systems such as the Resident Information Management (RIM), adopted by some caregivers.

The AMMS2 system, represented by the Enobio device, is used to monitor the brain waves for the sleep-related falls. It is in a developing stage: with the help of doctors we are analyzing the different sleeping stages and its significance to identifying emergency events. An initial

SVM algorithm has been developed in order to study the feasibility of predicting sleep-related fall and we obtained some preliminary results. The prediction process was quick (less than 1 second). However dealing with real-time brain waves requires more time consuming tests and large data sets. The integration of both AMMS1 and AMMS2 enables data fusion and intelligence gathering for proactive fall monitoring in the MIMS platform. The preliminary results of this study can be found in [91].

4.6 Usability Study

Healthcare technologies are slowly entering into our daily lives, replacing old devices and techniques with newer ones. Although they are meant to help people, the reaction and willingness to use such new devices by the people are quite different, especially among elderly. A usability and acceptability study was conducted with two of our fall-detection monitoring sensor systems in a nursing home.

The contribution of this section tries to bridge this gap by reporting an exploratory study on how to make acceptable two wearable monitoring devices to a small group of elderly living in a long-term nursing home, by assessing their usability and adaptability. In the study, we monitored and detected falls using wireless sensors integrated in a MIMS system [91]. The subjects of the study were elderly affected by either AD or dementia and were monitored for two weeks. Usability and acceptability metrics have been defined for this study and evaluated with tests, which provided interesting insights on the human interactions with the new devices proposed.

4.6.1 Materials and methods

The monitoring system used in this study consisted of four sensing systems, as shown in Figure 4.11.

System 1 includes a WSN-based (SHIMMER device [9]), which can sense the movements of a person using an embedded accelerometer, can perform on-chip analysis and communicate with a base station wirelessly to report alert signals. The device is battery powered, has a small form factor, and is lightweight and therefore suitable to be worn near the waist of an elderly, for best results. It runs an algorithm for fall detection described in Section 4.4 and also available in [73].

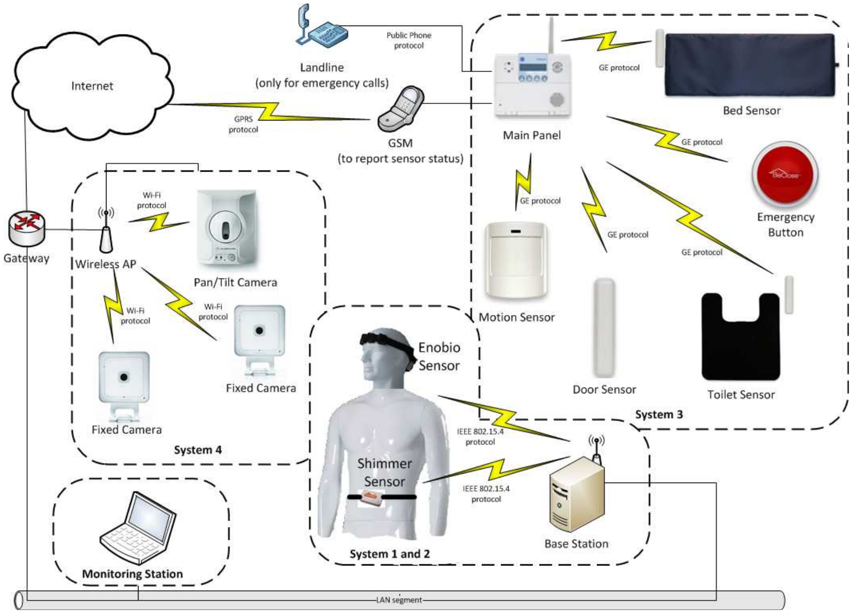


Figure 4.11: System overview.

System 2 is a wireless biosensor device (Enobio [94]) which is able to capture the brain activity of a person in real-time. Enobio is a wearable, modular and wireless electro-physiology sensor system made of four electrodes that capture and transfer data from four channels wirelessly to a base station, a transceiver/battery pack and two wired ear clips that act as references for sensed signals. Enobio can be worn like a hat and can record not only brain activity but also ECG, heart activity and Electrooculography (EOG), eye movement. It is used here to analyze sleep-EEG of a person in order to infer brain activity and EEG potentials during different stages of sleep and to identify brain signal patterns leading to a fall.

System 3 is composed of passive sensors such as pressure pads and volumetric motion detectors, placed in the care environment to monitor, if a person is lying on a bed, sitting on a chair or entering/leaving a room and other activities of daily living. *System 4* is a web camera monitoring system using wireless IP to continuously stream the video through the Internet, recording human activity (visual motion detection) over a selective region and over a specified observation period such as movements during the night.

In the case of resource constraints for sensing, processing, storing, and communicating, some computational operations can be delegated to a Virtual Hub, which is a base station running on a smart phone environment. The Virtual Hub receives also data from Systems 3 and 4 and is connected to the local healthcare information system with friendly graphical user interfaces to the caregivers.

System 1 and System 2 are wearable devices and therefore they offer more advantages in terms of continuous monitoring, cost and efficiency but may reveal some problems from usability and acceptability point of view. System 3 and System 4 are environmental systems. They require less interaction with the elderly and hence there are not many issues for usability from the patient's point of view. Although they are intrusive and have privacy implications, their usability study is not considered.

