Doctorate in Computer Science and Automation

Department of Engineering

XXIX Ph.D Cycle

Hybrid Rescue Team Localization System

Ph.D Student Federica Inderst

_____

Supervisor Prof.ssa Federica Pascucci

_____

Ph.D Coordinator Prof. Stefano Panzieri

_____

"Chi più in alto sale, più lontano vede.
Chi più lontano vede, più a lungo sogna."
W. Bonatti

Abstract

Localization and tracking support is useful in many contexts and become crucial in emergency response scenarios. Being aware of team location is one of the most important knowledge for incident commander. Team localization and tracking open new prospects to increase safety, efficiency, and also to decrease mission time: localized rescue personnel can be better coordinated, commanded and guided. Moreover a reliable localization system reduces the possibility of disorientation and failure to locate victims, that are contributing factors to rescuer deaths. In this work a technique for first responders localization support in emergency indoor scenarios is presented. The peculiarity of the proposed system relies on the integration between human operators and robots in a hybrid team. The proposed localization technique takes advantage from both passive and active infrastructure composed by nodes that can provide information about their positions and the resources/hazards nearby. The main contribution of the thesis is the development and the testing of a localization system for hybrid rescue teams, able to handle both a priori knowledge, i.e., maps of the environment, with large uncertainties and information updated during rescue tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"...nothing is so much wanted and desired at sea, as the discovery of the longitude, for the safety and quickness of voyages, the preservation of ships, and the lives of men..."
The Longitude Act, 1714

1.1 Overview

The problem of locating a person or an object in an environment has been largely investigated: location allows to a human being to perceive and understand the surrounding world. To this end many different methods have been employed over the recent centuries.

Solving the *Longitude problem* was one of the greatest scientific challenges to European governments in Eighteenth Century, since it was a significant obstacle to global navigation and maritime supremacy. This problem was considered so intractable and so important that the British Parliament established in *The Longitude Act* a £20,000 reward (corresponding to about 3.000.000$ in 2016) for anyone who could find a practical way of determining longitude at sea having an accuracy not greater than half a degree (equates to 30 nautical miles - 56 Km).

North/South latitude could be easily and faithfully reckoned by measuring the angle of prominent stars. On the contrary, West/East longitude estimation was more difficult to determine, since time and longitude are coupled. Nevertheless, it was only in 1765

that the self-taught inventor John Harrison crafted the first reliable marine chronometer, able to lose less than 5 seconds every ten days, winning the prize.

Nowadays, tracking the location of a person or an object is primarily done by exploiting satellite-based systems. The first deployed is the Global Positioning System (GPS). Discontinuing the use of Selective Availability (an intentional degradation of public GPS signals implemented for national security reasons) improved the predicted accuracy of GPS for civilian users from within 100 m to within 20 m. This performance boost enabled GPS to be applied in its most basic form to a variety of civilian activities - land, sea, air, and space - where it could not previously.

It is useful to underline that the need for localization solutions is not just confined to vehicles or transportation services. In the Internet of Things age, due to the ubiquitous availability of powerful mobile computing devices, the realization of personalized context/location-aware applications has become an active field of research. However, the natural habitats of human individuals, the indoor environments, are inaccessible for the signals of the GPS satellites. Accurate indoor localization has the potential to transform the way people navigate indoors in a similar way that GPS transformed the way people navigate outdoors.

Over the last 15 years, several indoor localization technologies have been proposed by both academia and industry. In Chapter 2 an overview of the most promising ones is reported. However, large-scale deployments or off-the-shelf solutions are still to be achieved.

1.2 Scope of the work

Different indoor localization solutions have been based on different information sources according to the constraints of the particular application scenario. Mass-market applications for indoor positioning require the use of standard devices without supplementary physical components: the consumer market cannot easily accept major modifications to mobile phones in order to include positioning functions. To cope with this problem, most of the proposed approaches exploits the availability of an external infrastructure, pre-deployed in the environment (e.g., Wi-Fi signal) and applies trilateration or fingerprinting techniques to determine the position of a user.

However, in an emergency scenario, the requirements are quite severe [19]. Indoor localization for first responders cannot rely only on an external infrastructure, that could result unavailable during emergencies. As described in Chapter 2, many localization and navigation solutions have been proposed for rescue framework, based on simple techniques coupled with low-tech while robust equipment. In fact, due to hard operating conditions (such as high temperature, thick smoke, noise, and obstacles) high tech systems exploiting radio, ultrasound, and laser signals cannot be adopted. Furthermore, the devices have to be designed as robust as possible to withstand rough handling and very high temperatures.

The lack of an efficient indoor localization method for rescuers stimulates research to the design of alternative localization systems specifically adapted to emergencies. It is useful to underline that location awareness is an important aspect in disaster response. In our vision, the effectiveness of incident command operations depends heavily on tracking first responders, in field injured persons triaging, and physical environment monitoring [34, 35]. During emergency, first responders face unknown and complex situations in which classical perception schemes fail or are significantly limited. Based on these considerations, it is worthwhile the investigation of novel tools for improving human natural abilities. One of possible ways to achieve this augmented-reality goal is to consider a tele-operated system composed by teams of cooperating humans and robots (namely, mobile agents).

According to this vision we develop a localization framework able to localize both rescuers and robots working in an emergency area [53]. We proposed an approach based on both proprioceptive sensors and data retrieved from external infrastructure, adopting the hybridization schema used in robotic localization. In this way, the mobile agent can be always localized by dead reckoning algorithm and update the estimate when the external infrastructure is available. Although the approach has been developed considering the requirements of the firefighters, the proposed localization procedure can be easily used also in mass-market applications [39, 37].

1.3 Contributions

The proposed thesis is based on three studies. First, we provide a theoretical framework for the design of a localization system for mobile agents in emergency scenario. Considering the dead reckoning for pedestrian, we develop algorithms for activity recognition exploiting signal processing techniques for improving the performances of human tracking. Finally, we present specific strategies for exploiting passive or active external infrastructure. All the developed algorithms satisfy the requirements of the firefighters and are able to produce an update estimate in an on-line fashion. To this end, all the procedures are designed with high computational efficiency.

1.3.1 Design of the localization systems

The approach adopted to develop the localization system for hybrid teams is based on the prediction-correction schema used in robotics [55, 38]. Specifically, we suppose that the mobile agent (human or robot) in the emergency area is equipped with proprioceptive sensors (i.e., an Inertial Measurement Unit - IMU -, encoder, etc.). These sensors are used to obtain a rough *a–priori* estimate on the position of the mobile agent with respect to the environment. The estimate is further refined *a–posteriori* using the external infrastructure, which provides information about the surroundings. The correction step depends on the type of information that the nodes of the external infrastructure are able to provide: in this work, we consider both range capable and range free nodes.

According to the system requirements and the IoT perspective, no a-priori knowledge is supposed available to the mobile agent (i.e., the map of the area) that acquires information during mission by smart environment. The proposed algorithms fulfill the requirements described in the previous Section for being used in an emergency scenario. Furthermore, their low computational complexity allows their implementation in low energy consumption, low computational power nodes. To the best of our knowledge previous works address specific problems without developing a theoretical framework.

1.3.2 Rescuer filter

Concerning the rescuer localization, we design several algorithms based on signals provided by a waist-mounted IMU. The most interesting contributions regard the algorithms for activity recognition and the pre-processing of the IMU signal for both position and heading estimate.

The activity problem is addressed in two different ways. The first algorithm is based on a decision tree: simple features of the signal in the time domain are analyzed to exclude subsets of activities [20, 33]. The second algorithm is based on a Linear Discrimant Analysis [56]. A learning step and a simple classifier characterize this approach. The first phase requires high computational load and analyzes features in both time and frequency domain. It runs only once off-line to determine the parameters of the classifier, which do not depend on the particular user. The overall computational load is limited resulting in a possible on line application of the proposed system. The current state of the art algorithms to compute the heading of a pedestrian are improved using quaternions-based filtering and preventing the use of outliers in the nonlinear filter [18]. Concerning the hybridization in passive environments, it is supposed that the network is able to provide information on the environment [10, 90], however in passive environment nodes or sensors (items) cannot directly communicate each other. A mobile user can detect items that are in its proximity, since the nodes of the infrastructure are considered range free. The presence information is, therefore, properly mapped in a probabilistic framework.

1.3.3 Localization for Wireless Sensor Networks

In active environment, the nodes of the external infrastructure compose a wireless sensor networks and the problem of concurrently estimate the position of the mobile agents and the nodes is addressed. The novelty of the proposed approaches relies on the exploitation of information about the presence or absence of a communication link between nodes [54]. Using the lack of communication, indeed, the localizability of the network can be extended and the mobile agents are able to properly augment the maximum localizable component. Two different approaches are proposed to estimate the maximum localizable component in a network: they mainly differ for the modality

of interaction between the mobile agent and smart environment (i.e., loose or tight cooperation).

1.4 Outline of the Thesis

The remainder of the thesis is organized in the following way.

Chapter 2 presents the related work on the topic covered during the doctorate. Specifically, the pedestrian dead reckoning, the hybridization techniques and the localization approaches for wireless sensors networks are introduced .

In Chapter 3, the design of the proposed localization systems is sketched. The theoretical background is recalled and the filter that runs on the mobile agent is depicted.

In Chapter 4, the rescuer filter is explained: the algorithms for activity recognition, heading and step computation are deeply investigated, furthermore the refinement of the position estimate based on range free and range capable sensors is explained.

In Chapter 5, the simultaneous localization of the mobile agent and the nodes of the smart environment is addressed. The localizability of a network using a mobile agent is addressed.

In Chapter 6, the experimental results performed to evaluate the approaches are reported and discussed.

Some conclusive remarks and suggestions for future development can be found in Chapter 7.

The interested reader can find some introductory material on the analytical tools and the devices used in this thesis in the Appendices.

Chapter 2

State of Art

2.1 Localization for Rescue Personnel

Navigation by sight is impossible during emergency since darkness, smoke, or dust limit visibility to less than one meter. Out-of-reach people or objects can be passed unnoticed. Moreover, the environment can change as ceilings, floors, or shelves collapse, furniture moves, and people searching for an exit open or close different doors. The fire noise can hide PASS alarms, interfere with radio conversations, and make cries for help difficult to locate.

In the past decade, researchers put much effort into this challenging problem and developed a wide variety of ideas to address it. A complete taxonomy of the technologies used for localization systems is hard to drawn. Here we investigate some localization principles, propose some evaluation criteria and discuss the localization systems proposed in literature.

The approaches investigated during the years can be divided in three sets:

- Infrastructure Base Localization – Exteroceptive sensors are used, the user carried the sensors receiver and brings the information from the sensors deployed in the environment;

- Dead-Reckoning – Proprioceptive sensors are considered and the information provided by the human movements are used to compute the position of the rescuer;

- Hybrid System – Proprioceptive and exteroceptive sensors provide information that are combined to obtain the position of the rescuer.

2.1.1 Infrastructure Based Localization

The information provided by sensors deployed in the environment are used to compute the position of the rescuer. The method that can be used to compute the location can be classified as: Lateration and angulation systems, Proximity systems, and Radio fingerprint systems [52, 95, 71, 48].

- Lateration and angulation systems – Distances or bearing between a set of know locations, called base stations, and a mobile unit are computed. A drawback of this system is due to the indoor scenario itself, in fact wall and obstructions may not propagate the radio signals may not propagate through the environment.
- Proximity systems – These kind of systems provide only coarse location, they have as advantage little calibration, but as disadvantage a high density of readers to gain reliable, ubiquitous coverage are required.
- Radio fingerprint systems – Signal properties are compared to a database of properties previously collected at a variety of locations. The closest match is returned as the estimated position.

2.1.2 Dead-Reckoning

These systems use proprioceptive sensors to estimate relative and/or absolute location and consists of determining the movements of a target to build its trajectory from a given starting point. Pedestrian dead reckoning (PDR) involves estimating step lengths and directions using a pedometer and a compass. More complex approaches exploit foot-mounted inertial sensors (accelerometers and gyroscopes) to track every the motion of the legs [36]. Dead reckoning only requires sensors to be carried or worn by the person being tracked, making it particularly attractive for localization in unstructured environments. Inertial sensors can also provide the incident commander with information about the posture of the rescuer, such as whether he or she is standing, crouching, or lying down [106].

PDR are affected by incremental error in position estimate, since the small errors in each measurement accumulate and, consequently, the positioning error increases without limit. Eric Foxlin reported very accurate tracking [41]; Lauro Ojeda and Johann Borenstein also achieved good accuracy in challenging environments [78]. However, other researchers reported much larger errors. [40, 102]. The accuracy of the sensors, the way they are attached to rescuer bodies and the walking pattern affect results and make predicting uncertainty in a PDR position estimate very difficult. Current PDR systems are unusable in scenarios in which the wearer crawls, climbs, or walks with an irregular pattern (while transporting a victim, for instance). The challenge for researchers is therefore to couple dead reckoning with other localization techniques, without substantially increasing the deployment effort.

2.1.3 Hybrid System

Sometimes the position provided by a PDR system needs to be corrected using an external (absolute) positioning system. These kind of strategies are called *"Hybrid System"*, one of the most popular is composed by PDR and WiFi fingerprinting. With fingerprinting, the main difficulties are associated with comprehensively surveying the building to form a radio map and in keeping that map up-to-date. It is also possible to invert this approach and to use the PDR system to support WiFi fingerprinting as proposed by [84]. In [84] is proposed a positioning system, the Zee, based on fingerprinting. The Zee system proposes positioning primarily through fingerprinting, but with a radio map that is crowd-sourced from the subset of users with Step-and-Heading Systems (SHSs) capable smartphones, the system uses a SHS based on Particle Filter combined with a floor map.

In [32] and hybrid system that did not depend on a centrally-established database or a pre-supplied building map of signals is presented a hybrid system. It took as input an SHS based on accelerometer and compass, as well as radio signal strength measurements from the WiFi and GSM (Global System for Mobile Communications Groupe spécial mobile) cellular radios. The drift (the output signal slowly changes independent of the measured property) is corrected dividing the area into a grid and applying SLAM (Simultaneous Localization And Mapping) technique. The approach idea is to monitor

the radio fingerprints to detect any loops in the path taken, moreover the loops can be use as constraint to recompute the full path taken.

2.2 Indoor Localization

An Indoor Localization System (ILS) is a system to locate objects or people inside a building according to their coordinates or their displacement. According to the movement of the user's body it is possible to estimate the position. The movement can be detected using Inertial Sensors that can be placed in different spots on the user as: foot, hand or torso. Anyway, independently from the Inertial Platform displacement, to estimate the rescuer position three step must be done:

- Activity;
- Attitude Estimation;
- Step Computation.

2.2.1 Activity

A variety of pedestrian dead reckoning techniques using proprioceptive sensors have been proposed, as in [59] and in [79], where the tracking of the pedestrian is limited to a 2-dimensional space. In [79], a tracking algorithm consisting of step detection, step length estimation and heading estimation is introduced. The estimates of the position computed by two or more sets of sensors carried by the same walking pedestrian are fused in [59] to reduce the average tracking error. However, in several real world applications, walking paths may involve more than one floor within a building.

The integration of a pedestrian dead reckoning with a barometer to extend the tracking algorithm to the vertical dimension is presented in [36]. Common barometers, however, may not provide a desirable resolution, as they are only sensitive to vertical displacements in the order of meters. In [74], the authors introduce a 3-dimensional pedestrian dead reckoning technique based on a handheld IMU, which might not apply to application scenarios where free hands are required, e.g. rescue applications.

Recent research faced the challenge of analyzing the class of pedestrian activity based on the data provided by the proprioceptive sensors [20]. Features of the accelerometer signal are studied both in the frequency and in the time domain. In [49], the adoption of delta features, low and high frequency signal variations, and energy variations in both frequency bands is investigated. The extraction of features of acceleration data in anterior-posterior (AP), medio-lateral (ML) and vertical (V) direction was proposed in [75] where the acceleration signal in each direction was decomposed to six detailed signals at different wavelet scales by using the wavelet packet transform. In [65], empirical mode decomposition and an amplitude/frequency modulation model is proposed to extract relevant features. The authors of [49, 75, 65] did not investigate the integration of the proposed models in an operating 3-dimensional pedestrian dead reckoning mechanism though. A decision tree is introduced in [33] to identify the different walking patterns. However, the proposed approach requires an expert to perform a manual calibration of parameters that vary depending on the body features of the user.

2.2.2 Attitude Estimation

The Attitude Filter estimates the attitude of the IMU wore by the rescuer with respect to a navigation reference frame. It is obtained merging data collected from gyroscopes, accelerometers and magnetometers. The orientation is compute using quaternions: these mathematical entities, indeed, need less computational effort in recursive updating and avoid the singularity issues that affect angular descriptors, like Euler angles.

Several efforts are spent in the threshold-based algorithms where the threshold can be computed according to different information provided by the sensors; in [94, 66, 13] accelerometer magnitude is considered, in [104, 14, 78] angular velocity thresholds are used, and in [41] a combination has been trailed. In [58] it is shown how also a threshold applied on magnetometer signals can give usable stance detection under some circumstances, moreover applying the threshold to the moving variance of the signal has been preferred to applying it to the instantaneous magnitude [44, 58].

Even if the results obtained shown errors of 0.1% and 0.2% on thresholding accelerometer variance and angular velocity (see [58]), threshold-based stance detection requires a sensor on the foot. In rescue scenario it is not always possible to use a foot

mounted sensors to this end there position are investigated as: waist mounted.

Unfortunately in waist mounted context it is not always possible to be axis-aligned but the sensors rotate respect to the world frame (with axes in the horizontal and vertical planes) during the gait cycle, that implies the necessity to track the sensor rotation using the data (angular velocity) provided by the gyroscope. That means that for each position update and the measurement error (drift) growth in time.

2.2.3 Step Computation

In this work the Step Computation is done using Dead-Reckoning technique. According to the distinction proposed by [51] the Dead-Reckoning systems can be divided in two set: *Inertial Navigation Systems* (INSs) and in *Step-and-Heading Systems (SHSs)*. The INSs is used to track the position by estimating the full 3D trajectory of the sensor at any given moment and SHS estimates the position by accruing distance, heading vectors representing either steps or strides. More over it is possible to mix these strategies providing an SHS-INS system.

The first goal of SHS and also INS is to identify the set of the data that represents a step or a stride event. Even if the algorithm must permit for accurate step counting, sometimes an accurate step segmentation is required, according to that it is possible to identify two main algorithm types:

- Stance detection – periods of data are identified when a given foot is planted on the floor. They are appropriate for step counting but give poor segmentation output more over they need a foot-mounted sensor;
- Step cycle detection – the cycles in the sensor data caused by the repetitive motion of walking are detected. They need a repetitive data patterns or event and are well suited to step segmentation.

Usually the stance detection algorithms are threshold-based because the key idea is that during the stance phase the sensor is static and report a lack of activity that thresholding can easily identify.

To identify specific events for data segmentation several strategies are available, and they relax the foot-mounted sensor constrain. These strategies are summarized as:

- **Peak detection** – The heel strike is associated with changes to the vertical acceleration, unfortunately the foot impact may generate multiple local peaks, due to the higher forces resulting in sensor bounce [31, 110];
- **Zero crossings** – The acceleration value for zero crossings [46] is monitored, this strategy is used for pedometers or activity monitors due to its simplicity;
- **Auto-correlation or template matching** – The cycle can be extracted by seeking maxima in the mean-adjusted autocorrelation of a sequence of sensor data [110]; the drawback of this strategy is that the peaks corresponding to a step or a stride will depend on where the sensor is attached;
- **Spectral analysis** - The frequency spectrum of the cyclic data is computed and strong peaks at typical stepping frequencies are identify. Windowed subsets of the data are converted to the frequency domain and the dominant frequency taken as the walking frequency [70].

Usually these techniques are applied to accelerometer signal. It is worth to notice that most of the implementations claim to use only the vertical acceleration, but do not compensate for changes in the global pose of the sensor during a step. Instead they assume that one of the accelerometer axes remains vertical throughout. This assumption is most valid for waist mounted inertial sensors [67, 26], even if [85] shown acceptable results also with foot mounted sensors.

The techniques presented above are inertial-based sensors but a lot of non-inertial sensors can be used to facilitate step or stance detection. In [80] pressure sensors embedded in the sole of the shoe are used; in [98] impact switches are used in a similar manner; [87] used ultrasonic ranging between the feet; and [99] used electromyography (EMG) sensors attached to the calf. However, these solutions require more invasive attachment to the user and rarely offer any benefit over the use of inertial sensors (which are still needed for subsequent PDR analysis anyway).

2.2.3.1 Inertial Navigation Systems

An INS uses triaxial accelerometers and gyroscopes to track orientation and position changes [105], in this configuration the sensors are embedded into a rigid package and

attached to the body. Tracking position then involves subtracting the gravitational signal from the vertical accelerometer signal and performing double integration on the remaining 3D acceleration.