4.6.2 Measuring Usability and Acceptability

The AD and dementia subjects living in a home or nursing home require 24X7 continuous care and monitoring. Some of their regular activities during day time are: walking on a corridor, watching television, and with the help of caregivers, they have regular breakfast, lunch, dinner and medication; some subjects sleep a couple of hours during the afternoon after lunch. They are prone to disorientation and tendency to wander during any time of the day or night, and a probability to fall. Statistics show that fall among AD elderly is more than twice compared to general elderly population. To compensate psychomotor deficiencies, various medical equipment is used such as canes, crutches and wheelchairs. However, not all the subjects are able to understand (or remember) that they need to use them during their walking activity.

Usability in this context is defined as the level at which a device can assist a user without interfering with the activities of daily living. Acceptability is defined as the constraint which guides the designer to realize factors that satisfy one's need and therefore people's willingness to use.

The following are the metrics proposed to measure usability and acceptability:

- Willingness to use (WTU).

- Easiness to learn (ETL).
- Time to accept (TTA).
- Willingness to keep (WTK).
- Number of errors (NOE) due to incorrect interactions.
- Level of satisfaction (LOS).
- Interference with activities of daily living (IWA).

The evaluation criteria levels are tabulated in Table 4.9.

Metrics	**Levels**		
WTU	Low	Medium	High
ETL	Low	Medium	High
TTA	Short	Average	Long
WTK	Low	Medium	High
NOE	None	Few	Many
LOS	Low	Indifferent	High
IWA	None	Low	High

Table 4.9: Usability and acceptability parameters.

Generally, according to Human Engineering principles [], the design must follow the users' needs, their fear, mental models, ability for self-learning, social behavior, lifestyle and fashion tastes. Therefore a correct knowledge of the end users can be achieved only by observing them closely.

4.6.3 Results

A usability and adaptability analysis with respect to System 1 and System 2 have been carried out on four female subjects, aged 70 and above, and having AD. Their experiences in adapting to these systems and their different reactions have been observed and recorded [97]. Table 4.10 gives the behavior of the subjects, in term of the above usability and acceptability metrics along with some descriptions.

The study showed that with some modifications in wearing the devices, and after some convincing story about the importance of wearing the devices from care-givers/family, AD individuals can wear and benefit from these monitoring technologies. Comparing observation results from AD individuals to healthy individuals, and as expected, SHIMMER had higher level of usability and adaptability compared to Enobio, due to the ease in wearing it during the day time. The SHIMMER device could be worn effortless since it is worn on a waist. It was necessary to integrate it with everyday clothing so that it would have been considered as an accessory. Since the device is sensitive to movements, special wearing solutions had been considered while placing/removing the device and during specific times of the day (e.g. afternoon nap). The difference in dress code of women from men was another reason why the SHIMMER device is the best to be worn on the waist. However, it was required to place the node on a comfortable position on the waist and under the shirt/vest to hide the sensor and avoid touching/meddling/breaking while sitting on a bed or a chair. Though SHIMMER was found to be easily adapted by the subjects, a large number of false positives were recorded when the subjects shook the node not knowing it is a device. The study showed that more robustness is required to wear this device for continuous monitoring.

The Enobio system required significant effort to increase its usability among dementia individuals. It was worn like a hat during sleep in the night with two reference clips on the ears. Since it was not comfortable to perform sleep tests with a bulky battery pack and transmitter lying on the nape, many adaptability changes were made. The first problem addressed was to improve the comfort level (willingness to use - WTU) during eight hours sleep. Then Enobio had to be least intrusive as possible so that people would not feel embarrassed to wear it (willingness to keep -WTK). Hence, to guarantee acceptability by elderly subjects, the Enobio system was hidden in hats to make it comfortable wearing it during sleep shown in (Figure 4.13). Note that elderly people often wore hats during night to keep themselves warm and cozy.

An important factor to be considered is the color used with the sensor devices. Colors have different impacts and meanings in one's space or environment. In fact, providing the Enobio with a pink cap increased the acceptability of one of the subjects (Subject #1) since it was her favorite color. In general, bright colors or color combinations can help elderly people who are visually impaired to better understand the object they are watching. For example, colorful door could help a subject to find his/her own room. Handrails colored brighter than walls can help people to walk in a corridor. It was found that warm colors (active) such as orange hues like red, pink, yellow, brown, and their shades are favorable for identifying objects. The color also improves their comfort level and keeps them calm. Cool colors (passive) such as blue, green, purple and their shades are useful to give an impression of coolness, discretion and serenity and give the illusion of larger perspective. Neutral tones are shades from beige, white, black and grey. In the study, it was observed that when the devices came in their favorite colors, it became easily adapted.