Usually the sensors used in INS are Micro Electro-Mechanical Systems (MEMS) technology. MEMS sensors are small and highly portable but they are also subject to more significant error sources [105]. Strapdown inertial navigation algorithms have been well studied and the standard approach to limit drift uses an Extended Kalman Filter (EKF) in the complementary or indirect form, whereby the filter tracks the errors in the system state rather than the system state directly.

Zero Velocity Updates (ZUPTs) can be used to reduce the drift, this method assumes that the sensor is stationary and can be applied during the stance phase. ZUPTs were first used in a PDR context in the NavShoe project by Foxlin, who reported good results in 2005 [41].

The application of ZUPTs means that open loop integrations only occur during the swing phase of the foot to which the sensor is attached. For such short durations, drift accrual is limited and longer tracking durations are thus feasible. For a reliable output, however, ZUPTs must only be applied when the foot (and hence sensor) is completely static. Issues can arise when the sensor is attached any higher than the ball of the foot. The peeling motion associated with the transition from stance to swing means the heel rises soon after the foot-down event and hence a sensor in the mid-foot will start experiencing an acceleration as the foot levers up. These small accelerations occur before the strict end of the stance phase and it is necessary to account for these errors by applying a non-zero covariance alongside the ZUPT pseudo-measurement.

Nowadays in commercial inertial sensors usually are integrated triaxial magnetometers with the accelerometers and gyroscopes necessary. The information provided by the magnetometers can give a direct estimate of the user's absolute heading. This information can be very useful to correct the drift that occurs in INS discussed above. Unfortunately, the Earth's magnetic field is relatively weak at its surface and modern buildings, filled soft and hard iron, can modify the signal, leading to local disturbances.

In [1] a detailed set of experiment are discussed and the effects on magnetometers change according to the different building materials with heading errors of up to 100◦

observed. A common correction algorithm is to reject any readings where the magnitude lies outside some tolerable range of the expected value.

In [2] the authors noted that this can be overzealous since vertically polarized magnetic fields will affect the magnitude but not the horizontal heading. To use this knowledge, however, requires that the pose of the sensor in the world frame of reference be known except for horizontal heading so there are practicality issues.

In [64] the complementary characteristics of gyroscope (long-term orientation errors) and magnetometer (short-term orientation errors) are analyzed by comparing the angular velocities measured by the gyroscope with those computed from successive magnetometer readings. The results obtained shown an improvement only if there is not strong magnetic disturbances. Magnetic disturbances can also be modelled using the EKF as in [41, 46]. By assuming a high spatial frequency of disturbances, the EKF is able to estimate the covariance in the heading measurement.

2.2.3.2 Step and Heading Systems

The key idea of the Step and Heading Systems (SHSs) is to provide as output a series of *{step length, step attitude}* or more often *{step length, step attitude change}* polar vectors. This output can be summed in a vector space to obtain the track position. Even if the vectors could be derived simply by degrading the INS output, they can also be estimated without using the triple integration and the drift related to the integration discussed previously. The SHSs are composed by two step:

- Length Estimation;
- Attitude Estimation.

One of the simplest approach to step length estimate it is to assume that the distance covered in a stet is constant in fact, usually, pedestrians have a natural walking pace with a constant step length. This walking pattern can change when rushing, ambling or walking with others. In [100] it is reported that a step length can vary by as much as 40% between pedestrians walking at the same speed, and up to 50% across the range of walking speeds of an individual. Moreover, in [100] a dynamic step length estimation using maximum vertical displacement of the hip is described. Here is show that the

angle between the highest and lowest point of the hip during a single stride is related to the leg length of he user.

An other way is to base step lengths on the observed step frequencies.

The biomedical researches provide the correlation between step frequency and the walking speed. Although the precise relationship is non-trivial (see [8]), it is common to fit a linear relationship, which is sufficient to achieve an error of 5.6% [108]. Direct measures of each step have also been used. For example, Saarinen used ultrasonic sensors mounted on the front and back of each shoe [87] and similar range-finding techniques have been used elsewhere. Such techniques unquestionably produce higher quality displacement estimates. However, the increase in accuracy is often negligible since the larger heading drift is typically the limiting factor. Finally, some systems estimate step lengths iteratively by evaluating the paths produced using different lengths and using building maps to choose an optimal length [84].

There are not significant differences between SHS heading estimation and the INS heading estimation since there are so few sensors available as inputs. To obtain the heading change a single integration of gyroscope signals can be used. Compering to the INS, SHSs can avoid using subsequent integration for the step length, that means that the drift does grow at least linearly instead than cubically as in INS. Moreover, some systems use only a single gyroscope mounted parallel to the torso and it is assumed that it remains vertical during walking.

2.2.3.3 Models

Several models and system for estimate the step length are presented in literature. The differences between the models are related to sensor position, type and number used. In its simplest form the model of human walking can be represented as an inverted pendulum. Starting from this mechanical model it is possible to obtain the horizontal shift forward, by means of a formula called M_1:

$$S_{M_1} = K2\sqrt{2lh - h^2} \tag{2.1}$$

where h is vertical displacement calculated by double integration of the vertical acceleration of the center of mass (COM), l is the length of the leg that represents the

vertical displacement of the COM, then K is a calibration parameter that have to be computed for each user. This model is composed by two steps:

- Calibration Procedure: the calibration procedure is carried out by K in the following way: K is the difference between the real distance and the estimated one according to a trial path.
- Computation of the COM vertical displacement: the vertical displacement h is calculated by double integration of the vertical acceleration of the COM. To avoid the error due to the drift of integration is necessary to reset to zero the value of the integral at each step. It is common choice set as zero point of the integral the instant in which the sole of the foot is fully adherent to the ground, in this precise moment the vertical acceleration of the COM appears to be nothing.

A second estimator called M_2 and characterized by the following equation:

$$S_{M_2} = \int_{t_0}^{t_1} (a_v + c_1)\, dt = 0 \tag{2.2}$$

where a_v is the vertical acceleration, t_0 and t_1 represent two consecutive instants in which the sole is fully adherent to the ground, and c_1 represents the offset. It assumes that when the sole of the foot is fully adherent to the ground, the speed of the COM is zero and the vertical coordinates of the latter are the same as the beginning of the step.

A third estimator called the M_3 model assumes that the COM is related to two pendulums. An inverted pendulum with a length equal to the length of the leg of the user during the lifting phase and a second pendulum (with unknown length) during the phase in which all the feet are on the ground:

$$S_{M_3} = S_1 + S_2 = 2\sqrt{2lh_1 - h^2{}_1} + S_2 \tag{2.3}$$

The displacement S_{M_3} is calculated as the sum of the lifting phase (swing) S_1, and the displacement due to the double support phase S_2. The lifting phase S_1 may be calculated by the formula used by the model S_{M_1}. The value of the vertical displacement of the COM, h_1 is calculated as half of the double integral with the instant extreme lifting of the tip (toe off) until the time in which the heel touches the ground again (heel strike).

The last estimator S_{M_4} for the long wheelbase (stride) is based on a empiric report between the maximum vertical acceleration and minimum vertical acceleration during a gait cycle:

$$S_{M_4} = K\sqrt{a^{v,max} - a^{v,min}} \tag{2.4}$$

where, $a^{v,max}$ and $a^{v,min}$, represent respectively the maximum and the minimum value of the first harmonic of the vertical acceleration of the COM.

The models presented above overestimate the value of the step. Even if these estimators have almost equal performances, the estimator M_4 is used in this work because it permits to reduce the variance between each step.

2.3 Wireless Sensors Network

The goal of localization in a WSN is to determine the position of the nodes in the network. In this work, a WSN Σ composed by N fixed sensors σ_j is considered. The network is randomly deployed in a planar environment, so that the position of the sensor σ_j can be written as $p_j = [x_j, y_j]^T \in \mathbb{R}^2$ with respect to a navigation reference frame. Each sensor σ_j is equipped with an isotropic communication antenna characterized by a maximum communication range ρ. Given two nodes σ_j, σ_i, the antenna is also used as a rangefinder for computing the Euclidean distance $d_{j,i} = ||p_j - p_i||$.

From a theoretical perspective, the fixed WSN can be described using the disk model: a node is capable of sensing only points that lie within its communication range ρ. According to this approach, a WSN can be represented by a unit disk graph $G = \{V, E\}$, where the set V denotes the nodes $v_1, \ldots, v_N$ (i.e., the sensors $\sigma_1, \ldots, \sigma_N$) and E is the set of edges (v_j, v_i) (i.e., the presence of a communication link between sensors σ_j and σ_i). The localization problem can be set as finding a coordinate assignment P for the non-anchor nodes, i.e., to choose the location p_j for each sensor $\sigma_j \in \Sigma/\Sigma_s$ in a way such that d_{ij} holds for all sensor pairs σ_j, σ_i such that $(v_j, v_i) \in E$. In Graph Theory, this problem is the same as finding a framework (G, P) such that the coordinate assignment P does not violate the distance constraints, where a framework is a graph $G = \{V, E\}$ together with a coordinate assignment $P : V \to \mathbb{R}^2$ for the vertices of the

graph. A unique solution for the theoretical problem is proved to exist if the framework is globally rigid, i.e., the position of the nodes cannot be continuously deformed nor flipped without breaking the distance constraints [68][21], however the computational load of the global rigidity test is high.

Several strategies are investigated and implemented to WSN localization that are related to the quantitative (position obtained according to a reference) or qualitative (network topology) nature of the solution [69, 73]. According to the [15] classification, here two kinds of localization are considered, *target/source localization* and *node self-localization*. These strategies can be subset as:

- **Target Localization**
  - Single-Target Localization;
  - Multiple-Target Localization.
- **Node Self-Localization**
  - Range-Based Localization;
  - Range-Free Localization.

In this work the focus is on a quantitative localization, range-based to a minimum number of anchors node both target and node.

2.3.1 Target/Source Localization

The source localization methods have a wide range of possible applications. The outdoor application includes vehicle or aircraft localization. In an indoor environment, this method could track the human speakers. In underwater environment, it can be used to locate the large sea animals and ships.

2.3.1.1 Single-Target/Source Localization in Wireless Sensor Network

.

Nowadays exists several way to the source location: energy-based, angle of arrival (AOA) [62], time difference of arrival (TDOA) [101, 81, 82, 9]. The most attractive

approach is the energy-based method, it requires low hardware configuration and it is an inexpensive approach.

2.3.1.2 Model-Independent Methods

In [83] a kernel averaging approach that does not need information about energy decay model is proposed. In [4], a novel model-independent localization method was proposed. The idea is that a the nodes with higher received signal strength measurement were closer to the source so a distributed sorting algorithm can be used. If the sensor nodes know their rank, the required distance estimates are obtained as the expected value of the respective probability density functions. Finally, the Projection Onto Convex Sets (POCS) method was used to estimate the location of the source.

2.3.2 Node Self-Localization

The process of estimating the unknown node position within the network is referred as node self-localization.

2.3.2.1 Range-Based Localization

The common approaches to estimate the indoor location are:

- Time of arrival (TOA);
- Time difference of arrival (TDOA);
- Angle of arrival (AOA);
- Received signal strength (RSS).

TOA method measures travel times of signals between nodes while TDOA locates by measuring the signals' arrival time difference between anchor nodes and unknown node. Even if it is able to achieve high ranging accuracy it requires extra hardware and consumes more energy. The RSS approach has established the mathematical model on the basis of path loss attenuation with distance [72, 16]. It is possible to compute the distance between the anchor node and unknown node using one of three measurement

methods described above. Defining the anchor node position as $\langle(x_1, y_1), ..., (x_N, y_N), \rangle$, unknown node position as $X = [x, y]^T$ and the estimated distance between $i - th$ anchor node and unknown node as $\tilde{d}$ it is possible to obtain the coordinate matrix of the unknown nodes as follows:

$$X = (A^T A)^{-1} A^T B, \tag{2.5}$$

$$A = 2 \begin{bmatrix} (x_1 - x_2), & (y_1 - y_2) \\ (x_1 - x_3), & (y_1 - y_3) \\ \vdots & \vdots \\ (x_1 - x_{N-1}), & (y_1 - y_{N-1}) \end{bmatrix} \tag{2.6}$$

$$B = \begin{bmatrix} \tilde{d_2}^2 - \tilde{d_1}^2 - ({x_2}^2 + {y_2}^2) + ({x_1}^2 + {y_1}^2) \\ \tilde{d_3}^2 - \tilde{d_1}^2 - ({x_3}^2 + {y_3}^2) + ({x_1}^2 + {y_1}^2) \\ \vdots \\ \tilde{d_{N-1}}^2 - \tilde{d_1}^2 - ({x_{N-1}}^2 + {y_{N-1}}^2) + ({x_1}^2 + {y_1}^2) \end{bmatrix} \tag{2.7}$$

The angles between unknown node and a number of anchor nodes are used in the AOA method to estimate the location, unfortunately this method requires the antenna array which is an expensive solution for low-cost sensor node.

2.3.3 Localization in Some Special Scenarios

Ideal and real applications of WNS are different, the scale factor in applications can be small or large, and the environments can be different, that means that the usual localization methods are not suitable for the special scenarios and some problems need to be solved. The problem investigated can be summarized:

- Energy consumption and localization accuracy problem;
- Cooperative node localization;
- Heterogeneous sensor network localization.

2.3.4 Energy Consumption and Localization Accuracy Problem

Due to the limited power of sensor node and hostile deployment environment, the node selection in WSN is different from the traditional node selection in traditional wireless network. If all the sensor nodes are used at the same time to executive localization task without selection, although the energy consumption of nodes selection are saved, but at this time the repeatability of the received information would be quite larger. If the nodes are randomly selected, the algorithm is simple and the extra overhead can be ignored, but localization accuracy is low in this case. This method cannot satisfy the user's requirements for the accurate localization, the unbalance of energy consumption will appear, and some nodes may fail due to the depletion of energy. This may affect the network connectivity and may result in losing the sensed data. These characteristics of WSN determine selection method which is different from traditional network. Therefore, it is necessary to investigate nodes' selection mechanism in WSN.

The primary algorithm makes decision with global information [60]; this method minimized the expected filtered mean-squared position error for a given number of active nodes by using a global knowledge of all node locations. This algorithm needs the positions of nodes and broadcasts them to all nodes and a lot of data communication; therefore, it only can be applied to small networks. Based on the former algorithm, a local selection strategy is investigated [61]. This method determines whether or not that node should be active by only incorporating geometrical knowledge of itself and the active set of nodes from the previous information. Based on this approach, the researchers have also investigated other strategies, such as the least square method, Bayes probability method [6]. Furthermore, in order to narrow the scope and scale of selected nodes, researchers proposed a method which combines the track and the current state of the robot. In [114] the authors propose a multi-node cooperation dynamic tree algorithm. This method ensured that spanning tree has low energy consumption and high information content by increasing and decreasing the number of the nodes dynamically. In [109] online prediction based on particle filter is proposed and the probability distribution of the target state under the Bayes framework is estimated. This method realized the optimal selection of the node sequence and introduced a shortest path algorithm to reduce the information transmission. In [50] a method that uses

moving speed of the mobile robot to improve the localization accuracy and consistency is described. The signal shielding and multipath interference make the channel parameters become too complex to definite error factor. [7] proposed two selection principles to reduce the number of active nodes, and the nodes with accurate measured value (RSSI value larger than specified threshold) are selected. This method is effective to balance the accuracy and energy consumption and is suitable for the WSN which is hardware resource constrained. [115] used the Cramer-Rao equation to select the next node to participate in positioning. Because of the complexity of the observed model and the non-Gaussian noise, it is hard to get the optimal solution of the problem.

As previously said a typical sensor network consists of a large number of small sensors that are deployed randomly. However, a sensor node has limited resources because of battery power and small memory. In typical sensor network applications, nodes are deployed in an unattended environment such as disaster management, habitat monitoring, industrial process control, and object tracking. Enormous event data will be generated for a long sensing time in WSN. Hence, by the methods of nodes resource management, effective usage of the vast amount of data is crucial. In the meanwhile, the scalability of both energy and spatial dimensions in distributed sensor network is a key issue. Sensor networks must track various phenomena at the same time and work within limited communication bandwidth, energy, and processing speed.

Energy consumption is one of the most important issues in recent years. In [86] authors proposed a localization algorithm based on particle filtering for sensor networks. Assisted by multiple-transmit-power information, it outperforms the existing algorithms that do not utilize multiple-power information. In [111] a specified positional error tolerance is proposed, the sensor-enhanced and energy-efficient adaptive localization system in an application. This localization system dynamically sets sleep time for the nodes and adapting the sampling rate of target's mobility level. However, the process of error estimation dynamically relies on several factors in the specific environment. [47] proposed a scheduling algorithm that selects a subset of active anchor nodes to be used in localization. It served to reduce the message overhead, increased network lifetime, and improved localization accuracy in dense mobile networks. However, maximizing the nodes' sleep time is much more energy efficient if the nodes never wake up until the

reception of wake-up messages. The above algorithms have the same feature that the duty cycle of the sensor nodes is fixed in advance. In [116], the authors proposed an innovative probabilistic wake-up protocol for energy-efficient event detection in WSNs. The main idea of it is to reduce the duty cycle of every sensor via probabilistic wake-up through the dense deployment of sensor networks.

The problem of unique network localization and a mathematical topic known as rigidity theory have a strong connection, [45] proposed a localization method for sparse networks by sweeping techniques. This method is saving all possible positions in each position step and pruning incompatible ones. One drawback of sweeping method is that the possible positions could increase exponentially as long as the number of nodes increased. Other types of localization methods are also available, such as using multi-dimensional scaling [57, 91] or mobile anchors [93, 30]. However, all the previous works tried to localize more sensor nodes in a network without guaranteeing all of them. In [63] it is introduced a localization method to localize all nodes by the minimal number of anchor nodes. However, they assume that the sensing range of each sensor can be enlarged to guarantee certain triangulation, so that three anchor nodes are enough to localize all sensors.

2.3.5 Cooperative Node Localization

There may be not enough information in the concentrated network or the node may contain the harmful information in sparse network. There are two branches in this area:

- access the accuracy and reliability of the neighborhood nodes;
- improve precision with the cooperation of the active and passive nodes.

Some nodes may bring unreliable or even harmful information [107], so it is essential to review the received information. In [96] it is employed the nearest link as reference to review the information. When there are massive link in dense network and positioning mainly depends on the geometry of the neighbor node topology information, the nearest neighbors may not correspond to the best link. Aiming at this issue, [25] proposed an adaptive method to eliminate the inefficient links, but this method has to work with neighbor node information, and the method cannot effectively reduce the

number of packet effectively. Therefore, [24] put forward a kind of distributed criterion; this method employed Cramer-Rao limit as identifiable parameters to identify the links. This method could avoid the invalid neighbor node links and unreliable transmission; thus, it can effectively reduce the computation time and the number of packets. The accuracy of master-slave node cooperative localization is mainly depended on the measurements accuracy and the number of Primary Reference Nodes (PRN). Unfortunately in the actual application, it is difficult to increase the number of primary reference nodes because of the factors of energy and the complexity. In [107, 42] it is put forward a new method: the nodes which received the information of the target and the PRN are termed as Secondary Reference Nodes (SRN), the SRNs participated in the localization in a passive way. This cooperative method reduced the required number of PRN with relatively higher localization accuracy. In [43] the maximum likelihood estimation method is used to obtain the target position. The authors formulated the localization problem into finding the intersection of the vertex set by using geometry description. This method avoids getting into the local optimum.

2.3.6 Heterogeneous Sensor Network Localization Algorithm

Most of the localization methods for the wireless sensor networks are only to consider the homogeneous network. The different kinds of the nodes such as the different maximum communication radius and the different nodes own the different localization mechanisms are not considered in homogeneous, so the localization methods for the homogeneous network cannot be directly applied in the heterogeneous wireless sensor networks. A new boundary nodes localization method by using a small number of anchor nodes is proposed in [28] . First the boundary nodes are elected and their positions are determined and then the location information of boundary nodes is sent to other nodes through a small hop communication range. Finally, other nodes estimate their locations by the hop count and hop range. The scheme uses fewer anchor nodes, but has much smaller localization error and standard deviation.

In [27], authors proposed a two-step localization method for two-tiered hierarchical heterogeneous sensor networks. The network is composed by of three types of nodes:

- Anchor nodes with known locations;

- Nodes equipped with both Ultrawide Band (UWB) and RF radios;
- Large number of normal sensor nodes.

The localization method works in two steps: first the high-accurate ranging capability of UWB nodes is used to estimate their location from a few anchor nodes, then, sensor nodes estimate their locations by using UWB nodes as anchor nodes. Sometimes the distance between some nodes can be measured directly, while others cannot be. Selecting a different positioning algorithms accord to the mutual distance between nodes can be measured or not. [17] proposed a Hybrid Unified Kalman Tracking (HUKT) technique. The accuracy of tracking is based on both time of arrival and time difference of arrival measurements. This method is proposed to adaptively adjust the weighting value between the TOA and TDOA measurements. The scheme can both provide higher localization accuracy for mobile network and adapt to environments with insufficient signal sources. According to the different communication radius of the nodes, some super nodes can be deployed at some areas with plenty communication demands to transmit the information. [92] considered the nodes with more power and longer communication range as the heterogeneous nodes and propose a heuristic relay positioning algorithm for heterogeneous wireless sensor networks, to achieve the sharing of resources in heterogeneous wireless sensor network by using the relay nodes.

2.4 Rescuer Localization

Researchers have built location systems exploiting a large variety of technologies.

2.4.1 Dead Reckoning

Dead reckoning consists of determining the movements of a target to build its trajectory from a given starting point. Pedestrian dead reckoning (PDR) involves estimating step lengths and directions using a pedometer and a compass. More complex approaches exploit foot-mounted inertial sensors (accelerometers and gyroscopes) to track every the motion of the legs [36]. Dead reckoning only requires sensors to be carried or worn by the person being tracked, making it particularly attractive for localization in

unstructured environments. Inertial sensors can also provide the incident commander with information about the posture of the rescuer, such as whether he or she is standing, crouching, or lying down [106].

PDR are affected by incremental error in position estimate, since the small errors in each measurement accumulate and, consequently, the positioning error increases without limit. Eric Foxlin reported very accurate tracking [41]; Lauro Ojeda and Johann Borenstein also achieved good accuracy in challenging environments [78]. However, other researchers reported much larger errors. [40, 102]. The accuracy of the sensors, the way they are attached to rescuer bodies and the walking pattern affect results and make predicting uncertainty in a PDR position estimate very difficult. Current PDR systems are unusable in scenarios in which the wearer crawls, climbs, or walks with an irregular pattern (while transporting a victim, for instance). The challenge for researchers is therefore to couple dead reckoning with other localization techniques, without substantially increase the deployment effort.

2.4.2 Infrastructure-based localization

Many location systems, including GPS and some commercial indoor location systems, use distance and angle measurements to determine a target coordinates. Typically, this target is a tag able to transmit a radio, ultrasound, or infrared signal detectable by sensors installed at known locations in the building. The sensors measure the distance or direction of the signals and a central computer estimates the position of the tags by means of trilateration or triangulation. In addition, the sensors and central computer have to be connected to each other and the whole system has to be calibrated. The estimated positions are typically detailed and accurate enough for the software to display them on a floor plan.

A different method called finger-printing uses existing Wi-Fi access points to locate a device [112]. Visible access points and their respective signal strengths are different throughout a building, so a wireless client can use them to estimate its location. The level of detail and reliability depend on the number of locations for which finger- prints have been pre-recorded.

A position estimate's detail, reliability, and update rate can be very good, but

infrastructure-based methods fail when the conditions in the building change (temperature rises, furniture moves, floors collapse), power supply is no more available, and/or cables are cut.

2.4.3 Localization with Wireless Sensor Networks

In the field of wireless sensor networks, researchers have developed algorithms to determine the positions of the nodes in the network. The sensors share individual range or connectivity measurements, and each sensor contributes to the calculation of their locations instead of relying on a central device [22]. If a few sensors are anchor nodes with known absolute positions, the estimates improve and the distributed algorithms can compute absolute locations for all the sensors and place them on a floor plan. These methods require no calibration except for the anchor nodes position - all the other nodes can just be dropped or scattered. Computation is distributed and communication is wireless, which means no infrastructure is required. Individual nodes can fail without compromising the whole system. However, some algorithms might provide incorrect position estimates if sensors are moved from their initial position. Such systems typically consist of nodes distributed quite densely in an open space. Although this might not be a realistic assumption for a search-and- rescue mission in a family home, it could apply to an underground parking lot or an airport terminal. Sensor nodes can also serve as radio repeaters to ensure reliable communication when standard radios fail due to thick reinforced walls or underground levels. Because nodes are deployed throughout the building, they can also monitor temperature, the presence of toxic gases, or vibrations signaling that the structure is unstable [103].

2.5 Discussion

Even if a lot of interest and effort are pointing at the PDR systems there are still a lot of issues that need to be investigated. The most important cans be summarized as follows:

- **Consistent evaluation methodologies and metrics** – The outputs from a system can be affected by the characteristics of the test subjects, by the building

size, layout and construction materials, or by the test procedure followed. It is common to report results for a single user over a series of walking tests. Such tests serve as proof-of-concept but there is a growing need for thorough system evaluations over sustained periods with a diverse set of test subjects;

- **Unconstrained sensor sites** – Smartphone-based PDR systems are highly attractive, but they introduce new difficulties by loosening the attachment constraints. Any deployed system must cope with the sensors changing orientation in a pocket, or being taken out to use, etc. In addition, there is often a heading offset between the direction a smartphone is facing and the direction the user is moving. This offset must be accounted for if using absolute heading measurements from a compass;

- **Sensor calibration** – The output of MEMS sensors will vary over time as temperature and other environmental effects affect their bias and scaling factors. For unconstrained smartphones there is typically no opportunity to estimate the sensor biases online, and no opportunity to perform a user-controlled recalibration. This is particularly important for the heading estimation;

- **Battery power requirements** – In the move to mobile devices such as smartphones, it is important to understand the associated energy costs. An SHS system utilizing gyroscope, accelerometers, magnetometers and WiFi signals will put a large draw on the available battery power. Understating which sensors are required and when will be crucial to maintaining sufficient system users;

- **Processing power requirements** – The processing power demanded by the bandpass filters, peak detectors, auto-correlators or particle filters in PDR systems cannot be over-looked. Understanding how to deploy these systems online on the devices themselves will be important;

- **Wearability** – To enable larger scale, more natural testing, it will be necessary to have PDR techniques that do not require foot-mounting, this implies a need for a SHS approach;

Inertial Pedestrian Dead-Reckoning (PDR) systems are presented in this section. It is shown how these kind of systems are done to use the inexpensive and small MEMS sensors that are entering in the smartphone market. It was shown hot the triple integration can be applied only using foot-mounted sensor the represents a strong constrains. To move the sensors from the foot Step and Heading Systems (SHSs) have begun to dominate, where each step is identified and characterized in terms of a length and heading vector.

Using only PDR it is possible to track a constantly-walking user, problems occurs when during periods of non-walking activity, when false positives and false negatives in the step detection, combined with compass and gyroscope errors will contribute positional error that will accrue over time. Anyway for many building user most of the day is spent seated and moving between two location for several minutes. In these conditions it is possible to apply a pure PDR system maintaing a room level tracking details.

Unfortunately in some situation it is not possible to consider only few minutes of activity, in this situation the position estimated using PDR can be corrected by external positioning system. There are several approach, one of the most investigate in this years are based on WiFi but also other sensors information can be used as: RFID (Radio-Frequency IDentification) Technology, Bluetooth Low Energy (BLE) and so on.

It seems likely we will see the emergence of hybrid systems with a variety of positioning modalities to augment PDR. Absolute position fixes will be obtained opportunistically from whatever is installed in the current building – it can be WiFi, Bluetooth, GSM, RFID, or some future communications medium. In this work several of the problem are addressed. Battery and processing power requirements are solved using low computational cost algorithm. Waist mounted platform is considered according to rescuers requirements to avid the problem related to wearability and sensors position. Moreove a sensor calibration procedure is implemented to reduce sensors errors and bias.

Chapter 3

System Overview

3.1 Rescue scenario

In this Section, the scenario usually faced by rescuers, the motivations of using a localization system, and the benefits of exploiting a hybrid human-robot intervention team are introduced.

In recent years, many emergency situations (i.e., natural or un-natural disasters, terrorist activities, etc.) have sharply risen around the world. When adversity strikes, first responders have very limited time to react and rescue.

In search and rescue framework, a fundamental task is the localization in the emergency area of first responders and, eventually, of injured or in distress human subjects. In fact, for effective helping people, activities of the first responders (human or robot) have to be optimized and a real time evacuation support service must be provided. To this end, a reliable localization system is required to be used by the first responders for self localization and for tracking from the operation command center their activities and/or the injured people.

Location awareness is an important aspect in disaster response. The potential benefits of an effective and ubiquitous indoor location for rescuers are twofold: efficiency of the rescue activities could be enhanced by an effective coordination of the resources deployed and safety of the rescuers could be better assured. However, indoor orientation and localization of rescuers within a complex and/or smoke-filled infrastructure is

currently not sustained by technical tools, due to the lack of suitable indoor location technologies.

Localization and tracking are important methodologies and represent one of the industry's top priorities, as underlying by the US National Institute for Occupational Safety and Health (NIOSH). Due to the relevance of this issue, NIOSH explicitly highlights the need for a localization and tracking systems in its reports [76] and [77]:

- Train firefighters on actions to be taken when trapped or disoriented inside a burning structure;
- Consider using exit locators such as high intensity floodlights, flashing strobe lights, hose markings, or safety ropes to guide lost or disoriented firefighters to the exit;
- Ensure that the Incident Commander receives relevant information (i.e., location of stairs, number of occupants in the structure, etc.) from subjects on the field and that information is relayed to team members during size-up;
- Working in large structures (high rise buildings, warehouses, and supermarkets) requires that fire fighters be cognizant of the distance traveled and the time required to reach the point of suppression activity from the point of entry;
- Conduct research into refining existing and developing new technology to track the movement of fire fighters inside structures.

In addition, other studies (such as www.cdc.gov/niosh/fire/pdfs/face200716.pdf) have emphasized the need for reliable communication about inside conditions to the incident commander for inferring building stability. Temperature, smoke, sounds, and vibrations are indicators of a fire's progression and the building's stability.

In our vision, the effectiveness of incident command operations depends heavily on tracking first responders, in-field injured persons triaging and physical environment monitoring. As already stated in the Introduction, during emergency situations, first responders face unknown and complex environments where classical perception schemes fail or are substantially limited. Therefore, it is worthwhile the investigation of novel tools for improving human natural abilities. One of possible ways to achieve this

augmented-reality goal is to consider a remotely operated system composed by hybrid teams of cooperating humans and robots. These teams offer complementary capabilities that can be exploited in a synergetic fashion. Robots are able to gain the access to places forbidden to humans to quickly explore the workspace. They are able to retrieve accurate measurements from environment and are disposable. Humans are able to cope with unexpected situations, using their abilities of reasoning and abstraction. Furthermore humans operate and make decisions faster than robots. These capabilities will keep humans in rescue situations for many years before robots can take over humans' tasks in life threatening situations.

3.2 System requirements and design

The rescue localization problem can be addressed as finding the position (x, y, z) and the heading (ϕ) of the mobile agent (rescuer and/or robot) with respect to a Cartesian navigation reference frame.

According to NIOSH and Italian Firefighters Corp recommendations, the localization system should:

- locate the mobile agents using lightweight wearable sensors;
- exploit, when available, a network pre-deployed in the environment;
- guarantee room-level accuracy;
- provide and continuously update the position estimate;
- adopt low cost device and computational efficient solution;
- consider only the most common activities of a rescuer, i.e., *staying still, walking, going upstairs, going downstairs.*

It is worth noticing that the wearable sensors used in localization must not interfere with the operations during mission.

In this work, to solve the localization problem the Hybrid Rescue Team Localization System (HRTLS) has been developed for rescue tasks in which fire fighters and supporting robots are exploring emergency area together with the help of a remote operator,

i.e. the incident commander. HRTLS aims at building a spatial model of the disaster area to achieve situational awareness for both external operator and rescuers.

In Fig. 3.1 the overall architecture of HRTLS is sketched. In HRTLS there are three main actors: the *supervisory entity*, the *mobile agents* (blue and red circles in Fig. 3.1), and network of landmarks deployed inside the environment. The supervisor is placed outside the emergency area for coordinating the hybrid rescue teams and collecting data from mobile platforms and information from rescuers, in order to assess the situation. Moreover, he/she is able to send orders to rescuers and control the robots. The mobile agents, both robots and rescuers, operate inside the emergency area.

The proposed localization system tries to integrate data collected from mobile agents and smart landmarks deployed in the environment. During emergency phases, indeed, rescuers only rarely know their accurate position. Human beings mostly exploit visual information for localization, which is based on recognition of an object and estimation of their distance to the object. In presence of darkness or smoke the accuracy of the localization downgrades. Furthermore, human localization is relative by nature: only when a rescuer identifies a known object (e.g., a corridor crossing or a door), he/she is able to estimate his/her position with respect to the object. It is supposed that mobile agents can be regarded as nodes of the same network, able to grab and/or forward location information with the network of landmarks. Two types of landmarks are considered: composed by passive RFID tag or by wireless sensor networks (WSN). Mobile agents are equipped with proprioceptive sensors (e.g., IMU for rescuers and encoder for mobile robots) and a radio device able to read information from the passive landmarks or to communicate with the WSN. The passive external infrastructure is composed by RFID tags used as range free sensors; on the contrary, the WSN is composed by range capable nodes. According to the rescuer requirements, landmarks are able to provide the basic location message which is composed by the following fields:

- *Geographical coordinates* provided adopting the WGS-84 standard for cartography, geodesy, and navigation or the Navigation reference frame;
- *Bearing* provide the set of the directions that can be assumed to pass through the environment where the landmark is located (e.g., a door can be cross to get inside or outside a room);

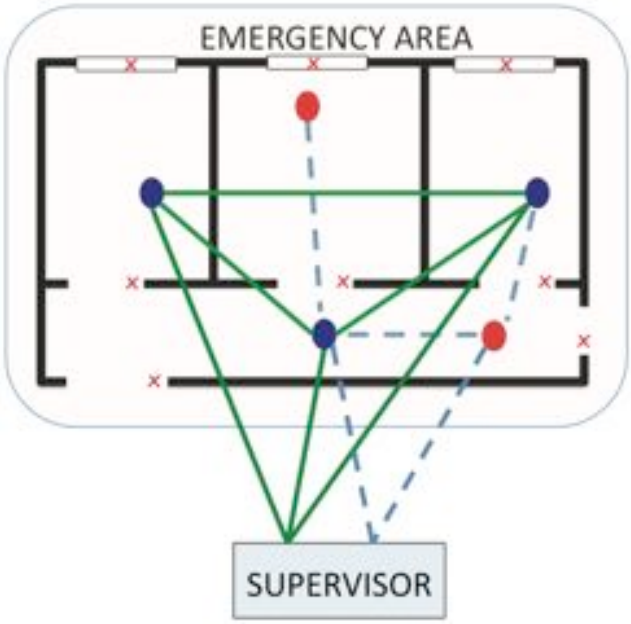


Figure 3.1: HRTLS architecture: blue circles represent rescuers while red circles are mobile robots. Dashed lines represent TETRA radio communication are represented by green solid lines and blue dashed lines wireless network. Supervisor communicates with human operator using both TETRA and wireless network and controls robots by wireless network. Red crosses represent landmarks.

- *Device classification* identifies the type of device (e.g., emergency lamp, sign, etc.) and its position in the emergency area (e.g., floor, mezzanine, corridor, etc.);
- *Sensor classification* indicates the type of the tag (i.e., passive, semi-passive, or active) or wireless device;
- *Accuracy* characterizes the shape of the electromagnetic field set off by the tag/wireless device antenna;
- *Orientation* is the direction of the tag antenna;
- *Date* represents the last time update of the information stored into the device.

To further fulfill the rescue requirements, the proprioceptive sensors for rescuer is a lightweight wearable device, mounted on the waist, while the radio device (tag reader or wireless antenna) is embedded in the handset (e.g., the TETRA radio) used during

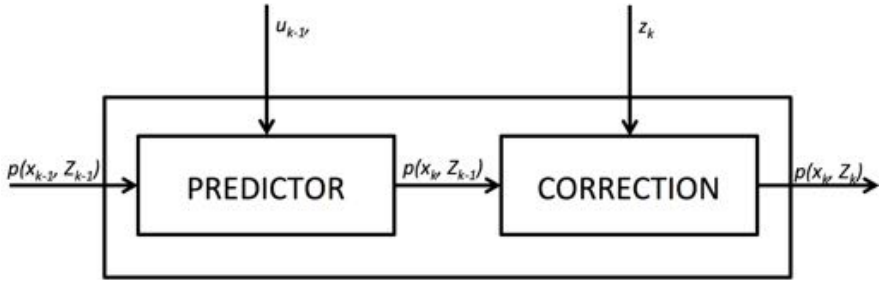


Figure 3.2: Bayesian filter.

mission. The landmarks are supposed to be embedded in the safety sign deployed in the environment. Moreover, RFID tags represent anchors, since their location is recorded during the deployment; on the contrary, WSN is activated on emergency and nodes cooperatively localize themselves. In the following, the theoretical localization model exploited in the proposed framework is presented. The novelty of the proposed approach is in the joint exploitation of fixed and mobile nodes.

3.3 Theoretical Framework

According to HRTLS framework, mobile agents explore the emergencies area. During navigation, the mobile agents are able to autonomously localize themselves. However, when *rendez–vous* occur, mobile agents and external infrastructure cooperate for improving location awareness. Due to the harsh operating condition, the localization system has to cope with a large amount of uncertainties that dwells both in the information provided by the smart environment and in the interaction between mobile agent and external infrastructure. Moreover, as stated in the requirements, the localization system should provide and continuously update the position estimate.

To this end, the localization of both mobile agents and node in the external infrastructure is addressed by adopting the well-known prediction-correction schema adopted in robotic localization. According to this approach, an efficient theoretical framework to model uncertainty is represented by Probability Theory. In this context, localization is mapped into a stochastic estimation problem and solved applying the well-known

predictor–corrector structure of Bayes filter (see Fig. 3.2).

This class of filters extends the Bayes rule to temporal estimation problems: the key idea is to recursively compute the so-called Belief:

$$Bel(x_k) = p(x_k|U_k, Z_k) \tag{3.1}$$

that is, the joint posterior density of the mobile agent/node position and heading (x_k) given all the history of the proprioceptive (U_k) and exteroceptive (Z_k) measurements up the time instant k. During the prediction step, the *Prior* $Bel^-(x_k)$ is computed and the *Posterior* $Bel^+(x_k)$ is computed in the correction. Specifically, the former, defined as:

$$Bel^-(x_k) = p(x_k|U_k, Z_{k-1}) \tag{3.2}$$

is the Belief of the mobile agent/node after the integration of the proprioceptive sensors u_k and before it receives the exteroceptive data z_k. The latter, defined as:

$$Bel^+(x_k) = p(x_k|U_k, Z_k) \tag{3.3}$$

is the *Belief* of the agent/node when exteroceptive data z_k becomes available. Applying the Total Probability theorem the $Bel^-(x_k)$ can be rewritten as

$$Bel^-(x_k) = \int_{\Xi} p(x_k|x_{k-1}, U_k, Z_{k-1}) \times p(x_{k-1}|U_{k-1}, Z_{k-1}) dx_{k-1} \tag{3.4}$$

where the *Prior* to being in x_k is given by the sum of the probabilities of coming from x_{k-1} to x_k conditioned on all the measurements so far. This equation represents the prediction step of the Bayes filter and can be further simplified by means of the *Markovian assumption* as follows:

$$Bel^-(x_k) = \int_{\Xi} p(x_k|x_{k-1}, u_k), Bel^+(x_k) dx_{k-1} \tag{3.5}$$

where the first term is an integral that depends only on the applied control u_k and the immediate preceding state x_{k-1}. This last contains all the statistical information carried by U_{k-1} and Z_{k-1}.

Moreover, by applying Bayes rule the Posterior can be expressed as:

$$Bel^+(x_k) = p(z_k|x_k, U_k, Z_k) \frac{p(x_k|U_k, Z_{k-1})}{p(z_k|U_k, Z_{k-1})} \tag{3.6}$$

The equation states that the *Posterior* is the conditional probability of observing z_k, weighted by the ratio of the prior belief of being in x_k , $Bel^-(x_k)$ and the probability of observing measurement z_k conditioned on all information so far. Introducing the *Markovian assumption*, again, the *Posterior* can be rewritten as:

$$Bel^+(x_k) = \frac{p(z_k|x_k)Bel^-(x_k)}{p(z_k|U_k, Z_{k-1})} \tag{3.7}$$

Combining Eq. 3.6 and Eq. 3.7 the correction equation can be obtained:

$$Bel^+(x_k) = \eta\, p(z_k|x_k, U_k, Z_{k-1}) \int_{\Xi} p(x_k|x_{k-1}, u_k) \times Bel^-(x_k) dx_{k-1} \tag{3.8}$$

where η represents $p(z_k|U_k, Z_{k-1})$ that is a normalization factor. Prediction Eq. 3.4 and correction Eq. 3.8 equations cannot be implemented in their general form stated in the foregoing section, as the Belief over the localization domain is a density over a continuous space, hence has infinitely many dimensions.