4.6.4 Lessons learned

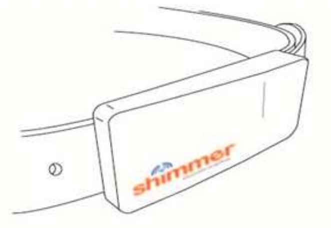
The study showed that ergonomic and aesthetic modifications were necessary for System 1 and System 2 to enable easy adaptation and acceptability. The subjects wearing the sensors were elderly; therefore the sensors should suit specific ergonomic, aesthetic, easy wear and handle conditions. Analysis of the subject's dress was fundamental to find a comfortable and wearable solution. On the basis of their age, elderly people are attached to a specific aesthetic dress code, characteristic of their likes/dislikes. They prefer simple, loose and comfortable dress and therefore the focus should be on a retro style. Such loose dresses make difficult to put wearable devices close to the body. At the same time, these devices should not cause itching, rashes or skin diseases if worn too tight. Since a SHIMMER has to be worn on the waist, a natural solution was to attach the device on a belt.

Unfortunately not every elderly wears a belt, especially women. Moreover, for safety considerations, the devices must be prevented from choking the individual. Therefore, the following two were proposed:

1. The SHIMMER device was integrated onto a belt buckle (see Figure 4.12(a)). The buckle is designed in an aesthetic way suitable for both men and women. A leather style gives it a retro aesthetic, perfect for the elderly.

2. The SHIMMER device was attached to a Velcro stripe made as belt. Two small stripes were crossed to hold the device safe and wear on the waist (see Figure 4.12(b)).

To improve the usability of the Enobio sensor during sleep, the battery/transmitter was moved from the back of head/neck to the top of the head on a belt, thus increasing the ergonomic and comfort level. To provide an aesthetic feature, the Enobio sensors were embedded in a cap



(a) Integrated into a belt buckle.

(b) Attached to a Velcro stripe.

Figure 4.12: SHIMMER modifications.

(bonnet style), with a light texture to prevent sweating (see Figure 4.13). Elderly usually wear caps in the night to keep them warm and hence when fitted onto a hat it was accepted without much difficulty. However some AD individuals took the hat off often during night before getting into a deep sleep. This identified problem was solved to some extent by allowing them to wear the hat during the day time without the sensing device to make them feel comfortable wearing the hat. In this way they did not care much when the device was embedded in the hat during their sleep in the night.

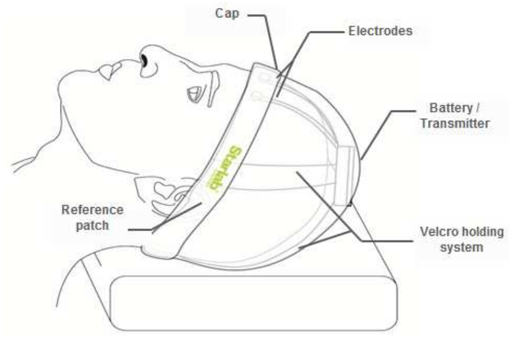


Figure 4.13: The modified cap was more comfortable during sleep/going to bed.

In general, the study showed that the design and development of a monitoring device must consider its target subjects' usage before it can be broadly used. A prototype, or a very basic version of the system, will not take into account the interaction with non-technical people or its use in a generic environment. Further work on the system development must be accomplished to reap all benefits from the technology. For example, the SHIMMER prototype designed to work close to the base station (within 100 meters of range), did not take into consideration the elderly subjects walking along the corridor. Hence, the sensor devices kept sending false positives when the sensors went out of range. After this study, the system architecture was modified to increase the range of operation.

In addition, more SHIMMER devices should be capable of sharing the same base station. Since elderly walk a lot on the corridors, spend some time in their room and in the common space, there is a need to extend the coverage of the sensors for continuous monitoring. A solution could be to configure each sensor to run a routing protocol to forward information to the base station. However, this could force each node to frequent communications other than continuous sampling. As a result, the battery lifetime could be reduced drastically.

Another solution identified is to distinguish two types of nodes:

1. Sensing nodes, worn by the individuals.
2. Forwarding nodes, placed in fixed positions in the nursing home connected to the power outlets.

Sensing nodes can communicate to the base station through the forwarding nodes, by selecting the closest node as first hop. In this case, the advantage was that each sensing node will use their radio only for sending data. Forwarding nodes can run a routing algorithm that guarantees no packet loss; this is important because the system cannot tolerate an undelivered alarm. The forwarding nodes can have a fixed location and therefore they can be configured to determine a coarse location of each person within the building.

From a physical point of view, the device worn by the patient must be robust and waterproof to avoid accidental damages (e.g. the sensor can fall to a sink full of water or it can be thrown away or tampered). The sensor should also provide a switch or a special combination of buttons in order to be activated and deactivated by the nurse before placing the device on the waist or before charging it respectively. A battery indicator is another element that would help nurses to identify if the device needs to be charged. The caregivers interface is fundamental to understand how to assign each sensing node to a person, to check the history and general status. In particular, the sensing nodes should periodically send a message to signal that they are working correctly and, in this case they will also provide an update to a localization system.

The Enobio system in our experiments was used for collecting data without a real-time analysis of the brain activity on-chip. Similar to the SHIMMER case, Enobio should be able to perform local processing and send data to a base station though a forwarding network. Further steps are required, especially from the manufacturing point of view, to make it wearable easily before testing with a larger number of residents sleeping in their rooms during the night. When Enobio was formerly tested with healthy subjects, the subjects were usually aware of the sleep study and reported a slight discomfort while moving in the bed during night. They were always conscious of wearing the hat and often their movements were restricted by it in finding a good position to sleep. During this study, the elderly people affected by dementia showed different reactions. They did not understand about the sleep tests performed. Some thought that the hat was to keep them warm. Although some subjects did not move in the bed, assuming a supine position for the entire night, some subjects kept removing the hat. As a result, their sleep was interrupted by the testing, as the hat had to be put back to the correct position again and again.

Shimmer	Subject # 1	Subject #2	Subject #3	Subject #4
Day 1 :	WTU: Low ETL: High TTA: n/a WTK: Low NOE: Low LOS: Low IWA: None Didn't want to wear it. After 30 minutes the subject forgot about it and was happy to remove it towards the end of day.	WTU: n/a ETL: Low TTA: Short WTK: High NOE: Many LOS: High IWA: None No resistance. Then the patient removed it and played with it. At the end, he didn't want to give it back.	WTU: n/a ETL: High TTA: Short WTK: Medium NOE: None LOS: Indifferent IWA: None No resistance.	WTU: High ETL: High TTA: Short WTK: High NOE: None LOS: Indifferent IWA: None No resistance. The patient was already using a pouch around the waist.
Day 2 :	WTU: Medium WTK: Medium NOE: Low LOS: Indifferent Light resistance to wear it initially.	WTU: High NOE: Few Happy to wear it.	LOS: None Totally indifferent.	WTK: Medium Indifferent.
Day 3 :	WTU: High TTA: long WTK: High NOE: None Happy to wear it. The family members convinced the subject to wear it saying it was meant for stomach pain relief.	NOE: None LOS: Indifferent Indifferent.	(not tested)	(not tested)
Enobio	**Subject # 1**	**Subject #2**	**Subject #3**	
Night 1 :	WTU: High ETL: Low TTA: n/a WTK: Low NOE: Few LOS: Low IWA: High No resistance at the beginning (the patient loves hats). The patient removed it after some time.	WTU: Medium ETL: Low TTA: n/a WTK: Low NOE: High LOS: Low IWA: None The patient kept for some minutes then removed it. After three times the patient did not want to wear it anymore.	WTU: Low ETL: High TTA: Short WTK: Medium NOE: None LOS: Low IWA: Low No resistance but the patient did not like it.	
Night 2 :	NOE: Many IWA: Low The subject wore it but moved it many times.	WTU: Low WTK: Low NOE: High Removed it many times during night.	WTU: Medium WTK: Medium LOS: Medium No resistance.	
Night 3 :	ETL: Medium TTA: Average WTK: Medium NOE: None LOS: Indifferent The subject was willing to wear it and did not move at all during night.	TTA: Average WTK: Medium NOE: None LOS: Indifferent The subject was monitored from bed-side all night to make sure she wore it.	(not tested)	

Table 4.10: Behavior of the subjects towards new technologies.

Chapter 5

Conclusion and future work

WSN are at the basis of new and innovative applications. Furthermore, some today's applications can be improved with the help of sensor networks. To this end, companies and organizations all over the world are massively making investments. As a result, many researchers studied solutions for applications in the fields of Industry, Construction, Home Automation, Logistics, Territory Monitoring, Healthcare, Military, and Security. However, most of the research works rely on strong assumptions which make a system work only from a theoretical point of view. There is a need to bridge the gap between the theory and the practical implementation of such systems. To better understand the reasons of this gap, two representatives categories of WSN applications have been considered, namely Localization and Monitoring. For each of them we stated some non-functional issues, related to the specific application and some functional issues to be addressed in most of the WSN applications, such as energy consumption, fault tolerance, scalability, production costs, and deployment. We showed that to deploy a real system, there are some necessary elements to be considered such as reliability, robustness, usability, and acceptability. This can be achieved by starting assessing algorithms through their implementation in working prototypes and observing the interactions of the WSN with the other actors.

The specific characteristics of Localization and Monitoring have been analyzed, with the long-term goal to implement (or give the tools to implement) one or more real world applications.

In Localization we:

1. Considered and analyzed range-free techniques to avoid the estimation of the range and angle measurements.
2. Studied the target application, that is the localization of shipping containers.
3. Analyzed the problems we wanted to avoid such as having a fixed infrastructure or using a GPS.
4. Designed a more reliable range-free localization technique working in presence of missing information.

The localization solution proposed exploits sensor networks to overcome the problems of currently available systems using GPS and RFID. In fact, there is no need of line-of-sight as for GPS to localize containers or a fixed infrastructure as for RFID, to be able to retrieve information from each container. The use of sensor networks enabled also to receive data in real-time, providing an up-to-date status of the containers placement.

With a first algorithm, namely strawman approach, the localization algorithm guarantees scalability. Then, with the introduction of the Integer Linear Programming algorithm for the fault tolerance, the system was made robust and resilient. Geometrical constraints where considered to reduce the space of possible solutions of the localization problem. Localization techniques based on the strength of the received signal are characterized by high error levels, but the discretization of node positions made the localization process simple and scarcely sensible to RSS errors. This can be applied to other localization scenarios characterized by similar geometrical constraints. Moreover, two scenarios are considered for which the overall localization rate is increased by 4.45 and 2.4 times with respect to the strawman approach.