In this thesis, this theoretical framework is applied to both the mobile agent and WSN localization. Specifically, since a large literature on probabilistic localization in robotics is available, the focus of attention is related to rescuer and WSN. To this end, a novel localization algorithm for rescuer (the *rescuer filter*) is developed by modeling the human motion and the information from the external infrastructure. Using passive infrastructure, a strategy to cope with context information (i.e., the presence of a rescuer in a certain area) is developed. Using active infrastructure, an algorithm to cope with nonlinearities is needed.

In localization for both rescuer and active infrastructure, we assume that an initial guess on the position of the rescuer and the node of the infrastructure is known. This is not an issue for rescuer, since it is reasonable to suppose that a rescuer starts mission in a known position outside the emergency area. Concerning the WSN, the same assumption cannot be adopted, however, under certain conditions, it is possible to prove that an initial guess for the position of the node can be find. To this end, in this thesis the problem of the localizability of a network is deeply investigated and some theoretical results are presented.

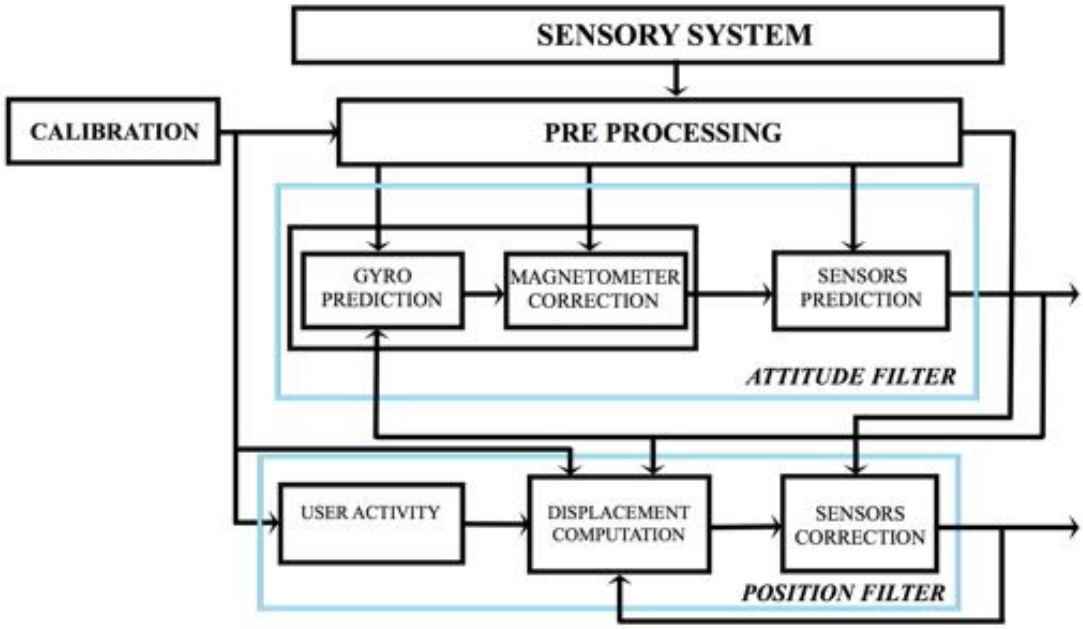


Figure 3.3: Rescuer localization filter.

3.4 Rescuer localization filter

The *rescuer filter* estimates the position $p_r = [x_r, y_r, z_r]$ and the heading ϕ_r of the rescuer with respect to a fixed reference frame, referred as *Navigation frame*. It exploits the data from IMU (i.e., accelerations, angular rates, and orientation), that are provided in the IMU reference frame that is referred as *Body frame* and data from the external infrastructure in the Navigation frame, according to the theoretical framework adopted.

In the prediction step a pedestrian dead reckoning algorithm has been developed adopting the SHS schema: it is sketched in Fig. 3.3 and it is composed by three different algorithms:

1. *Activity recognition* classifies different walking patterns, such as flat walking, going-up stairs, going-down stairs;

2. *Heading computation* estimates the attitude of the IMU wore by the rescuer with respect to the navigation reference frame. It is based on an Extended Kalman Filter (EKF) that merges data collected from gyroscopes, accelerometers and mag-

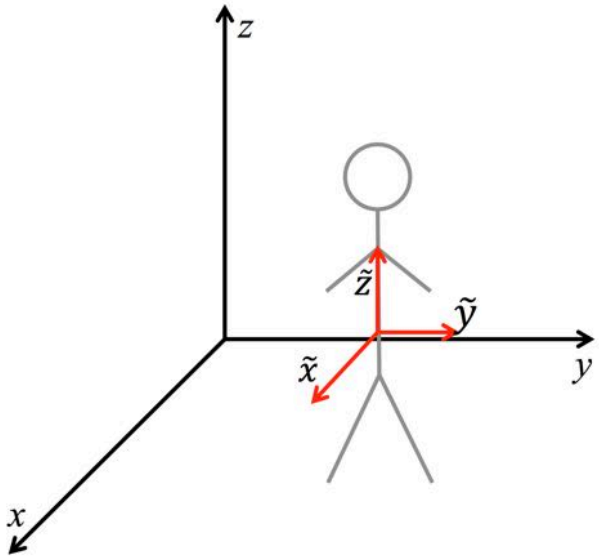


Figure 3.4: Inertial platform position.

netometers;

3. *Position tracking* updates user pose according to the walking feature and the heading.

According to the requirements, the IMU is mounted on the waist, as depicted in Fig. 3.4: the $\tilde{z}$-axis of the Body frame represents the vertical axis of the rescuer while the $\tilde{y}$-axis and the $\tilde{x}$-axis are aligned with the *medio-lateral* and the *antero-posterior* components of the acceleration vector, respectively. At the starting pose, indeed, we assume that the attitude of the Body frame corresponds to the attitude of the Navigation frame (i.e., they are aligned). Thus, the z-axis of the Navigation frame is vertical upward and represents the vertical axis of the environment; the operator moves in the $(x - y)$ plane of the Navigation frame with the x and y-axis aligned with the $\hat{x}$ and $\hat{y}$-axis of the Body frame at the starting pose. In this configuration, the gravity acceleration vector is aligned with the $\hat{z}$-axis and can be easily compensated.

These algorithms will be detailed in Chapter 4.

3.5 Localization in WSN

According to the HRTLS architecture, the active network is switched on when an emergency occurs. The network is not localized; however, the node positions $p_j = [x_j, y_j]^T \in \mathbb{R}^2$ with respect to the Navigation frame can be computed using a three steps procedure:

1. *Ranging* determines the inter-nodes distances;
2. *Node positioning* determines the position of a node with respect to anchors (i.e., nodes that know their positions with respect to the Navigation frame);
3. *Refinement* improves the accuracy of the estimate according to the proposed theoretical framework using a static model in the prediction.

The *Node positioning* step is fundamental since it computes the initial position for the Bayesian filter. To this end, a WSN Σ it is defined by a set of N fixed sensors σ_j deployed in a planar environment. Therefore, the position of a sensor σ_j can be written as:

$$p_j = [x_j, y_j]^T \in \mathbb{R}^2 \tag{3.9}$$

with respect to the Navigation reference frame.

The sensors are all equipped with an isotropic communication antenna. The node is able to sensing only points lying in the antenna maximum communication range ρ. Two nodes σ_j, σ_i use their antennas as a rangefinder for computing the Euclidean distance $d_{j,i} = ||p_j - p_i||$.

From a theoretical point of view, a WSN can be described using the disk model; that means that a WSN can be represented as a disk graph $G = \{V, E\}$ where:

- V: set of nodes $v_1, \ldots, v_N$ (i.e., the sensors $\sigma_1, \ldots, \sigma_N$)
- E: set of edges (v_j, v_i) (i.e., the presence of a communication link between sensors σ_j and σ_i).

The localization problem can be compared to the problem of finding a framework (G, P) such that the coordinate assignment P does not violate the distance constraints as in Graph Theory. A framework is a graph $G = \{V, E\}$ with a coordinate assignment

$P : V \to \mathbb{R}^2$ for the vertices of the graph. For the theoretical problem, it is possible to find a unique solution if the framework is globally rigid, i.e., the position of the nodes cannot be continuously deformed nor flipped without breaking the distance constraints [68, 21]. It is useful to notice that the computational load of the global rigidity test is high, thus not suitable to be used in the particular application scenario.

The classical approach for solving the localization problem is based on trilateration graph $G_T = \{V, E_T\}$ with $E_T \subseteq E$ [29, 5]. Trilateration is the process of determining the position of a sensor σ_j by using the information of the distances d_{ji}, d_{jh} and d_{jk} from three localized sensors (i.e., the *localization seeds*) [89, 29]. A way to localize a sensor network (composed by all distance-based sensors) is to let each sensor checks for the existence of 3 localized neighbors, while localized sensors broadcast their position to their neighbors. When an unlocalized sensor discovers 3 localized neighbors, it estimates its own location by means of trilateration and broadcasts its location to its neighbors. In this way the localization is propagated through the network, until no other sensor can be localized. If the graph G contains a trilateration graph, all the nodes will eventually be localized; when $G_T = \{V_T, E_T\}$ with $V_T \subset V$, a localizable component of the WSN is retrieved.

It is worth noticing that some nodes in the network could result unlocalized, i.e. $G_T \leq G$, due to the node positioning procedure (ranging): in this work, the problem of localizability in wireless sensor networks is addressed and some improvements with respect to the state of the art are proposed in Chapter 5.

Chapter 4

Rescuer Filter

4.1 Activity recognition

Activity recognition aims to classify different walking patterns (e.g., walking, running, etc). It can be regarded as the problem of defining the map:

$$\begin{aligned} a\colon \mathbb{R}^n &\to \mathbb{A} \\ s_k &\to c_k \end{aligned} \tag{4.1}$$

where s_k is the k–th samples of the raw signals retrieved from the IMU and $c_k \in \mathbb{A}$ is the categorical variable that defines the activity. According to the firefighters requirements, the set of the activities is defined as:

$$\mathbb{A} = \{standing\,still,\ walking,\ going\,upstairs,\ going\,downstairs\}$$

In this work, the activity recognition problem is addressed using a procedure based on the following steps:

- *Data pre-processing* eliminates bias and noise;
- *Windowing* selects a small subset of data for processing and analysis;
- *Feature extraction* computes a set of parameter useful for the classification, according to the domain of the parameters, *Time domain features*, *Frequency domain features*, and *Time-Frequency multi domain features*;

- *Classification* determines the activity associated to the signals.

Two different algorithms have been developed for identifying the different walking pattern: they differ mainly in the selected features and in the classification techniques adopted.

In both of the approaches, the signal s_k is represented by the acceleration vector retrieved by the IMU sensor, in the Body frame, that corresponds to $a = [a_{AP}, a_{ML}, a_V]^T$.

The raw signal are pre-processed by applying a 5th-order low pass digital Butterworth filter with normalized cutoff frequency $\omega_n = 6$ Hz. A time-window is applied to the samples of pre-processed accelerations so to obtain the set:

$$A^w = \{a_k | W(k) = w\} \tag{4.2}$$

where $W(k)\colon \mathbb{N} \rightarrow \mathbb{N}$ defines the mapping between the k–th time interval and the corresponding time-window w. The mapping $W(k)$ is defined so that $|A^w| = j, \forall w \in \mathbb{N}$ and $A^w \cap A^u = \emptyset, \forall w, u \in \mathbb{N}$ when $w \neq u$.

4.1.1 Activity recognition based on Decision Tree

The first algorithm developed for activity recognition is based on Decision Tree. It considers the following features in the time domain:

- the variance of the vertical acceleration in the time-window w, σ_w^2
- the amplitude of the medio-lateral acceleration in the time-window w, α_w;
- the vertical dynamics index in the time-window w, i_w.

To compute the vertical dynamics index, only the vertical acceleration A_V^w is considered. A threshold γ is set as

$$\gamma_w = \frac{\bar{a}_V^w}{n_u} \tag{4.3}$$

where $\bar{a}_V^w$ is the mean value of $a_k \in A_V^w$ and n_r represents an anthropometric parameter calibrated on the rescuer by experimental trials. Applying the threshold γ_w, the A_V^w is partitioned in two subsets $A_V^{wM} = \{a_k | a_k \in A_V^w \wedge a_k \geq \gamma_w\}$ and $a_V^{wm} = \{a_k | a_k \in$

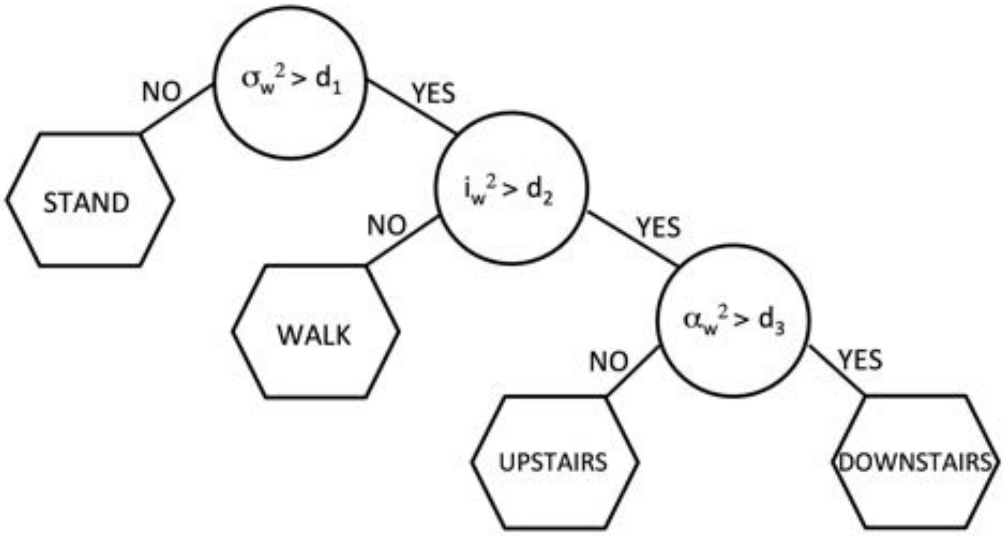


Figure 4.1: Pattern Recognition using Decision Tree.

$A_V^w \wedge a_k < \gamma_w\}$. This set are used to compute the mean positive index i_w^M and the mean negative index i_w^m as

$$i_w^M = \bar{a}_{kV}^M - \bar{a}_V^w \tag{4.4}$$

$$i_w^m = \bar{a}_{kV}^m - \bar{a}_V^w \tag{4.5}$$

where $\bar{a}_{kV}^M$ and $\bar{a}_{kV}^m$ are the mean value of the $a_k \in A_V^{wM}$ and $a_k \in A_V^{wm}$ respectively. The vertical dynamics index is the Euclidean norm if the vector $i_w = [i_w^M, i_w^m]^T$.

A decision tree is used to classify the activity in the set $\mathbb{A}$. A decision tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from an internal node are labeled with each of the possible values of the target or output feature or the arc leads to a subordinate decision node on a different input feature, each leaf of the tree is labeled with a categorical variable.

The decision tree for activity recognition is depicted in Fig. 4.1. By comparing the variance of the vertical acceleration in the time-window w with respect to a threshold d_1 standing still and moving activities are separated. In fact during the stand activity, the noise covariance is considerably reduced (see Fig. 4.2).

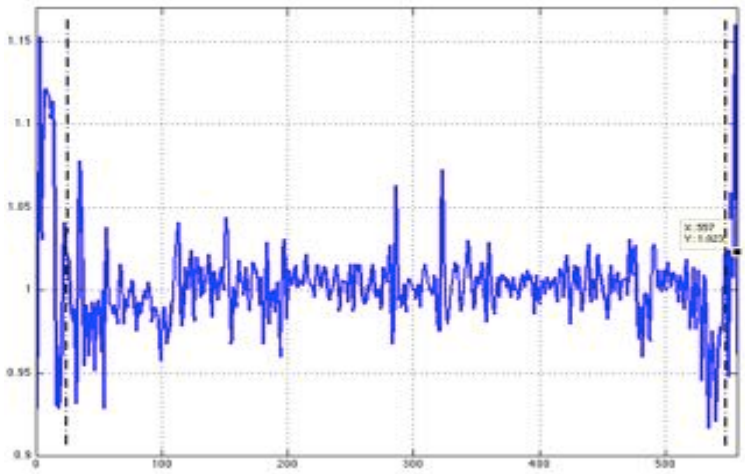


Figure 4.2: a_V $[g]$, vertical acceleration vs time $[s]$ during stand still.

The second decision is about the type of movement (i.e., walk or going up/downstairs). The decision is made by comparing the vertical dynamics index with respect to a threshold d_2.

Finally the upstairs or downstairs movement is discriminated by comparing the amplitude of the medio-lateral acceleration with respect to a threshold d_3. When the user moves up or down a stair, indeed, the CoM begins rising or falling and a shift to a single limb occurs. The amplitude of this shift is amplified going downstairs (see Fig. 4.3). The drawback of this approach is the estimation of the decision thresholds d_i, which depends on the rescuer and need to be calibrated by executing the different activities.

4.1.2 Activity recognition based on Linear Discriminant Analysis

The feature selection process entails a tradeoff between the relevance of the features and their computational cost, due to the online-algorithm constraints. According to this perspective, the Linear Discriminant Analysis (LDA) considers a limited number of features, which guarantee both a low computation effort and a desirable classification accuracy. Specifically, only 7 features are considered: they are reported in Tab. 4.1. The features $\theta_1^w, \theta_2^w, \theta_3^w$ represent the standard deviation of the components of the acceleration in A^w and are mostly used to detect the standing still activity. The features

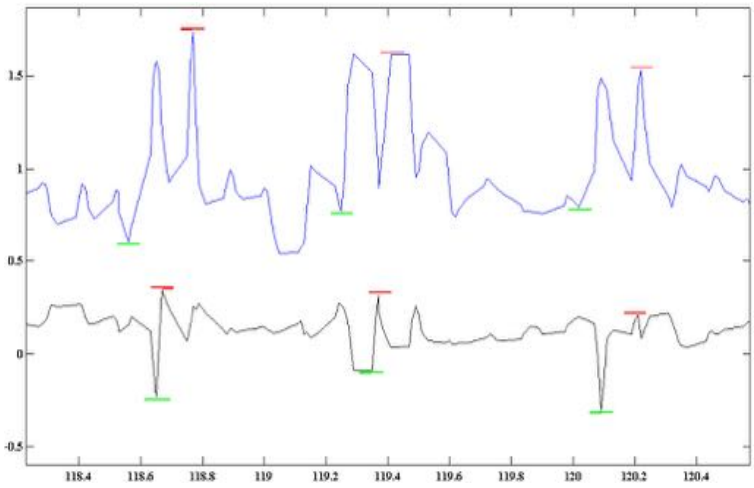


Figure 4.3: a_V (blue) and a_{ML}(black) $[g]$ vs time $[s]$ during stairs climbing: red line is ${a_V}^M$ and ${a_{ML}}^M$, green line is ${a_V}^m$ and ${a_{ML}}^m$ in the gait cycle.