Even thought the presented algorithm and overall approach to localization has been strictly focused on container localization we believe that there are a number of applications in the field of logistics that can drastically improve the way they operate. For example, an indoor warehouse with goods placed in boxes can take advantage of sensor networks to locate all the items in real-time. If the boxes are organized on shelves, the system can be simplified and the number of sensors reduced. In general, this approach can be used in most of the systems whose localization can be modeled using geometrical constraints. Moreover, in a near future, the Internet of Things [66] will finally take place and it is likely that also the ordinary shipping containers will be enhanced with computing and communication capabilities. For this reason, it is worth to start working now to study and expand the possibilities offered by WSN, well beyond localization (security, tracking, monitoring of goods, etc.).

In Monitoring of human movements for the detection falls among elderly we:

1. Studied the bio-mechanics of human movements.
2. Analyzed the end-users requirements.
3. Evaluated which movements sensor use.
4. Considered the mobility of the sensors.
5. Designed an algorithm to be assessed according to some performance indicators.
6. Proposed a minimally invasive monitoring sensing platform for sensors deployment.
7. Studied and improved the usability and acceptability of the sensing devices worn by people.

Monitoring is closely related to the targeted application. In our case, the development of a fall detection system required a non-negligible warm-up time to fully understand the problem of falls and the most relevant approaches adopted. Since there was not a standard evaluation of the different systems, we defined also some standard testing guidelines.

Sensor networks can be used to monitor the acceleration of the user, but the number of the sensors to be worn can affect the usability aspect of the system. On the other hand, with a small number of sensors, the systems have less information and can experiment many false positives. We designed a fall detection system based on a single wearable device able to filter false alarms by recognizing the most common fall-like Activities of Daily Living. The device is worn on the waist and has low energy consumption because it uses low sampling frequency (50 Hz) and it communicates with the base station only when an alarm is detected. The performance of the fall detection system has been evaluated in a testing environment, with good results. A further study has been done to define a Minimally Invasive Monitoring Sensing (MIMS) platform in order to provide a schema in deploying the system. MIMS guarantees interoperability with the existing infrastructure and with other sensing devices. To this end we considered not only SHIMMER sensors for movements monitoring, but also Enobio device for brain wave analysis during sleep time.

We had the opportunity to study the feasibility of the platform and the usability of the sensors among AD elderly, in a long-term care home. It was found that the time to adapt was different among AD elderly, which relied on the users likes and dislikes (e.g. colors, dress). It was also found that feedback from such testing would enable modifications to system in order to improve performance and shorten time of adaptation. In fact, it was learned that the adaptation of new technologies is possible, even in the Alzheimer's Disease elderly care. However, it required also non-technological elements to be considered from both subjects and the environment infrastructure for quick adaptability and benefit to improved care.

There are a wide number of small, portable devices already used for health diagnosis or check. They are effective but quite expensive and they are not interconnected. With the flexibility and connectivity of a wireless sensing device, it is possible to overcome these problems and provide a better and responsive service to people. In particular, the extension and testing of the MIMS architecture will be performed for real time acquisition of physiological measurements given by neuro-sensing devices, which need to be studied in depth to reduce their main issue, the invasiveness. This could help not only toward the prediction of falls, but also to the discovery of early symptoms of cognitive degeneration and thus diagnose Alzheimer Disease sufficiently in advance to be treated.

Bibliography

[1] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.

[2] N. R. Council and N. R. Council, *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. National Academies Press, October 2001.

[3] M. Weiser, "The computer for the twenty-first century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[4] F. Zhao and L. J. Guibas, *Wireless sensor networks : an information processing approach*. The Morgan Kaufmann series in networking., Amsterdam ; San Francisco: Morgan Kaufmann, 2004. 2004301905 Feng Zhao, Leonidas J. Guibas. Includes bibliographical references (p. 323-345) and index.

[5] S. Abbate, M. Avvenuti, P. Corsini, and A. Vecchio, "Localization of shipping containers in ports and terminals using wireless sensor networks," in *Proc. of the International Conference on Computational Science and Engineering*, vol. 2, pp. 587 –592, Aug. 2009.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," *SIGOPS Oper. Syst. Rev.*, vol. 34, no. 5, pp. 93–104, 2000.

[8] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN '05: Proceedings of the 4th interna-*

tional symposium on Information processing in sensor networks, (Piscataway, NJ, USA), IEEE Press, 2005.

[9] Realtime Technologies LTD, "Shimmer sensing health with intelligence modularity, mobility and experimental reusability," 2009.

[10] TinyOS Community, "Tinyos community forum," 2008.

[11] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesc language: A holistic approach to networked embedded systems," in *PLDI '03: Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*, (New York, NY, USA), pp. 1–11, ACM, 2003.

[12] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 263–270, ACM, 1999.

[13] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008.

[14] P. J. Marron, K. Stamatis, and D. Minder, *Research Roadmap 2009*. CONET (Cooperating Objects Network of Excellence), 2009.

[15] E. Petriu, N. Georganas, D. Petriu, D. Makrakis, and V. Groza, "Sensor-based information appliances," *Instrumentation and Measurement Magazine, IEEE*, vol. 3, pp. 31–35, Dec 2000.

[16] T. H. Keitt, D. L. Urban, and B. T. Milne, "Detecting critical scales in fragmented landscapes," *Conservation Ecology*, p. 4 (online), 1997.

[17] M. P. Hamilton and M. Flaxman, "Scientific data visualization and biological diversity: new tools for spatializing multimedia observations of species and ecosystems," *Landscape and Urban Planning*, vol. 21, no. 4, pp. 285 – 287, 1992.

[18] A. Ollero and L. Merino, "Control and perception techniques for aerial robotics," *Annual Reviews in Control*, vol. 28, no. 2, pp. 167 – 178, 2004.

[19] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on*, pp. 42–47, July 2008.

[20] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy, "Fall detection - principles and methods," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 1663–1666, Aug. 2007.

[21] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[22] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang, "Design considerations for distributed microsensor systems," in *Custom Integrated Circuits, 1999. Proceedings of the IEEE 1999*, pp. 279–286, 1999.

[23] S. Abbate, M. Avvenuti, A. Biondi, and A. Vecchio, "Estimation of energy consumption in wireless sensor networks using tinyos 2.x," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pp. 842 –843, jan. 2011.

[24] G. Hoblos, M. Staroswiecki, and A. Aitouche, "Optimal design of fault tolerant sensor networks," in *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*, pp. 467–472, 2000.

[25] R. Stoleru, T. He, and J. A. Stankovic, *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks (Advances in Information Security)*, ch. Range-Free Localization. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[26] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 166–179, ACM, 2001.

[27] J. D. Gibson, *Mobile Communications Handbook*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1999.

[28] R. Grossmann, J. Blumenthal, F. Golatowski, and D. Timmermann, "Localization in zigbee-based sensor networks."

[29] D. Niculescu and B. Nath, "Ad hoc positioning system (aps) using aoa," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, pp. 1734–1743 vol.3, March-3 April 2003.

[30] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *2nd International Workshop on Information Processing in Sensor Network (IPSN)*, April 2003.

[31] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2652–2661 vol.4, March 2004.

[32] D. of Homeland Security. `http://www.dhs.gov`.

[33] R. Craddock and E. Stansfield, "Sensor fusion for smart containers," in *The IEE Seminar on Signal Processing Solutions for Homeland Security*, p. 12 pp., Oct. 2005.

[34] S. J. Kim, G. Deng, S. Gupta, and M. Murphy-Hoye, "Intelligent networked containers for enhancing global supply chain security and enabling new commercial value," in *Proc. of the 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE)*, pp. 662 –669, Jan. 2008.

[35] J. Bian, R. Seker, S. Ramaswamy, and N. Yilmazer, "Container communities: anti-tampering wireless sensor network for global cargo security," in *Proc. of the 17th Mediterranean Conference on Control and Automation*, pp. 464–468, Jun. 2009.

[36] S. Bukkapatnam, S. Mukkamala, J. Kunthong, V. Sarangan, and R. Komanduri, "Real-time monitoring of container stability loss using wireless vibration sensor tags," in *Proc. of the IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 221 –226, Aug. 2009.

[37] Y. Jarma, G. Karbaschi, M. de Amorim, F. Benbadis, and G. Chelius, "Vaps: Positioning with spatial constraints," in *Proc. of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops (WoWMoM)*, pp. 1–8, Jun. 2009.

[38] V. Recagno, A. Derito, and R. Nurchi, "Mocont: a new system for automatic identification and location of containers," in *Proc. of the Vehicular Technology Conference*, vol. 4, pp. 2609 –2613 vol.4, 2001.

[39] R. Schmidt, A. Gagnon, and D. Allcock, "Maritime containers tracking trial results," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 24, pp. 10–14, Sep. 2009.

[40] S. Mahlknecht and S. Madani, "On architecture of low power wireless sensor networks for container tracking and monitoring applications," in *Proc. of the 5th IEEE International Conference on Industrial Informatics*, vol. 1, pp. 353 – 358, Jun. 2007.

[41] E. Herrera, M. Ribo, A. Pinz, R. Quiros, and G. Fabregat, "Wearable system for the localization of shipping containers in a maritime terminal," in *Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE*, pp. 611–616, April 2006.

[42] N. Patwari, J. Ash, S. Kyperountas, I. Hero, A.O., R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, pp. 54–69, July 2005.

[43] Y. Depeursinge, J. Krauss, and M. El-khoury, "Device for monitoring the activity of a person and/or detecting a fall, in particular with a view to providing help in the event of an incident hazardous to life or limb," May 2001.

[44] M. A. Clifford, R. L. Borras, and L. Gomez, "System and method for human body fall detection," July 2007.

[45] Y. Lee, J. Kim, M. Son, and M. Lee, "Implementation of accelerometer sensor module and fall detection monitoring system based on wireless sensor network," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 2315–2318, Aug. 2007.

[46] U. Lindemann, A. Hock, M. Stuber, and W. Keck, "Evaluation of a fall detector based on accelerometers: A pilot study," *Medical and Biological Engineering and Computing*, vol. 43, no. 5, pp. 548–551, 2005.