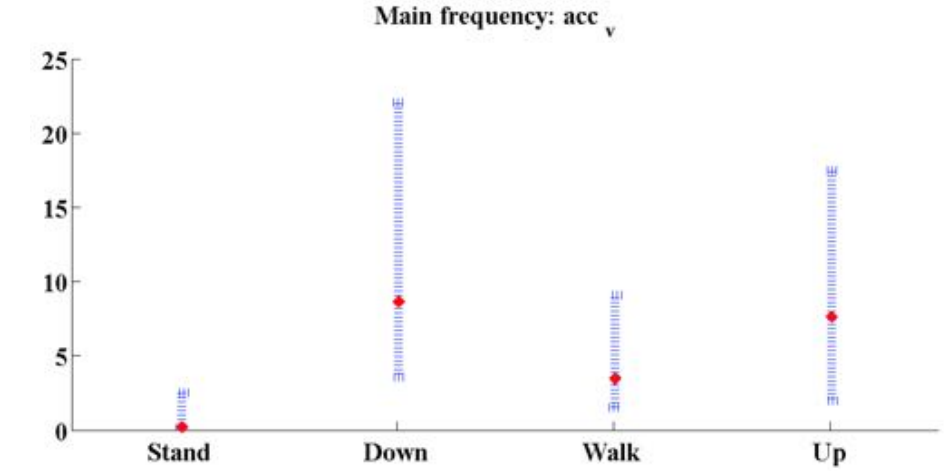
Table 4.1: Selected features θ^w on window-matrix A^w for dimension i

	A^w_{ML}	A^w_V	A^w_{AP}
$stdev(A^w_i)$	θ^w_1	θ^w_2	θ^w_3
$max(DFT(A^w_i))$	θ^w_4	θ^w_5	θ^w_6
$max(A^w_i) - min(A^w_i)$	θ^w_7	-	-

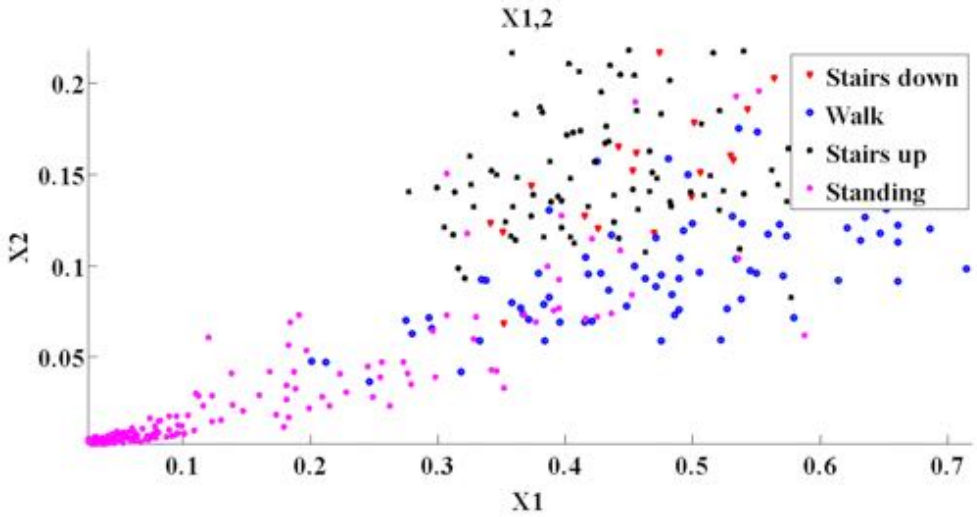
$\theta^w_4, \theta^w_5, \theta^w_6$ represent the peak value of the Discrete Fourier Transform of the components of the acceleration in A^w: they are mostly used to identify different moving activities. Finally, the feature θ^w_7 represents the amplitude of the medio-lateral acceleration and it is mostly used for discriminating the going up/down stairs activities. The features θ^w, summarized in Tab. 4.1, have been selected for the classification phase by evaluating their discriminant potential and their computational complexity. Each feature is computed on a column i of the window-matrix A^w.

The relevant features composing θ^w have been selected by adopting the visual representation method proposed in [65]. In fig. 4.4 two examples of this visual comparison are reported. Specifically, in fig. 4.4a an example of the distribution of the classes along a feature (θ^w_5) is reported. In fig. 4.4b, a comparison of the distribution of the classes across two features (θ^w_2 and θ^w_7) is proposed.

LDA is a method used in statistics, pattern recognition, and machine learning to



(a) Distribution of the classes along feature θ_5^w



(b) Distribution of the classes across two features θ_7^w (x-axis) and θ_2^w (y-axis)

Figure 4.4: Distribution of the classes: The red cross indicates the median of the redistribution. In 4.4a the blue bar indicates the range between 0.1 and 0.9 quantiles.In 4.4b the blue circles represent *walk*, the red triangles are *going downstairs*, the black crosses mean *going upstairs*, and the magenta stars represent *standing*.

find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

At each stage, LDA selects the class label, which minimizes the expected classifica-

tion loss

$$\hat{c} = \arg\min_{c \in \mathcal{C}} \sum_{k \in \mathcal{C}} \hat{P}(k|\theta^w) L(c|k) \tag{4.6}$$

where $\hat{P}(k|\theta^w)$ is the posterior probability of class k for observation θ^w, and $L(c|k)$ is the loss incurred when classifying an observation as c given its true class is k. The posterior probability is defined by the Bayes theorem as

$$\hat{P}(k|\theta^w) = \frac{\hat{P}(\theta^w|k)\hat{P}(k)}{\hat{P}(\theta^w)} \tag{4.7}$$

where $\hat{P}(\theta^w|k)$ is the likelihood of the observation, $\hat{P}(k)$ is the prior probability distribution of class k, and $\hat{P}(\theta^w)$ is a normalization factor given by $\hat{P}(\theta^w) = \sum_c \hat{P}(\theta^w|c)\hat{P}(c)$. The likelihood is approximated by a normal distribution

$$\hat{P}(\theta^w|k) = \frac{1}{\sqrt{2\pi|\Sigma_k|}} \exp\left(-\frac{1}{2}(\theta^w - \mu_k)^T \Sigma_k^{-1} (\theta^w - \mu_k)\right) \tag{4.8}$$

where μ_k and Σ_k are respectively the mean and the covariance of the feature vector distribution under class k. The prior distribution is assumed to be uniform.

The main advantage of this approach is that the LDA parameters are independent from the rescuer.

4.2 Heading computation

The heading measures the orientation of the rescuer with respect to the Navigation reference frame. In this framework, two different approaches based on quaternion have been implemented: the first one is based on the Extended Kalman Filter (EKF), [88, 34] the second one on the Complementary Filter (CF) [97, 39].

Both approaches consider all the data retrieved from the IMU (i.e., the data from tri-axial acceleration, gyroscope, and magnetometer) and exploit the device position. At the starting pose, indeed, we assume that the attitude of the Body frame corresponds to the attitude of the Navigation frame (i.e., they are aligned). Specifically, the $\tilde{z}$-axis of the Body frame represents the vertical axis of the rescuer while the $\tilde{y}$-axis and the $\tilde{x}$-axis are aligned with the *medio-lateral* and the *antero-posterior* components of the acceleration vector, respectively. At the same time, the z-axis of the Navigation frame is vertical upward and represents the vertical axis of the environment; the operator

moves in the $(x-y)$ plane of the Navigation frame with the x and y-axis aligned with the $\hat{x}$ and $\hat{y}$-axis of the Body frame at the starting pose. In this configuration, the gravity acceleration vector is aligned with the $\hat{z}$-axis and can be easily compensated. Knowing the attitude of the two frames at the starting pose, we can compute the attitude $\imath hi = [theta, \psi, phi]$ of the Body frame with respect the Navigation frame by simple rotation. Finally, the heading can be calculated as the yaw angle.

4.2.1 EKF attitude filter

The *Attitude Filter* estimates the attitude of the Body frame with respect to the fixed Navigation frame. It is based on an Extended Kalman Filter (EKF) that merges data collected from gyroscopes, accelerometers and magnetometers. The vector state is represented by means of quaternions: these mathematical entities need less computational effort in recursive updating and avoid the singularity issues that affect angular descriptors, like Euler angles. Quaternions can be defined as:

$$q = [\eta, \epsilon]^T \tag{4.9}$$

where $\eta = cos\frac{\theta}{2}$ is the scalar part of the quaternion, $\epsilon = sin\frac{\theta}{2}r$ is the vector part of the quaternion, θ is the rotation about the *Euler axis*, and r is the unit vector of the Euler axis with respect to a Cartesian reference frame *O–xyz*.

In the prediction step, the following differential equations is used to evaluate the angular motion of a rigid body:

$$\frac{d}{dt}q = \Omega q \tag{4.10}$$

where:

$$\Omega = \frac{1}{2}\begin{bmatrix} 0 & -\omega_z & \omega_y & \omega_x \\ \omega_z & 0 & -\omega_x & \omega_y \\ -\omega_y & \omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}^T \tag{4.11}$$

in which $\omega = [\omega_x, \omega_y, \omega_z]$ represents the angular velocity measured by the IMU in the Body frame, where the time-dependence has been omitted for sake of simplicity.

The corresponding discrete-time model is as follows:

$$\begin{cases} \hat{q}_k = e^{(\Omega_k \Delta t_k)} \hat{q}_{k-1} = \Phi_k \hat{q}_{k-1} \\ \hat{q}_0^T = \hat{q}(0) \end{cases} \tag{4.12}$$

where $\Delta t_k = [t_{k-1}, t_k]$ is the sampling time interval and $\Phi_k = e^{(\Omega_k \Delta t_k)}$. The quaternion q_k is computed at each sampling time k, starting from the initial condition q_0.

The initial condition q_0 is obtained from the acceleration considering the rescuer standing still at the beginning of the mission:

$$q_0 = \frac{\left[-\frac{a_{0y}}{\sqrt{2(1-a_{0z})}},\ \sqrt{\frac{1-a_{0z}}{2}},\ 0,\ \frac{a_{0x}}{\sqrt{2(1-a_{0z})}} \right]^T}{\left\| \left[-\frac{a_{0y}}{\sqrt{2(1-a_{0z})}},\ \sqrt{\frac{1-a_{0z}}{2}},\ 0,\ \frac{a_{0x}}{\sqrt{2(1-a_{0z})}} \right] \right\|} \tag{4.13}$$

where $a_0 = [a_{0x}, a_{0y}, a_{0z}]$ is the acceleration vector measured by the IMU in the Body frame at the beginning of the experiment. It is worth noticing that q_0 is a unit vector. In order to preserve this feature, the estimated value of the quaternion q_k is normalized in both the prediction and correction steps of the attitude filter.

The covariance matrix of the prediction step can be expressed as:

$$P^q_{k|k-1} = \Phi_k P^q_{k-1|k-1} \Phi_k^T + Q_k \tag{4.14}$$

where Q_k is the accuracy of the gyroscopes.

The measurement z_k is built by stacking the accelerometer and magnetometer measurement vectors as shown in the following equation:

$$z_k = \begin{bmatrix} a_k \\ m_k \end{bmatrix} \tag{4.15}$$

The expected measurement $\hat{z}_k$ from accelerometers can be computed according to:

$$\hat{a}_k = h_a(\hat{q}_{k|k-1}) = K_a R(\hat{q}_{k|k-1}) g \tag{4.16}$$

where $\hat{a} = \begin{bmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \end{bmatrix}^T$ represents the acceleration in the Body frame, K_a is the scale factor matrix depending on the device used, g is the acceleration gravity vector, and

$R(\hat{q}_{k|k-1})$ is the rotation matrix from the Body frame to the Navigation frame:

$$R(\hat{q}_{k|k-1}) = \begin{bmatrix} 2(\eta^2+\epsilon_x^2)-1 & 2(\epsilon_x\epsilon_y-\eta\epsilon_z) & 2(\epsilon_x\epsilon_z-\eta\epsilon_y) \\ 2(\epsilon_x\epsilon_y-\eta\epsilon_z) & 2(\eta^2+\epsilon_y^2)-1 & 2(\epsilon_y\epsilon_z-\eta\epsilon_x) \\ 2(\epsilon_x\epsilon_z-\eta\epsilon_y) & 2(\epsilon_y\epsilon_z-\eta\epsilon_x) & 2(\eta^2+\epsilon_z^2)-1 \end{bmatrix} \tag{4.17}$$

It is worth underlying that data collected from accelerometers can be used only when the rescuer is in still position, otherwise the gravity cannot be compensated; therefore, a validation gate is set up and the acceleration correction is performed only when $|\ ||a_k|| - ||g||\ | < \varepsilon_a$, where $||\cdot||$ is the Euclidean norm and a_k is the vector of the accelerometer measurements.

The expected measurement from the magnetometer can be computed according to the following equations:

$$\hat{m} = h_m(\hat{q}_{k|k-1}) = K_m R(\hat{q}_{k|k-1}) d_m \tag{4.18}$$

where $\hat{m} = \begin{bmatrix} \hat{m}_x & \hat{m}_y & \hat{m}_z \end{bmatrix}^T$ represents the magnetic field vector in the Body frame, K_m is the scale factor matrix depending on the environment and d_m is the Earth magnetic field. To prevent the use of magnetometer measures affected by large magnetic disturbances, a matching test is set up, so the update is performed only when $|\ ||m_k|| - |d_m|\ | < \varepsilon_m$ and $||m_k - m_{k-1}|| < \varepsilon_{\Delta m}$, where m_k is the vector of the magnetometer measurements.

When the measurements are available, i.e. they have passed the test, the estimate is updated according to the EKF equations. The correspondent covariance matrix is given by:

$$\begin{aligned} S_k &= H_k P^q_{k|k-1} H_k^T + V_k \\ K_k &= P^q_{k|k-1} H_k^T S_k^{-1} \\ \hat{q}_{k|k} &= \hat{q}_{k|k-1} + K_k[z_k - h(\hat{q}_{k|k-1})] \\ P^q_{k|k} &= (I - K_k H_k) P^q_{k|k-1} \end{aligned} \tag{4.19}$$

where H_k is the Jacobian of the observation vector $h(\cdot) = [h_a^T(\cdot), h_m^T(\cdot)]^T$, V_k is the covariance matrix of the measurements, and K_k is the Kalman gain matrix.

4.2.2 Complementary Attitude Filter

The Complementary Filter for orientation, as the EKF, allows to compute the attitude of a rigid body to which the IMU is attached. The quaternions are defined as Equation 4.9 in Sec. 4.2.1.

According to the theory of Complementary Filter, this algorithm combines two different set of data, the static and the dynamic one [11]. Static data contains the information generated by accelerometer and magnetometer, while dynamic data is obtained by gyroscope. Using static data the *static quaternion* q_σ is obtained, while dynamic data provides *dynamic quaternion* q_δ. Static quaternion q_σ is computed by using the Factored Quaternion Algorithm (FQA). FQA is a geometrically intuitive algorithm for determining orientation of a static or slow-moving body from measured acceleration and local magnetic field vectors. FQA provides the static quaternion $q_{\sigma,k}$ using the acceleration from IMU and local magnetic field measurements from magnetometer, as in [113], when new measurements are available.

The static quaternion $q_{\sigma,k}$ is useful when sensor module is stationary or slowly moving. In this case, indeed, accelerometer and magnetometer can provide absolute body orientation. However, when the angular rate measurement is high, the accelerometer and magnetometer measurements are no longer accurate and gyroscope measures need to be used. To evaluate the angular motion of a rigid body the dynamic quaternion q_δ is computed according to the Equations 4.10–4.12.

Following [12], the corrected quaternion can be computed as:

$$q_k = \gamma_k q_{\sigma,k} + (1 - \gamma_k) q_{\delta,k} \tag{4.20}$$

where γ_k is the filter gain weighting the contribution of the static and the dynamic quaternion. The quaternion q_k is obtained in an adaptive way: the filter gain γ_k, indeed, is set according to the reliability of the measurements retrieved by accelerometer and magnetometer.

4.2.3 Heading hybridization

The heading estimate $\bar{\phi}_k$ can be further refined by the data provided by the passive/active sensor in the location message. The *bearing* $\varphi_i = [\varphi_{i1}, \varphi_{i2}]$, indeed, gives in-

formation about the directions that the mobile agent can assume when passing through the environment where the sensor is located. A data association problem is solved to choose the proper bearing, thereafter a correction is performed. The data association is solved by

$$\hat{\varphi}_i = \min\{|\varphi_{i1} - \hat{\phi}_{k|k}|, |\varphi_{i2} - \hat{\phi}_{k|k}|\} \tag{4.21}$$

The refinement of the heading is computed as:

$$\hat{\phi}_{k|k} = \begin{cases} \hat{\phi}_{k|k} & |\varphi_{i1} - \hat{\phi}_{k|k}| < \epsilon_\varphi \\ \hat{\varphi}_i & \text{otherwise} \end{cases} \tag{4.22}$$

where ϵ_φ is a threshold chosen by experimental trials.

4.3 Position tracking

The *Position tracking* recursively provides the position $p_{r,j} = [x_{r,j}, y_{r,j}, z_{r,j}]^T$ of the rescuer at each sample time j, determined by an activity recognition event. The position estimate is performed according to the approach in Sec. 3.3. It exploits the output of the *Activity recognition* and the *Attitude Filter* to compute the heading and assumes that the initial pose of the rescuer is known. The heading of the rescuer is therefore a parameter for the filter: the resulting state transition model is linear. Two different correction steps are performed, depending on the passive or active infrastructure. In the first case, the sensor provides only context information (i.e., presence of the rescuer inside a certain area), in the second case the infrastructure provides information on the distance between the sensor and the rescuer, that is modeled as a nonlinear map.

4.3.1 Position prediction

In the prediction step, the position $p_j = [x_j, y_j, z_j]^T$ of the rescuer with respect to the Navigation frame and the corresponding accuracy P_j^p is computed according to the following equation:

$$\begin{aligned} \hat{p}_{j|j-1} &= \hat{p}_{j-1|j-1} + l_j, \\ \hat{P}^p_{j|j-1} &= \hat{P}^p_{j-1|j-1} + Q_j^p \end{aligned} \tag{4.23}$$

where l_j is the displacement in the 3D environment that depends on the activity c_k recognized and $Q_j^p = l_j Q^p l_j^T$, where Q^p is the model uncertainty. Specifically, if the activity recognized is *standing still*

$$l_j = [0, 0, 0]^T$$

When the activity is *walking*, the displacement $\|l_j\|$ is computed using the vertical acceleration a^V using the Weinberg approach [100]

$$\|l_j\| = \beta \sqrt[4]{a_{j,M}^v - a_{j,m}^v} \tag{4.24}$$

where $|\cdot|$ is the Euclidean norm $a_{j,M}^V$ and $a_{j,m}^V$ are the maximum and the minimum vertical acceleration during a step event and β is a parameter depending on the user, that has to be experimentally identified. Therefore,

$$l_j = [\|l_j\| cos\bar{\phi}_j, \|l_j\| sin\bar{\phi}_j, 0]^T$$

where $\bar{\phi}_j$ is the average heading during the sampling interval $[j-1, j]$.

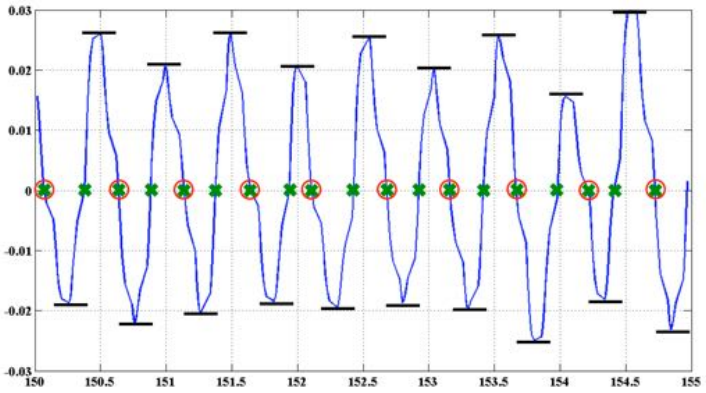
When the activity is *going up/downstairs*, the displacement l_j is computed as

$$l_j = \begin{cases} [\rho \cos\phi_j, \rho \sin\phi_j, +\zeta]^T & c_k = upstairs \\ [\rho \cos\phi_j, \rho \sin\phi_j, -\zeta]^T & c_k = downstairs \end{cases} \tag{4.25}$$

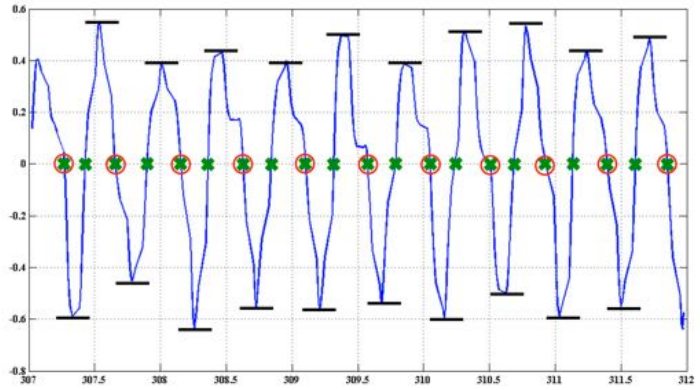
where ρ is the expected stair depth, and the vertical displacement corresponds to the stair height ζ.

It is worth mentioning that when the rescuer is moving (i.e., when the resulting activity is either *walking* or *going up/downstairs*), the time-length of a single step event needs to be determined. To this end, the vertical acceleration, pre-processed during activity recognition is exploited: a peak detection and zero crossing algorithm are applied to identify the sharp changes to the vertical acceleration associated to the heel strike as shown in Fig. 4.5.

At the end of the inertial prediction phase, the result consists in a rough position estimate affected by sensor drift errors.



(a) The number of sample in a step is ~ 30.



(b) The number of sample in a step is ~ 20

Figure 4.5: Zero-Crossing Step Detection: a zero crossing is represented using green crosses, a step is represented whit red circles and a_{VM} and a_{Vm} are represented using black line.

4.3.2 Hybridization using passive infrastructure

The correction step refines the position estimate upon tag detection. According to the HRTLS architecture, the tag provides its own *Geographical Coordinates*, its *Orientation*, and its *Accuracy* w_i. The *Geographical Coordinates* and the *Orientation* are used to compute the center of the radiation lobe T^i that is exploited to update the position of the rescuer. It is worth noticing that the *Geographical Coordinates* need to be expressed with respect to the Cartesian navigation reference frame. Since no ranging technique is adopted in this work, only the position of the rescuer is corrected, due to observability issues. When a rescuer is in the main radiation lobe of the tag, the reader receives infor-

mation from the tag and the rescuer's position is updated according to four strategies as shown in Fig. 4.6.

The first two strategies consider a single tag detection, specifically in the first one (see Fig. 4.6a), the inertial prediction estimates the rescuers outside the radiation lobe of the perceived tag i. In this case, the position and the accuracy are reset according to the following equations:

$$\begin{aligned} \hat{p}_{j|j} &= T^i, \\ P^p_{j|j} &= w_i \mathbf{I}_{2\times 2} \end{aligned} \tag{4.26}$$

where $\hat{p}_{j|j}$ represents a new initial position and the covariance $P_{j|j}$ is set according to the accuracy w_i provided by the tag.

In the second one (see Fig. 4.6b), the rescuer position is estimated inside the main radiation lobe of the perceived tag i, in this case the pose is not updated but the accuracy is eventually bounded according to the following equations:

$$\begin{aligned} &\hat{p}_{j|j} = \hat{p}_{j|j-1}, \\ &P_{j|j} = \begin{cases} P^p_{j|j-1} & \mathrm{tr}[P^p_{j|j-1}] < w_i^2 \\ w_i \mathbf{I}_{2\times 2} & \text{otherwise.} \end{cases} \end{aligned} \tag{4.27}$$

where $\mathrm{tr}[P_{j|j-1}]$ represents the uncertainty measurement.

In the last two cases, the rescuer is inside the main radiation lobe of r tags. If the inertial prediction locates the rescuer outside the radiation lobes (see Fig. 4.6c), the position $\hat{p}_{j|j}$ is updated according to the following equations:

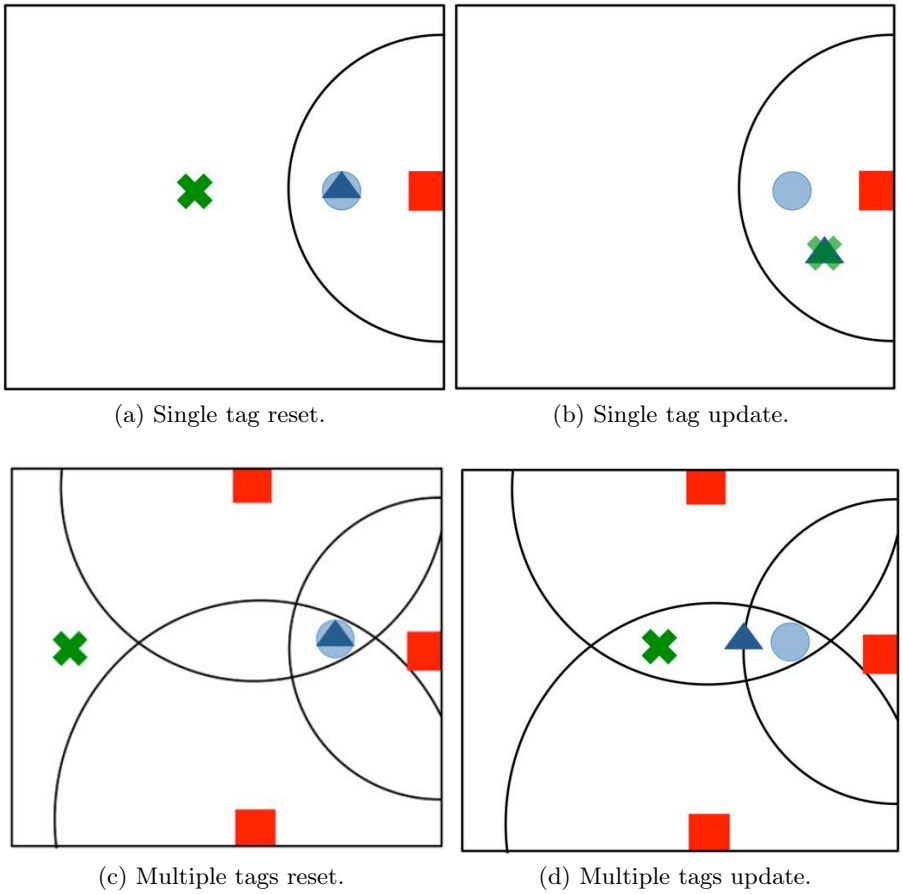
$$\begin{aligned} \hat{p}_{j|j} &= \bar{T}^i, \\ P^p_{j|j} &= \textstyle\sum_{i=1}^{r} w_i (T^i - \bar{T}^j)(T^i - \bar{T}^j)^T \end{aligned} \tag{4.28}$$

where $\bar{T}^j$ is the average center of gravity of the r tag radiation lobes.

Finally, when the inertial prediction locates the rescuer inside the radiation lobe of a subset of the r perceived tags, the position $\hat{p}_{j|j}$ is updated according to the following equations:

$$\hat{p}_{j|j} = \hat{p}_{j|j-1} + L_j(\mathbf{T}^j - [\hat{p}_{j|j-1}]_r) \tag{4.29}$$

$$P^p_{j|j} = \begin{cases} P^p_{j|j-1} & \mathrm{tr}[P^p_{j|j-1}] < \mathrm{tr}[S_j] \\ P^p_{j|j-1} - L_j S_j L_j^T & \text{otherwise} \end{cases} \tag{4.30}$$



(a) Single tag reset. (b) Single tag update.

(c) Multiple tags reset. (d) Multiple tags update.

Figure 4.6: Correction strategies: rescuer position in prediction step (green cross), rescuer position after correction (blue triangle), perceived tags (red squares), radiation lobe center (blue dot), and radiation lobe (black circles).

where $\mathbf{T}^j$ is the block vector of the coordinates retrieved from the r tags, $[\hat{p}_{j|j-1}]_r$ is a block vector stacking r times the coordinates of the position of the rescuer, $tr[S_j]$ is the tags combined uncertainty, and L_j represents a gain. This gain is computed as:

$$L_j = P_{pz,j} S_j^{-1} \tag{4.31}$$

where:

$$P_{pz,j} = \sum_{i=1}^{r} w_i (T^i - \hat{p}_{j|j-1})(T^i - \bar{T}^j)^T \tag{4.32}$$

and S_j is:

$$S_j = \sum_{i=1}^{r} w_i (T^i - \hat{p}_{j|j-1})(T^i - \hat{p}_{j|j-1})^T. \tag{4.33}$$

It is worth noticing that the correction step is performed only on the perceived tags.

4.3.3 Hybridization using active infrastructure

When information range information are available, the correction step is performed using the EKF equation, specifically the expected measurement from sensor i can be computed as

$$\hat{z}_j = \sqrt{(\hat{x}_{r,j|j-1} - x_i)^2 + (\hat{y}_{r,j|j-1} - y_i)^2} \tag{4.34}$$

where $p_i = [x_i, y_i]^T$ is the position of the sensor i, considered as a parameter.

The EKF equation can be written as

$$\begin{aligned} S_j &= H_j P^q_{j|j-1} H_j^T + V_j \\ K_j &= P^p_{j|j-1} H_j^T S_j^{-1} \\ \hat{p}_{j|j} &= \hat{p}_{j|j-1} + K_j[z_j - h(\hat{q}_{j|j-1})] \\ P^p_{j|j} &= (I - K_j H_j) P^p_{j|j-1} \end{aligned} \tag{4.35}$$

where H_j is the Jacobian of the observation map, V_j is the covariance matrix of the measurements, and K_j is the Kalman gain matrix.

Chapter 5

Wireless Sensors Network

5.1 Active Infrastructure

In HRTLS architecture, the active infrastructure is a WSN switched on during emergency. The WSN is supposed unlocalized. During WSN localization the network cooperates with the mobile agents in:

- *Loose Cooperation*: a set of nodes with known positions (*anchor nodes*) is used to retrieve the maximum localizable component and the mobile agent is used to augment this component;
- *Tight Cooperation*: no anchors are considered; the mobile agent has to localized three nodes to start the network localization.

In this work, a static WSN is deployed in a planar environment. The WSN can be described using the disk model: a node (sensor) is capable of sensing only nodes that lie within its communication range. According to this approach, a WSN can be represented by a disk graph. The localization problem can be set as finding a coordinate assignment for the non-anchor nodes. In Graph Theory, this problem is the same as finding a framework such that the coordinate assignment does not violate the distance constraints. A unique solution for the theoretical problem is proved to exist if the framework is globally rigid, i.e., the position of the nodes cannot be continuously

deformed nor flipped without breaking the distance constraints [68][21], however the computational load of the global rigidity test is high (see Sec. 3.5).

This problem is commonly solved applying a three-steps algorithm:

1. *Distance computation*: the distances between a pair of sensors are computed;
2. *Positioning*: the coordinates of the sensors are estimated;
3. *Refinement*: the coordinates estimated are updated using rescuer/network information.

The main novelty of the proposed approach is the use of *Shadow Edges* strategy for positioning in order to augment the maximum localizable component of the static WSN. In the following, the adopted strategy is reported for sake of clarity.

5.2 Positioning by Shadow Edges

Let us assume that each sensor σ_i is characterized by a communication radius $\rho > 0$ and is able to detect the presence of any sensor v_j within ρ (i.e., those sensors v_j such that $d_{i,j} \leq \rho$), obtaining also information about the distance $d_{i,j}$ between them. The resulting structure is a disk graph, since every piece of distance information $d_{i,j}$ can be obtained, given the communication radius ρ. Such an assumption is typically verified in practical situations, in particular when circular antennas are adopted.

Under the above assumptions, the sensor network may be localizable even when the underlying graph is not globally rigid, exploiting the idea illustrated in Fig. 5.1. In the figure, the sensor σ_i cannot be localized using trilateration. Although sensor σ_j is out of reach for sensor σ_i, however, the fact that σ_j is not sensed by σ_i may contribute to identify the correct position sensor σ_i. In the following, we will refer the edges like the one reported in a blue dotted line in Fig. 5.1 as shadow edges.

Definition 1 *(Shadow Edge)*: Let a disk graph sensor network, for some $\rho > 0$, be represented by a partially localized framework (G, P_l) where $V_l \in V$ contains the localized nodes (i.e., anchors and nodes localized via trilateration). Suppose that a node $v_i \notin V_l$ is connected to two nodes $v_h, v_k \in V_l$. Given the two distances $d_{i,h}$ and $d_{i,k}$ there are two admissible positions p_{i_1} and p_{i_2} for the location of node v_i (the intersections

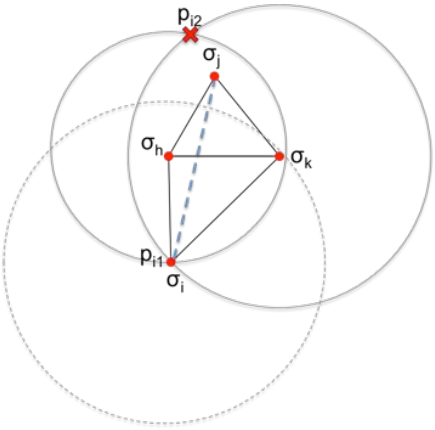


Figure 5.1: Shadow Edges Localization: sensors σ_j, σ_h and σ_k are localized, sensor σ_i is localizable by means of the *negative information* of not been connected to sensor σ_j, as represented by the shadow edge (blue dotted line).

of the circumferences centered in v_h and v_k, of radius $d_{i,h}$ and $d_{i,k}$, respectively). Let another node $v_j \in V_l$ such that $\{(v_j, v_h), (v_j, v_k)\} \in E$.

A shadow edge is an edge (v_i, v_j) such that:

1. $(v_i, v_j) \notin E$;
2. either $||p_{i_1} - p_j|| < \rho$ or $||p_{i_2} - p_j|| < \rho$

Hence, a shadow edge is a *virtual* edge that does not exist in the original graph. Notice that a shadow edge does not always exist, since the conditions of *Definition 1* may not be verified for particular frameworks. Let us denote by E_s a set of shadow edges. Note that, for any $(v_i, v_j) \in E_s$, v_i and v_j have to be considered as virtual two-hop neighbors, because their distance is $d_{i,j} \in (\rho, 2\rho]$.

Definition 2 *(Shadow localizable framework)*: Let a frame-work (G, P) with n nodes. A framework (G', P) contained in (G, P) is a shadow localizable framework if:

- G' is connected and perfectly chordal;
- for each perfectly chordal subframework (G_{sub}, P_{sub}) of (G, P) with 4 nodes which is not complete, the corresponding subframework (G'_{sub}, P_{sub}) of (G', P) is Delaunay and Gabriel.

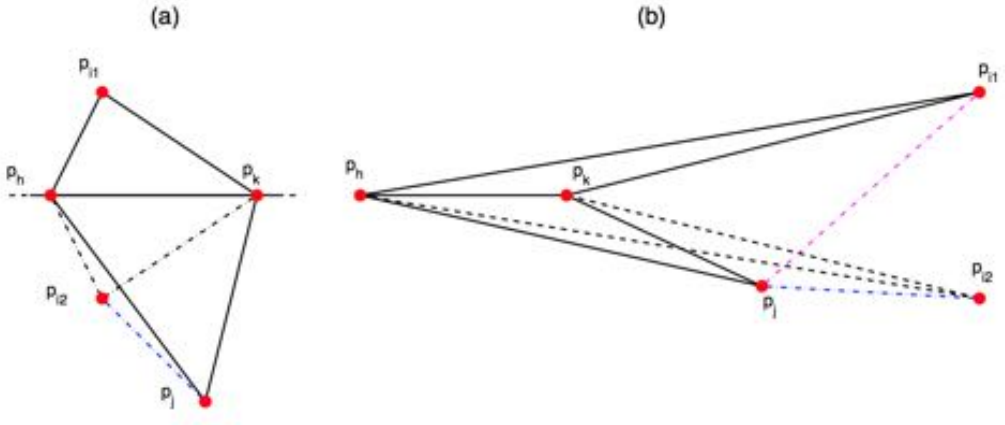


Figure 5.2: Situations considered in Lemma 1: (a) a framework which is both Gabriel and Delaunay is such that point p_{i2} lies in the lower half-plane with respect to the line that intersects p_h and p_k; moreover, the angles $\angle p_{i2}p_hp_j$ and $\angle p_{i2}p_kp_j$ are acute, hence $d_{i2,j} \leq \rho$; (b) only possible case where (G, P) is not both Delaunay and Gabriel and $di2, j \leq \rho$; in this case $p_{i2}p_hp_j$ is acute, hence $d_{i1,j}$ is smaller than ρ yielding to a contradiction.

Let us provide the following result:

Theorem 1: Let a unit disk graph sensor network with $n \leq 3$ sensors, represented by the graph $G = \{V, E\}$ and let E_s be the maximum set of shadow edges that can be obtained starting from a localized seed V_l. Let P be the position of the nodes in the sensor networks. The extended graph $G_e = \{V, E \cup E_s\}$ contains a trilateration graph $G_T = \{V, E_T\}, E_T \subseteq E \cup E_s$ *iff* the framework (G, P) contains a shadow localizable framework (G', P) with n nodes.

In order to prove Theorem 1 we need the following lemma, sketched in Fig. 5.2.

Lemma 1: Let a disk graph sensor network for some $\rho > 0$ be represented by a framework (G, P) with 4 nodes, where $G = \{V, E\}$ is such that $V = \{v_i, v_j, v_h, v_k\}$ and $E = \{(v_i, v_h), (v_i, v_k), (v_h, v_k), (v_j, v_h), (v_j, v_k)\}$.

Suppose the nodes v_j, v_h, v_k are localized and let p_{i1} and p_{i2} be the two options for the position of node vi given the position of nodes v_h, v_k and the distances $d_{i,h}$, $d_{i,k}$. Suppose further that p_{i1} is the true position of node v_1. It holds that $d_{i2,j} \leq \rho$ *iff* the framework (G, P) is Delaunay and Gabriel.

Proof 1: $\Leftarrow$: Suppose (G, P) is a Delaunay and Gabriel framework. Being (G, P)

a unit disk graph network, since $(v_i, v_j) \notin E$, it follows that $d_{i1,j} > \rho$. Since (G, P) is a Delaunay framework, the union of the triangles must coincide with the convex hull of the vertices. In this case p_{i_2}, obtained by mirroring the triangle $\triangle h_{i_1,k}$ with respect to the line overlapping with segment $p_h\hat{p}_k$, lies in the same half-plane where p_j lies, while p_{i_1} lies in the opposite half-plane; as a consequence $d_{i_2,j} < di_2, j$. The framework is Gabriel, hence both angles $\angle p_{i_2}p_hp_j$ and $\angle p_{i_1}p_kp_j$ must be acute; as a consequence, also $\angle p_{i_2}p_hp_j$ and $\angle p_{i_2}p_kp_j$ are acute. Since $d_{i,h}, d_{i,k}, d_{h,j}$ and $d_{k,j}$ are all smaller than ρ and $\angle p_{i_2}p_hp_j$, $\angle p_{i_2}p_kp_j$ are acute, we can conclude that $d_{i_2,j} \leq \rho$.

$\Rightarrow$: Suppose $d_{i_2,j} \leq \rho$ but the framework (G, P) is not both Delaunay and Gabriel. Note that, with respect to the above half-plane decomposition of the space, since $d_{i_1,j} > \rho$ and $d_{i_2,j} \leq \rho$, then $p_{i_2,j}$ must lie in the same half-plane of p_j, while p_{i_1} must lie in the other half-plane. In this case, since $\angle p_ip_hp_j$ is acute and both $d_{i_1,h}$ and $d_{j,h}$ are smaller than ρ, it follows that $d_{i_1,j} \leq \rho$; this is a contradiction.

Proof 2 *(Proof of Theorem 1)*: $\Leftarrow$: If G_e contains a trilateration graph with n nodes, then by Lemma 1 each perfectly chordal subgraph of (G, P) with 4 nodes is either complete or such that it is possible to find a shadow edge, hence (G, P) contains a shadow localizable framework (G', P) with n nodes.

$\Rightarrow$: The framework (G, P) contains a shadow localizable framework (G', P) with n nodes, which is perfectly chordal. This implies that (G', P) contains at least 3 fully connected nodes v_1, v_2 and v_3. By Lemma 1, for each non complete perfectly chordal subgraph of (G', P) with 4 nodes it is possible to obtain a shadow edge. It is therefore possible to label the remaining nodes so that vi is connected to at least 3 nodes v_j with $j < i, i = 4, \cdots, n$, hence Ge contains a trilateration graph and is generically globally rigid.

Let us provide the following corollary.

Corollary 1: A unit disk graph sensor network with $n \leq 3$ sensors, represented by a graph G can be localized over the extended graph G_e *iff* the framework (G, P) contains a shadow localizable framework (G', P) with n nodes.

The above result yields an algorithm to localize a sensor network based on shadow edges, the *Shadow Edge Localization Algorithm* (SELA). Starting with localized seeds (i.e., three localized connected non collinear nodes) we can say, without loss of generality

that $[V_l = \{v_1, v_2, v_3\}]$, the nodes are iteratively tested for localizability. Specifically, if each tested node v_i is connected to 3 already localized nodes, or to 2 localized nodes and there is a shadow edge, then it is localized. The algorithm terminates when all nodes have been localized or when the remaining nodes can no longer be localized.

The following result is a corollary of Definition 2.

Corollary 2: Let a unit disk graph sensor network with $n \leq 3$ sensors, represented by the graph $G = (V, E)$ and suppose that SELA and TLA algorithms are executed starting from the same localized seed $V_l \subset V$. Let V_S, V_T be the set of nodes localized by SELA and TLA, respectively; it holds $V_T \subset V_S$.

Proposition 1: Let a unit disk graph sensor network with $n \leq 3$ sensors represented by the graph $G = (V, E)$ and let $V_l \subset V$ be a localized seed with $3 \leq m < n$ nodes. Let V_{Sl}, V_{Tl} be the set of nodes localized via SELA and TLA algorithms, respectively, starting from the seed V_l . It holds: $max_{V_l \subset V}|V_{Sl}| \geq max_{V_l \subset V}|V_{Tl}|$

Proof 3 *(Proof of Proposition 1)*: Let V_S be the seed that maximizes the number of nodes localized by SELA, and let V_{Sl} be the set of such localized nodes. Similarly, let V_t be the seed that maximizes the number of nodes localized by TLA, and let V_T be the set of such localized nodes. From Definition 2 and Corollary 2, if V_t is used as seed for the SELA algorithm, then it holds $V_{Tl} \subset V_{Sl}$. Hence $|V_{Sl}|$ is a fortiori greater or equal than $|V_{Tl}|$, proving the statement.

Remark 1: The SELA algorithm can be easily implemented in a distributed fashion: each node is able to calculate its own position knowing the position of 3 one-hop localized neighbors and its distance from them (just like trilateration) or knowing the position of a two-hop localized neighbor if one of the 3 one-hop localized neighbors cannot be found.