[47] C.-C. Wang, C.-Y. Chiang, P.-Y. Lin, Y.-C. Chou, I.-T. Kuo, C.-N. Huang, and C.-T. Chan, "Development of a fall detecting system for the elderly residents," in *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on*, pp. 1359–1362, May 2008.

[48] M. Prado-Velasco, M. G. del Rio-Cidoncha, and R. Ortiz-Marin, "The inescapable smart impact detection system (isis): An ubiquitous and personalized fall detector based on a distributed divide and conquer strategy," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 3332–3335, Aug. 2008.

[49] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *BSN '09: Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, (Washington, DC, USA), pp. 138–143, IEEE Computer Society, 2009.

[50] N. Noury, P. Barralon, G. Virone, P. Boissy, M. Hamel, and P. Rumeau, "A smart sensor based on rules and its evaluation in daily routines," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 4, pp. 3286–3289 Vol.4, Sept. 2003.

[51] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron, "Monitoring behavior in home using a smart fall sensor and position sensors," in *Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000*, pp. 607–610, 2000.

[52] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *ISWC '07: Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers*, (Washington, DC, USA), pp. 1–4, IEEE Computer Society, 2007.

[53] M. Mathie, J. Basilakis, and B. Celler, "A system for monitoring posture and physical activity using accelerometers," in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, vol. 4, pp. 3654 – 3657 vol.4, 2001.

[54] J. Hwang, J. Kang, Y. Jang, and H. Kim, "Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly," in *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, vol. 1, pp. 2204 –2207, 1-5 2004.

[55] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, "Perfalld: A pervasive fall detection system using mobile phones," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 292 –297, march 2010.

[56] F. Sposaro and G. Tyson, "ifall: An android application for fall monitoring and response," in *Engineering in Medicine and Biology Soci-*

ety, 2009. EMBC 2009. Annual International Conference of the IEEE, pp. 6119 –6122, 3-6 2009.

[57] S. Abbate, M. Avvenuti, P. Corsini, B. Panicucci, M. Passacantando, and A. Vecchio, “An integer linear programming approach for radio-based localization of shipping containers in the presence of incomplete proximity information,” *IEEE Transactions on Intelligent Transportation Systems*, (Accepted) 2012.

[58] C.-I. Liu, H. Jula, and P. Ioannou, “Design, simulation, and evaluation of automated container terminals,” *IEEE Trans. on Intell. Transp. Syst.*, vol. 3, pp. 12 –26, Mar. 2002.

[59] P. Chen, Z. Fu, A. Lim, and B. Rodrigues, “Port yard storage optimization,” *IEEE Trans. Autom. Sci. Eng.*, vol. 1, pp. 26 – 37, Jul. 2004.

[60] Z. He, J. Liu, H. Ma, and P. Li, “A new automatic extraction method of container identity codes,” *IEEE Trans. on Intell. Transp. Syst.*, vol. 6, pp. 72 – 78, Mar. 2005.

[61] A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero, and G. Soncin, “Modeling and feedback control for resource allocation and performance analysis in container terminals,” *IEEE Trans. on Intell. Transp. Syst.*, vol. 9, pp. 601 –614, Dec. 2008.

[62] H. Durrant-Whyte, D. Pagac, B. Rogers, M. Stevens, and G. Nelmes, “Field and service applications - an autonomous straddle carrier for movement of shipping containers - from research to operational autonomous systems,” *IEEE Robot. Autom. Mag.*, vol. 14, pp. 14 –23, Sep. 2007.

[63] S. Schaefer, “Secure trade lane: a sensor network solution for more predictable and more secure container shipments,” in *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, (New York, NY, USA), pp. 839–845, ACM, 2006.

[64] W. Lang, R. Jedermann, D. Mrugala, A. Jabbari, B. Krieg-Brü andckner, and K. Schill, “The “Intelligent Container” – A Cognitive Sensor Network for Transport Management,” *IEEE Sensors J.*, vol. 11, pp. 688 –698, Mar. 2011.

[65] J. O. Lauf and D. Gollmann, “Monitoring and security of container transport,” in *In Proc. of New Technologies, Mobility and Security*, Springer, 2007.

[66] IoT-A, "Internet of things." `http://www.iot-a.eu/`, 2012.

[67] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks," in *Proc. of the workshop on real-world wireless sensor networks (REALWSN)*, (New York, NY, USA), pp. 1–5, ACM, 2008.

[68] E. Elnahrawy, X. Li, and R. Martin, "The limits of localization using signal strength: a comparative study," in *Proc. of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, pp. 406 – 414, Oct. 2004.

[69] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[70] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, Nov. 2002.

[71] *ILOG CPLEX 9.1 User's Manual*. ILOG Inc., 2005.

[72] S. Abbate, M. Avvenuti, P. Corsini, J. Light, and A. Vecchio, *Monitoring of Human Movements for Fall Detection and Activities Recognition in Elderly Care Using Wireless Sensor Network: a Survey*, ch. 9, pp. 147–166. Rijeka, Croatia: InTech, December 2010.

[73] S. Abbate, M. Avvenuti, G. Cola, P. Corsini, J. V. Light, and A. Vecchio, "Recognition of false alarms in fall detection systems," in *1st IEEE International Workshop on Consumer eHealth Platforms, Services and Applications (CCNC'2011 Workshop CeHPSA)*, (Las Vegas, NV, USA), pp. 538–543, 1 2011.