5.3 Loose Cooperation

In loose cooperation the mobile agent interacts with the active network to improve its localization and augment the maximum localizable component of the WSN.

According to HRTLS architecture, the *Safe Navigation Algorithm* is designed by iteratively executing the following steps:

- *Shadow Edges Localization Algorithm* for WSN;
- *Mission Planner* for safety augmenting the maximum localizable component (i.e., the area safe for the mobile agent);
- *Rescuer Tracking* for improve the localization accuracy using range measurements in the correction step of the rescuer filter in Sec. 4.3.

The mission planner identifies a set of via pint in order to augment the maximum localizable component using a mobile agent, that can be regarded as a mobile node. To this end, we introduce the *Minimum Augmentation Problem* (MAP), in order to find a set of target way points $P_t \in \mathbb{R}^2$ for the rescuer σ_r, so that we solve the following problem:

$$\begin{gathered} \min |P_t| \\ \text{subject to} \\ (G = \{V \cup V_t, E \cup E_t\}, P \cup P_t) \text{ shadow localizable} \end{gathered} \tag{5.1}$$

where $|P_t|$ is the cardinality of P_t, V_t is the set of nodes associated to the points in P_t and E_t is the set of edges that connect a node in V_t to a node in $V \cup V_t$ provided that the euclidean distance is less than ρ, i.e., the links of the unit disk graphs that are incident to the nodes in V_t. In other words, the rescuer navigates through the way points, and in each way point it is regarded as a node of the framework, which is thus augmented. As the agent moves previously unlocalized sensors become localized.

In this section, an approximated solution is provided for the MAP problem, based on the concept of 3-2 *coverage*. A node or a point on $\mathbb{R}$ is k-covered if it has k localized neighbors, while it is k-m covered if, placing a sensor in that location, it is k-covered and there is at least one pre-existing node which is at most $(m-1)$-covered and becomes m-covered because of the insertion of the new node. In the proposed approach, the way point need to be 3-covered and must provide 2-coverage to some unlocalized node. Once the set P_t is retrieved, the rescuer moves along the safe candidate positions.

As already stated, some nodes cannot be localized by SELA, depending on the localization seeds Σ_s position and the framework (G, P). The unlocalized nodes can be divided into three sets:

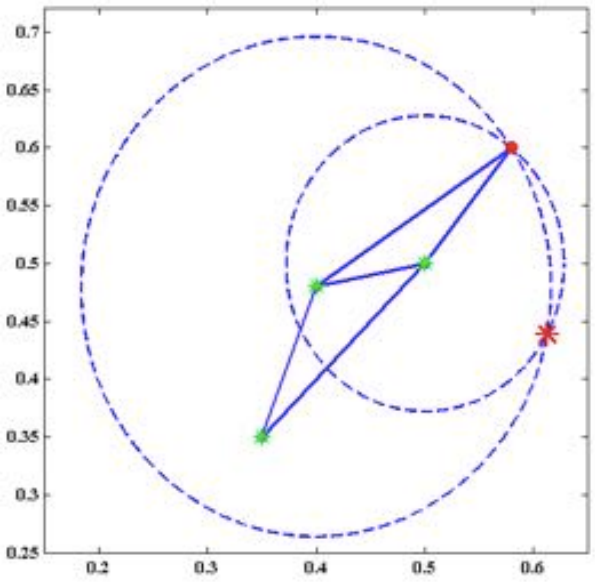


Figure 5.3: Unlocalized node (red circle) connected to 2 localized nodes (green asterisks): the way point can be either the unlocalized node position or the candidate position (red asterisk).

1. Nodes connected to two localized node;

2. Nodes connected to one localized node;

3. Nodes connected to unlocalized nodes.

Unlocalized nodes connected to two localized nodes can compute their 2 candidate positions using SELA protocol, but they are not able to discriminate the correct one using negative information (see Fig. 5.3). One of the candidate position, however, can be used as a way point for the rescuer navigation: when the rescuer reaches the way point, either it represents the unlocalized node location, or it can be discharged.

Let us refer to unlocalized nodes connected to a single node, as *pivots*. The way points for the rescuer are retrieved using a heuristic approach. A new localization seed Σ_j is selected to propagate localization exploiting the mobile node. The new seed is composed by the pivot and two localized neighbors, whose sensors monitor safe area. To guarantee localizability propagation, the way point for the mobile node needs to be 3-2 covered, i.e., 3-covered with respect to Σ_j and it has to provide 2-coverage for

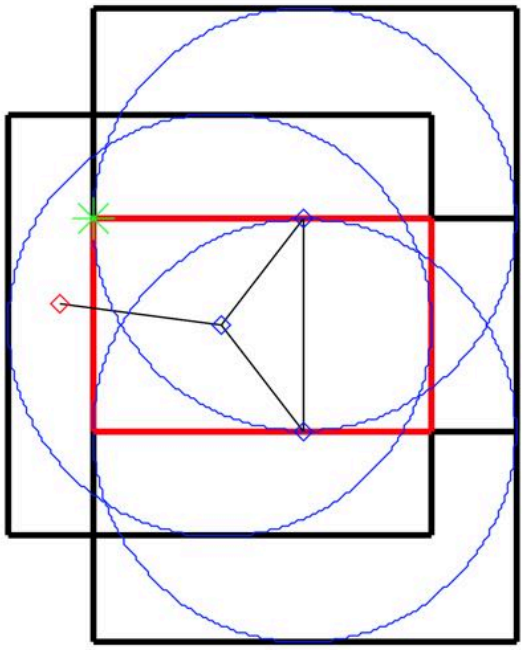
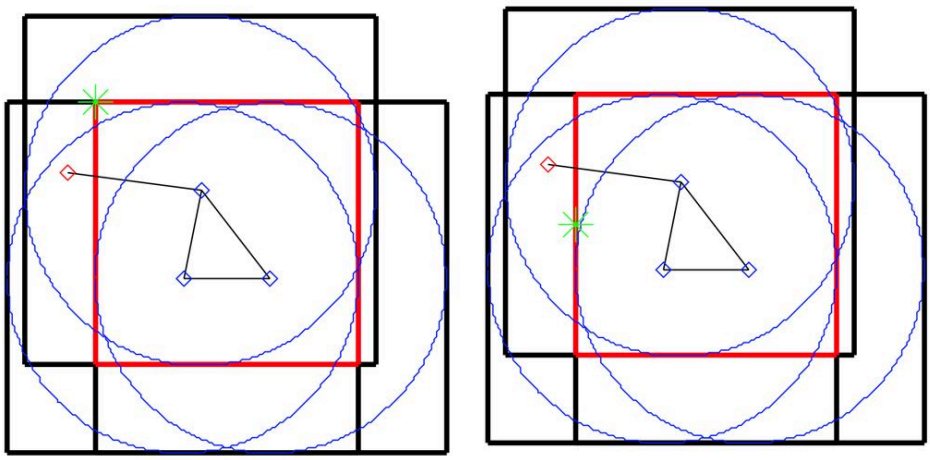


Figure 5.4: The way point (green asterisks) obtained as the nearest vertex to pivot.

the unlocalized sensors σ_j. Specifically, the way point is chosen within the region of overlap between the circles centered at the nodes in Σ_j and, at the unlocalized node σ_j. Unfortunately, this region cannot be a priori computed, since the position of sensors σ_j is unknown. The proposed approximated solution, therefore, consists in choosing a 3-covered point that lies in the intersection of the circles centered at the nodes in Σ_j and that provides 2-coverage to the pivot node, so that the SELA algorithm can localize it when the moving agent is at the designed way point. To reduce the computational burden in retrieving this point, the coverage area is approximated by using squares. In this way, the overlapping region becomes a convex polygon, whose vertexes can be easily computed knowing the vertexes of the coverage area. Specifically, the coverage area of a sensor σ_i in a planar environment is the square S_i

$$S_i = \{[x_p, y_p]^T \in \mathbb{R}^2 | x_p \in [x_i - \rho, x_i + \rho], y_p \in [y_i - \rho, y_i + \rho]\} \tag{5.2}$$

where, (x_i, y_i) is the position of the sensor σ_i and ρ is the maximum range of the node



(a) Nearest Vertex (b) Mean of Nearest Vertexes Distance

Figure 5.5: Way point estimation: if the nearest vertex is collinear (a) with the other nodes then the way point can be computed as the mean of the distance between the nearest vertexes to the pivot (b).

antenna. The overlapping area for the nodes in Σ_j is computed as

$$Q_j = \bigcap_{\sigma_i \in \Sigma_j} S_i, \tag{5.3}$$

while the intersection between the bounding box of the coverage area can be found applying the max-min algorithm.

The candidate positions for the way point are represented by the vertexes of the overlapped region Q_j. The way point is found in two different ways. Generally, it can be found as the nearest vertex to pivot (see Fig. 5.4). If the nearest vertex is collinear with the other nodes, as shown in Fig. 5.5a, then the way point can be computed as the average of the distance between the nearest vertexes to the pivot, as shown Fig. 5.5b. To further reduce the number of the way points P_t to be visited by the mobile node, unlocalized nodes that share two or three nodes in the new localization seeds can share the candidate position by using the center of mass of the position retrieved using the above procedure.

5.4 Tight Cooperation

In tight cooperation nodes in a WSN cannot be localized until, since no anchors are in the network - the localization seeds; i.e. a 3 non collinear connected node, are retrieved by the mobile agent during emergency mission. First of all we want to declare when a network is localizable and it happens according to the following theorem.

Theorem 2 *WSN Localization*:

Let consider a WSN Σ the corresponding disk graph $G = (V, E)$ and a mobile agent r whose position is $p_r = [x_r, y_r, \phi_r]$ and heading ϕ_r. Let suppose that:

- the mobile agent is localized during the motion and that is equipped with an antenna that has range $d_r = 2d$;
- it is always possible to localize the first node;
- the network is connected or can became connected if exists at least a disconnected component that is placed at $d \leq d_r$

then the network is localizable.

Proof *Theorem 2*: When the mobile agent is able to retrieve the first node of a network and a position assignment for that node can be done, three situations can be faced. If the first node is the only sensor of the network, the theorem is trivially proved. If the first sensor is connected to other nodes, the proof of the network localizability is constructive and it can be obtained by Alg. 1. If the sensor is disconnected from the remaining network, it can be perceived by the mobile agent, that is equipped with an antenna having range $d_r = 2d$, so the mobile agent can be further localize the remaining network according to Alg. 1. Once a component of a network is retrieved, other disconnected components placed at $d \leq d_r$ can be retrieved by the mobile agent, by traveling along the bounding box of the network.

In the Alg. 1, L and U are two sets that represent the set of the localized nodes and the set of the unlocalized node respectively.

At the beginning of the algorithm L and U are defined as:

$$\begin{aligned} L &= \emptyset, \\ U &= V \end{aligned} \tag{5.4}$$

Algorithm 1: WSN Localization using Mobile Agent Algorithm.

$L = \emptyset$ **AND** $U = V$
$\%\ INITIALIZATION$
Move r until $R(\hat{p}_r) \neq \emptyset$
for $V \in R(\hat{p}_r)$ **do**
  $SetPosition(v, \hat{p}_r)$
  $L := L \cup \{v\}$
  $U := U \setminus \{v\}$
end for
repeat until $|L| = 3$
for $V \in N_3(\hat{p}_r)$ **do**
  $p(v) = Trilateration(v)$
  $SetPosition(v, p(v))$
  $L := L \cup \{v\}$
  $U := U \setminus \{v\}$
end for
for $V \in N_2(\hat{p}_r)$ **do**
  $p(v) = ShadowEdge(v)$
  $SetPosition(v, p(v))$
  $L := L \cup \{v\}$
  $U := U \setminus \{v\}$
end for
if $L \neq V$ **then**
  go to **Move r**
end if

and the algorithm ends when:

$$\begin{aligned} L &= V, \\ U &= \emptyset \end{aligned} \tag{5.5}$$

During the initialization the mobile agent moves randomly in the environment until $R(\hat{p}_r) \neq \emptyset$ or in other words until:

$$R(\hat{p}_r) = \{v \in V \mid \|p(v) - \hat{p}_r\| \leq \Delta\} \tag{5.6}$$

where $\hat{p}_r$ is the position of the mobile agent and Δ is a threshold that represent the maximum distance between node v and mobile agent position $\hat{p}_r$ to set the node position as mobile agent position. This assignment can be done using the function $SetPosition(v, \hat{p}_r)$.

The initialization procedure stops when $|L| = 3$, i.e., the localization seeds is retrieved and the WSN autonomous localization can start. Specifically, nodes in WSN can be localized by using:

- the mobile agent if

$$N_1(\hat{p}_r) = \{v \in V \setminus L \mid deg_v(L, \hat{p}_r) = 1\} \; ; \tag{5.7}$$

  or

$$N_2(\hat{p}_r) = \{v \in V \setminus L \mid deg_v(L, \hat{p}_r) = 2\} \; ; \tag{5.8}$$

  and there is not any Gabriel subframework in $N_2(\hat{p}_r) \cup L$;

- the $ShadowEdge(v)$ function if $N_2(\hat{p}_r)$ and there exist at least one Gabriel subframework in $N_2(\hat{p}_r) \cup L$;

- the $Trilateration(v)$ function if

$$N_3(\hat{p}_r) = \{v \in V \setminus L \mid deg_v(L, \hat{p}_r) = 3\}. \tag{5.9}$$

The localization using trilateration and Shadow Edges has been already addressed, a different approach is used for localization exploiting the mobile agent.

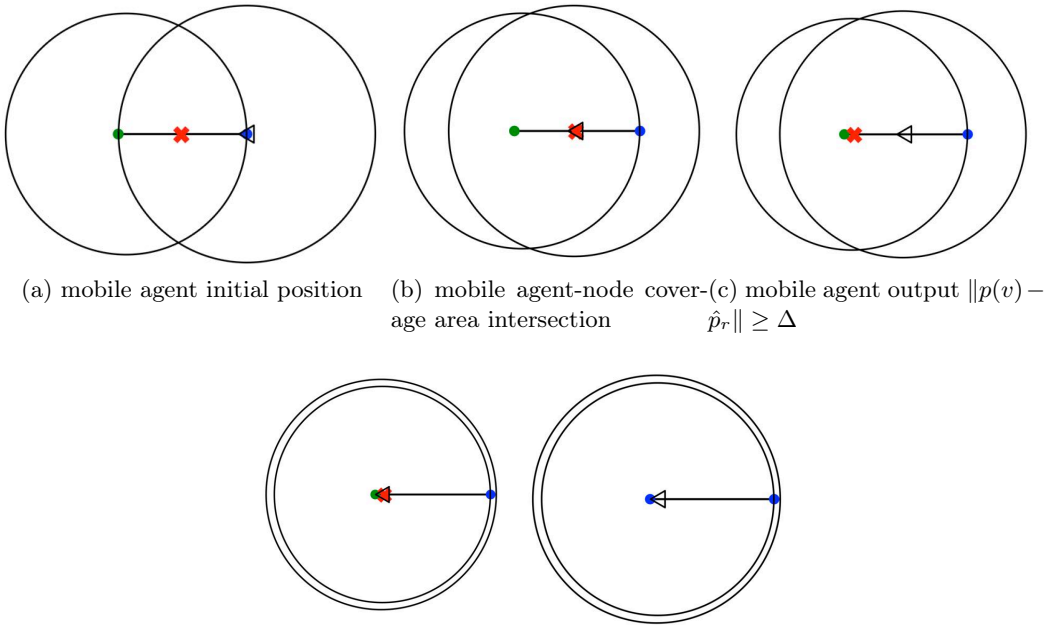
It is worth noticing that only available information's inter-nodes distances, so the mobile agent moves randomly until it can estimate a rough position for the unlocalized nodes. Once the motion direction is known the mobile agent travels in the center of mass of the estimated intersection between mobile agent coverage area and node coverage area until the unlocalized node is in the $R(p_r)$ (as shown in Fig. 5.6) and the assignment is done.

According to the algorithm and the localization strategies described it is possible to provide the following remarks.

Remark 2: The constraints to localize a WSN can be summarized as:

- the mobile agent position must be known even with a bounded uncertainty;

- the position of a node must not be known a priori, but can be estimated from the mobile agent going near to a node within a certain range Δ. The location of the network will depend from the positioning of this node;



(a) mobile agent initial position

(b) mobile agent-node coverage area intersection

(c) mobile agent output $\|p(v) - \hat{p}_r\| \geq \Delta$

(d) mobile agent-node coverage area intersection

(e) mobile agent output $\|p(v) - \hat{p}_r\| \leq \Delta$

Figure 5.6: The Mobile Agent (mobile agent) localization strategy: localized nodes (blue circle), unlocalized node (green circle), mobile agent (black triangle), and estimated position (red cross).

- the algorithm requires the knowledge of the maximum number of nodes in the network for the stop condition;
- the bounding box of the network can be obtained by min–max algorithm [5].

Chapter 6

Experimental Results

6.1 Experimental Set-Up

In this section the experiments done to evaluate HRTLS are discussed. The experiments proposed are divided according to the HRTLS structure in particular will be shown the results related to:

- Rescuer Filter;
- WSN Localization.

To evaluate the reliability one the accuracy of the Rescuer filtering, different devices has been tested. Specifically, as IMU sensors we adopt: iNEMO – STEVALMKI062V2, MPU 9150 MotionFit, and MPU 6150 embedded in Samsung Galaxy S5 Neo. The specifications of the devices are reported in Tab. 6.1, Tab. 6.2, and Tab. 6.3.

The passive network is composed by UHF passive Omni–ID Ultra Long Range RFID Tags Tab. 6.5 and the reader is the RFID CAEN A528 OEM UHF multi-regional compact Reader Tab. 6.6, the reader placed on the left shoulder of the rescuer.

Finally Active Networks are investigated. The results discussed will show in the positioning of the network according to the different strategies (Loose or Thight Cooperation) and in the second one the real cases that are implemented. In the Loose Cooperation the rescuer traveled in the environment crossing spaces labeled as safe area

by the sensors. Traveling along these areas he/she augment the localization component of the network.

In the Thight Cooperation the experiment shown the localization performed by a mobile agent both in simulated that real cases. The mobile agent used in the real experiments is an unicycle robot, KHEPERA III Tab 6.4.

The results presented here have been obtained by post-processing the data collected using MatLab. The Rescuer Filter using LDA, Complementary filter, and passive correction has been also implement on an Android device (see Appendix 7).

Table 6.1: Processors and sensors of iNEMO

DEVICE	SPECIFICATIONS
STM32F103RE	high-density performance line ARM-based 32-bit MCU with 256 to 512 kB Flash, USB, CAN, 11 timers, 3 ADCs and 13 communication interfaces
LPR430AL	2 axis gyro (roll, pitch) 300°/s full scale with analog output and optional additional filters
LY330ALH	yaw-axis gyro300°/s full scale with analog output and optional additional filters
LSM303DLH	6axis geomagnetic module: ±2g/ ±4g/ ±8g linear acceleration full scale, magnetic field configurable full scale ±1.3 to ±8.1 Gauss (max), I2C digital output
LPS001DL	pressure sensor ∠300 1100mbar absolute full scale with I2C digital output and barometer
STLM75	temperature sensor with, 55 to +125° range and I2C digital interface

Table 6.2: Sensors of MPU 9150 MotionFit

DEVICE	SPECIFICATIONS
Gyroscope	x-, y-, z-axis angular rate sensor with a full scale range of $\pm 250°/sec$
Accelerometer	x-, y-, z-axis accelerometer with a full scale range of $\pm 2g$
Compass	x-, y-, z-axis magnetometer with a full scale range of $\pm 1200\mu T$
Temperature	Untrimmed, Temperature Offset $35°C$, Linearity ($-40°C$ to $+85°C$)

All the simulations are implemented using the software MatLab and its toolbox.