[74] M. Gibson, R. Andres, B. Isaacs, T. Radebaugh, and J. Worm-Petersen, "The prevention of falls in later life. a report of the kellogg international work group on the prevention of falls by the elderly.," *Danish Medical Bulletin*, vol. 34, no. 4, pp. 1–24, 1987.

[75] A. Chapman, *Biomechanical Analysis of Fundamental Human Movements*. United States: Human Kinetics, 2008.

[76] C.-C. Yang and Y.-L. Hsu, "Algorithm design for real-time physical activity identification with accelerometry measurement," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pp. 2996 –3000, 5-8 2007.

[77] A. J. Campbell, M. J. Borrie, G. F. Spears, S. L. Jackson, J. S. Brown, and J. L. Fitzgerald, "Circumstances and Consequences of Falls Experienced by a Community Population 70 Years and over during a Prospective Study," *Age Ageing*, vol. 19, no. 2, pp. 136–141, 1990.

[78] L. A. Lipsitz, P. V. Jonsson, M. M. Kelley, and J. S. Koestner, "Causes and Correlates of Recurrent Falls in Ambulatory Frail Elderly," *Journal of Gerontology*, vol. 46, no. 4, pp. M114–M122, 1991.

[79] S. Lord, J. Ward, P. Williams, and K. Anstey, "An epidemiological study of falls in older community-dwelling women: the Randwick falls and fractures study," *Australian Journal of Public Health*, vol. 17, no. 3, pp. 240–245, 1993.

[80] T. Masud and R. Morris, "Epidemiology of falls," *Age Ageing*, vol. 30, no. 4, pp. 3–7, 2001.

[81] M. E. Tinetti, M. Speechley, and S. F. Ginter, "Risk factors for falls among elderly persons living in the community," *New England Journal of Medicine*, vol. 319, pp. 1701–1707, December 1988.

[82] C. Bradley, S. Pointer, A. I. of Health, Welfare., and F. U. of South Australia., *Hospitalisations due to falls by older people, Australia, 2005-06 / Clare Bradley, Sophie Pointer*. Australian Institute of Health and Welfare, Canberra :, 2009.

[83] S. Lord, S. C., and H. Menz, "Falls in older people: Risk factors and strategies for prevention," *Ageing and Society*, vol. 21, no. 05, pp. 667–675, 2001.

[84] R. Englander, F.and Terregrossa and T. Hodson, "Economic dimensions of slip and fall injuries," *Journal of Forensic Sciences (JOFS)*, vol. 41, no. 05, pp. 733–746, 1996.

[85] Massachusetts Department of Public Health, *Unintentional Fall-Related Injuries among Massachusetts Older Adults*. Commonwealth of Massachusetts, Department of Public Health, 2008.

[86] M. Kangas, A. Konttila, I. Winblad, and T. Jamsa, "Determination of simple thresholds for accelerometry-based parameters for fall detection," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 1367 –1370, aug. 2007.

[87] M. Kangas, I. Vikman, J. Wiklander, P. Lindgren, L. Nyberg, and T. Jamsa, "Sensitivity and specificity of fall detection in people aged 40 years and over.," *Gait Posture*, 2009.

[88] A. K. Bourke, J. V. O'Brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm.," *Gait & posture*, vol. 26, no. 2, pp. 194–9, 2007.

[89] University of Pisa and IMT Lucca, "Fall detection database." `http://wsn.iet.unipi.it/falldb/`, 2011.

[90] M. Avvenuti, C. Baker, J. Light, D. Tulpan, and A. Vecchio, "Non-intrusive patient monitoring of alzheimer's disease subjects using wireless sensor networks," *Privacy, Security, Trust and the Management of e-Business, World Congress on*, vol. 0, pp. 161–165, 2009.

[91] S. Abbate, M. Avvenuti, and J. Light, "Mims: A minimally invasive monitoring sensor platform," *IEEE Sensors Journal*, vol. PP, no. 99, p. 1, 2011.

[92] E. Mattila, I. Korhonen, J. Merilahti, A. Nummela, M. Myllymaki, and H. Rusko, "A concept for personal wellness management based on activity monitoring," in *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pp. 32 –36, 30 2008-feb. 1 2008.

[93] M. Alwan, D. Mack, S. Dalal, S. Kell, B. Turner, and R. Felder, "Impact of passive in-home health status monitoring technology in home health: Outcome pilot," in *Distributed Diagnosis and Home Healthcare, 2006. D2H2. 1st Transdisciplinary Conference on*, pp. 79 –82, 2006.

[94] Starlab - Living Science, "Enobio," 2011.

[95] American Academy of Orthopaedic Surgeons, "Fall prevention facts," August 2009.

[96] I. Korhonen, J. Parkka, and M. Van Gils, "Health monitoring in the home of the future," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 22, no. 3, pp. 66 – 73, 2003.

[97] S. Abbate, M. Avvenuti, and J. Light, "Usability study of wireless monitoring technologies among alzheimer's disease elderly population.," (Under revision) 2012.