Table 6.3: Processors and sensors of Samsung Galaxy S5 Neo

DEVICE	SPECIFICATIONS
Processor	Octa-Core 1.6GHz
Operative System	Android OS 5.5.1. Lollipop
MPU 6150 Gyroscope	x-, y-, z-axis angular rate sensor with a full scale range of $\pm 250°/sec$
MPU 6150 Accelerometer	x-, y-, z-axis accelerometer with a full scale range of $\pm 2g$
MPU 6150 Compass	x-, y-, z-axis magnetometer with a full scale range of $\pm 1200\mu T$

Table 6.4: Processors and sensors of KHEPERA III

DEVICE	SPECIFICATIONS
Processor	DsPIC 30F5011 at 60MHz and Arm at 600MHz
RAM	4 KB on DsPIC, 128 MB on KoreBot II Extension
Flash	66 KB on DsPIC, 32 MB on KoreBot II Extension
Motion	2 DC brushed servo motors with incremental encoders
Speed	8 Infra-red proximity and ambient light sensors with up to 30cm range, 2 Infra-red ground proximity sensors for line following applications, 5 Ultrasonic sensors with range 20cm to 4 meters
I/O	Several IO with KoreIO Extension 2 programmable LED
Communication	Standard Serial Port, up to 115kbps USB communication with KoreBot Wireless Ethernet with KoreBot and WiFi card
Extension Bus	Expansion modules can be added to the robot using the KB-250 bus
Size Diameter	130 mm Height: 70 mm
Weight	$\sim 690g$
Payload	$\sim 2000g$

Table 6.5: Specification UHF passive Omni–ID Ultra Long Range RFID tags

DEVICE	SPECIFICATIONS
Frequency	$860 \div 960[MHz]$
Temperature	$-40°C \div +6°5[C]$
EPC	$96[bits]$
User Memory	$512[bits]$

Table 6.6: Specification RFID CAEN A528 OEM UHF Reader

DEVICE	SPECIFICATIONS
Transmission power	$500[mW]$
Frequency	$867.6[MHz]$
Temperature	$-20°C \div +60°[C]$

6.2 Activity Recognition

6.2.1 Parameter Definition

To use the Activity Recognition Algorithms several tests are to compute the parameters that are necessary to estimate the activities. These parameters are:

- in the Decision Tree approach the threshold d_1, d_2, and d_3 ;
- in the LDA approach the linear classificator coefficients.

To obtain this information a set of tests are done using different users. The users considered were 10 and are different in: age, height, weight, and gender.

The users have done two different tests: in the first one a $\sim 30m$ path was traveled with different speeds (see Fig. 6.1) while, in the second one, the user have to travel up and down the stairs repeatedly and continuously (see Fig. 6.2).

According to the tests performed by the users was possible to extract the threshold used by the Decision Tree to classify the activity. To do that the data provided by all the component of the triaxial accelerometer are used, in particular Vertical and Media-Lateral acceleration are analyzed to estimate the kind of activity according to the natural pelvis oscillation during going up or down the stairs.

The threshold need to be set according to the users characteristics, for this reason it is not possible to produce a confusion matrix. It was noticed that for people with similar characteristics (usually height and weight) the thresholds are almost the same for this reason was conjecture to dived the user in classes based on these characteristics. If the threshold computed on the training test is correct the classification of the activity is correct on the 94% of the experiment.

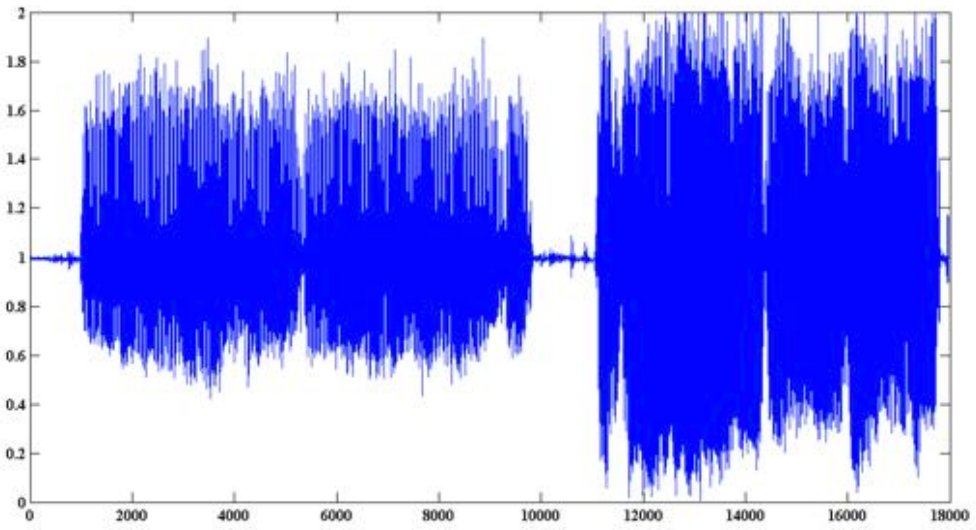


Figure 6.1: Test 1: $\sim 30m$ path with different speeds.

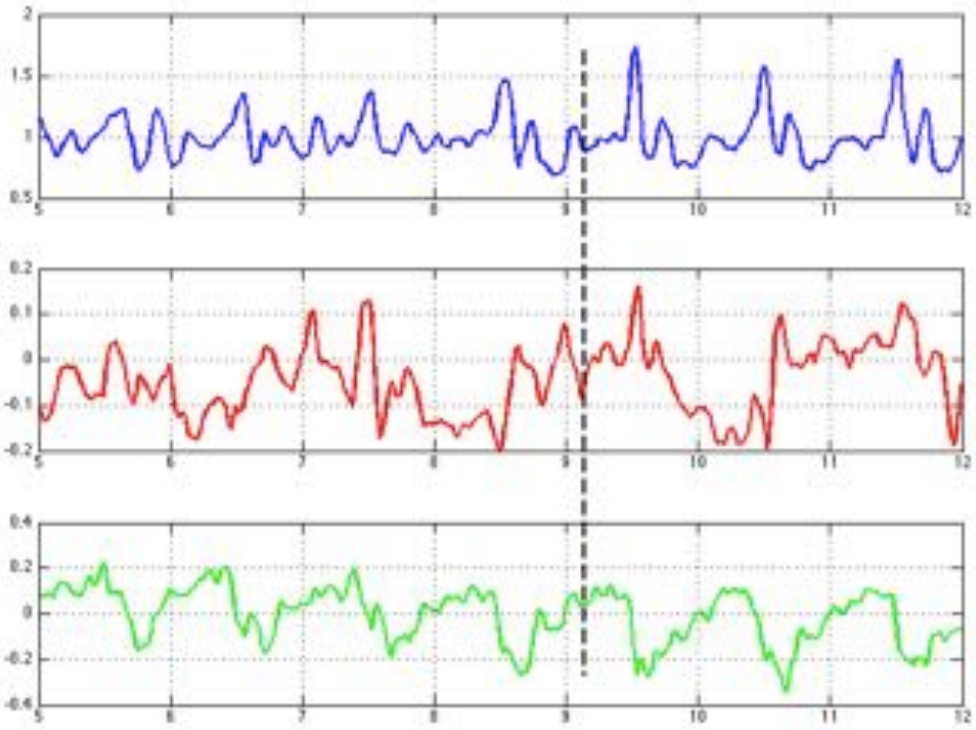


Figure 6.2: Test 2: Up and Down Stairs.

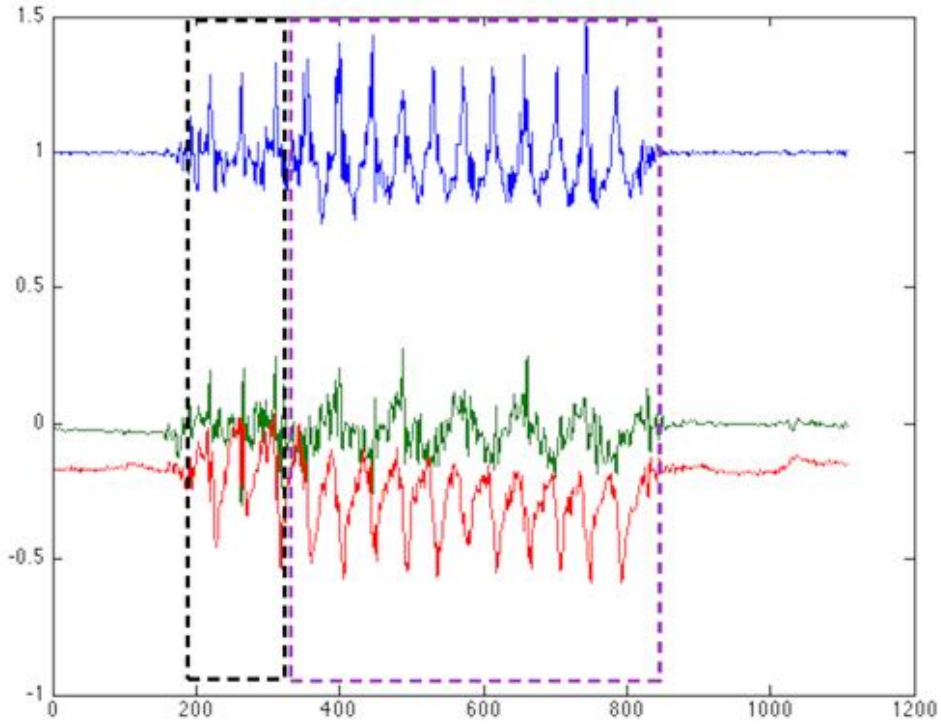


Figure 6.3: Accelerations of step-stairs test: black dashed square is walk activity and purple dashed square walk is stairs activity.

Unfortunately this technique for some particular subject, as gymnast or dancer in which the posture is usually rigid, does not recognize the activity done by the user because it is not possible to discriminate the movement from the accelerations (see Fig. 6.3).

For the LDA approach the same set of tests are used to produce the parameters used by the linear classificator. To do this the set of subject was dived in two subsets where a set was used to define the threshold using learning algorithm while the other was used to validate the results.

Also in this approach the data used to compute the threshold was obtained from the acceleration signal in particular all the three components (Vertical acceleration, Medio-Lateral Acceleration, and Antero-Posterior acceleration) are considered.

The parameter obtained in this way produce a classification independent from the user, that means that without changing parameter the classification produced is correct

Table 6.7: Classification confusion matrix: real values on rows, predicted values on columns

	stand	downstairs	walk	upstairs
stand	206	3	16	1
downstairs	2	27	10	1
walk	8	0	82	0
upstairs	0	0	15	71

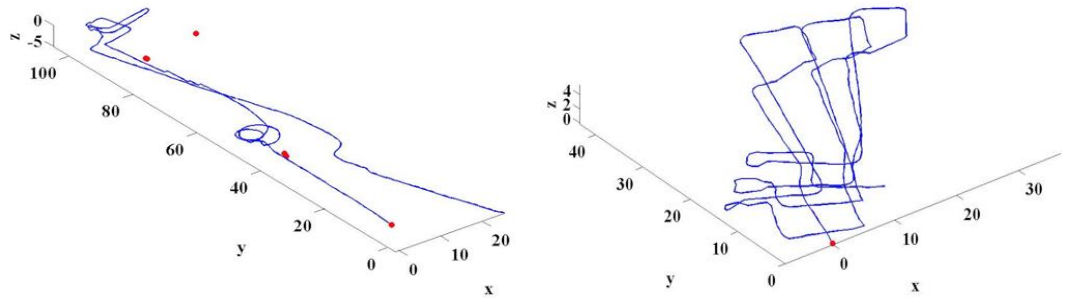
in the $\sim 88\%$ of the experiment.

The accuracy of the classifier was evaluated by collecting a dataset among 10 users performing different activity patterns: standing, walking, going upstairs, and going downstairs.

A 10-fold cross validation is performed by iteratively training the classifier on 9 users and testing its accuracy on the user that is left out. In this way, the independence between the samples in the training set and the testing set is guaranteed. The results of the experiment are shown in Tab. 6.7. The classifier achieves a global error rate of 12.44%. We observe that the choice of the loss matrix L d lead the classifier to rarely classify false-*upstairs* or false-*downstairs*. As a consequence, Tab. 6.7 shows that the major contributor to the error is the rate of false-*walk* predictions.

6.3 Rescuer Filter

The following experiment aims to show the reliability of the step detection and of the activity classification, for sake of space, only the results obtained using LDA are shown. The two components of the HRLTS are tested. Concerning the prediction step Activity recognition based on LDA and PDR are discussed while, the correction passive network are considered. The two strategy of passive network correction based on position correction and position and heading correction are reported. The experiment will be performed using different devices and testing also the Android application.



(a) Estimated path during the first experiment. Axis unit: [m]

(b) Estimated path during the second experiment. Axis unit: [m]

Figure 6.4: Results of the two experiments. Figure 6.4a and Figure 6.4b show the path estimated by the algorithm (blue line) and the checkpoints where the error is measured (red circles).

Table 6.8: Euclidean errors $[m]$

	1°	2°	3°	4°	5°	6°	3°	2°	1°
PREDICTION	0	7	8.5	10	10	22	26	22	24

Table 6.9: Euclidean errors $[m]$

	1°	1°	1°	1°
PREDICTION	0	6.8	9	10

6.3.1 Pedestrian Dead Reckoning

The goal of the experiments is to track a user wearing a waist-mounted IMU (Invensense MPU9150 MotionFit) in a realistic indoor scenario.

Two experimental settings were designed to measure the accuracy. In the first experiment, the user is walking along an uneven path to evaluate the robustness of the algorithm against non-straight lines. In the second experiment, the user is moving across different floors of a building to evaluate the performance of activity classification along a path where the user is repeatedly going up or down the stairs.

In both experiments, a set of checkpoints was fixed at known locations. The accuracy of approach is measured as the error between the estimate of the user position at the checkpoints and the user' real position.

The results of the first experiment are shown in Figure 6.4a and in Tab. 6.8. The

experiments consists of a path of approximately 280 meters combining different walking patterns. In the first phase, the user is performing a planar walk including three circular rotations. The user is then going upstairs and upstairs, and, in the last phase, the user is walking flat again back to the starting point. In Figure 6.4a, a 3-dimensional plot presents the checkpoints as red circles and the walking path estimated as a blue line. Tab. 6.8 shows the error measured during the experiment along the checkpoints deployed throughout the path. An error of 26 meters in the estimate of the position was observed after a walking path of approximately 280 meters. The classifier performs well: the percentage of acceleration samples correctly recognized is 83.20%.

The results of the second experiment are presented in Figure 6.4b and in Tab. 6.9. In this experiment, the user walks along a path of approximately 350 meters. In particular, the user repeats three times the same path combining flat walking, going downstairs and upstairs. At the end of each path, the user steps by the starting point which corresponds to the axis origin in Figure 6.4b. The aim of this experiment is indeed to stress the algorithm by repeatedly going up and down the stairs. Furthermore, iron components were in the environment during the experiment. A final error of 10 meters in the estimate of the position was observed as shown in Tab. 6.9. In this experiment the percentage of acceleration samples correctly recognize is 82.87%, although the path is more complex compared with the one in the first experiment.

Both the algorithms for the activity classification are suitable to be implemented on a standard smartphone device due to its low computational complexity.

Here, an office like scenario is considered. The testing environment is composed by a long ring-shaped corridor bounds by rooms. This environment has been selected for its closed-loop layout, that allows a better assessment of the performance of the localization algorithm.

During experimental trials, the rescuer is equipped with a waist-worn iNEMO STEVAL-MKI062V2 device (STMicroelectronics, Geneve, CH). The trial represents a penetrating mission along the corridor. The rescuer executes a distance of about 500 m. The results of the experiment are depicted in Fig. 6.4. Specifically, Fig. 6.5 shows the path of the rescuer computed by the PDR. It can be noticed that the PDR is not suitable by itself for the pose estimation: the positioning error grows along the path and, at the end of

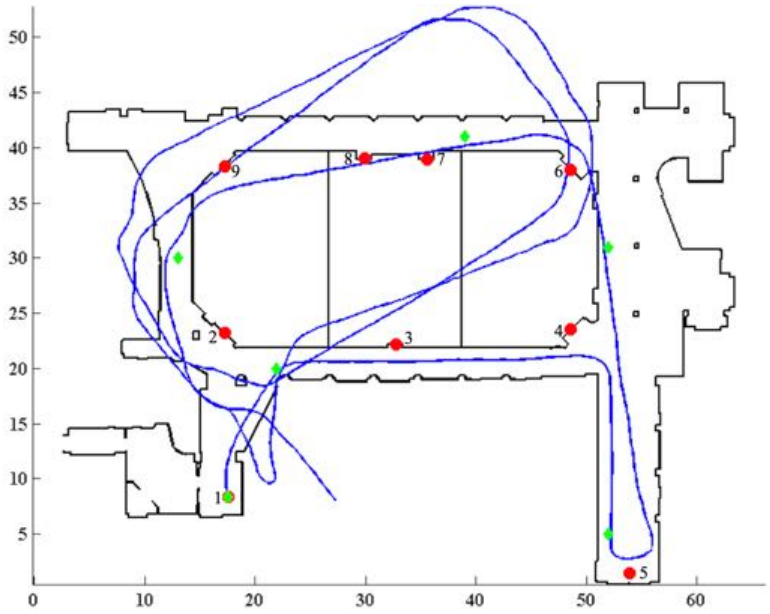


Figure 6.5: Inertial PDR estimate (blue line).

Table 6.10: Euclidean errors $[m]$

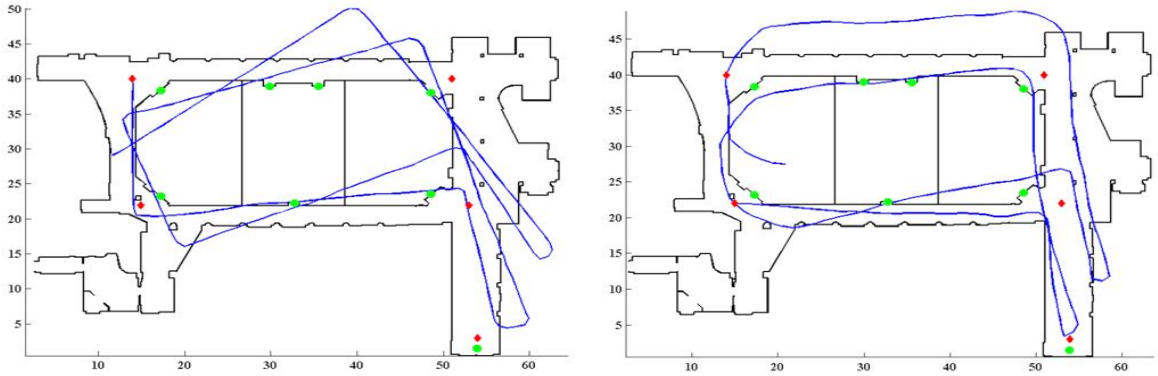
	1°	2°	3°	4°	5°	6°	1°	3°	4°	5°	6°	3°	4°	5°	6°	1°
PREDICTION	0	0.56	0.29	0.72	1.79	7.01	4.42	3.52	15.89	11.50	6.01	1.42	16.13	11.60	9.25	9.71

the experiment, the accuracy is highly downgraded.

The trial in Fig. 6.5 aim is to determine the performance of the PDR approach considering a closed-loops task. To this end 6 check points deployed in the environment and are considered. The Euclidean distance between the check points and the estimated ones (i.e., the positioning error) has been used as performance index (see Tab. 6.10).

According with Fig. 6.4, the positioning error when the pose is evaluated using PDR grows along the path and at the end the accuracy is highly downgraded.

In these tests, the goal is the tracking of an user in the environment using a smart device, waist mounted, with an embedded inertial platform and the improvement of this estimate using RFID technology. The error (i.e., e_i) during the time is estimated according to the position of some checkpoints; it is represented by the Euclidean norm of the difference between checkpoint (i.e., real position p_i) and user position (i.e., estimate



(a) Estimated path using only Inertial Platform data – User A. Axis unit: [m]

(b) Estimated path using only Inertial Platform data – User B. Axis unit: [m]

Figure 6.6: Results of the two experiments. Figure 6.4a and Figure 6.4b show the path estimated by the algorithm (blue line) and the checkpoints where the error is measured (red diamonds).

position, p_u)

$$e_i = \|p_u - p_i\|. \tag{6.1}$$

The error is computed for both the experiment (User A and User B) in the three different cases: Prediction, Correction using only position information, and Correction using position and orientation information. The device used is the Samsung Galaxy S5 Neo, running Android OS 5.5.1. lollipop and equipped with Invensense MPU6150.

The position computed by the algorithm is obtained by analyzing a stream of data provided by the IMU. Data from accelerometer and gyroscope are available at sampling frequency 100 Hz, and data from magnetometer at 50 Hz. The data computed to produce the pat of the user are corrected using the information provided by external infrastructure composed by exteroceptive sensors.

In Fig. 6.6 results of the comparative experiment is reported. The objective of the test is to highlight the robustness of approach when different users perform similar activities. Specifically, users move along a complex path (about 305 m) characterized by straight lines and curves in order to stress the heading estimation process.

Several check points have been considered and the errors are reported in Tab. 6.11 and Tab. 6.12: the average error for the user A in the experiment (Fig. 6.6a) is about 7.2 m, while the average error for the user B in the same experiment (Fig. 6.6b)

Table 6.11: Euclidean Error User A [m]: Kind of localization vs check points

Euclidean Error - User A	1	2	3	4	5	1	2	3	4	5	1
Prediction	0	1.61	2.04	4.73	6.78	5.83	7.76	8.3	14.25	15.6	11.12

Table 6.12: Euclidean Error User B [m]: Kind of localization vs check points

Euclidean Error - User B	1	2	3	4	5	1	2	3	4	5	1
Prediction	0	2.96	1.05	2.46	6.41	7.55	4.52	8.9	7.88	3.99	7.98

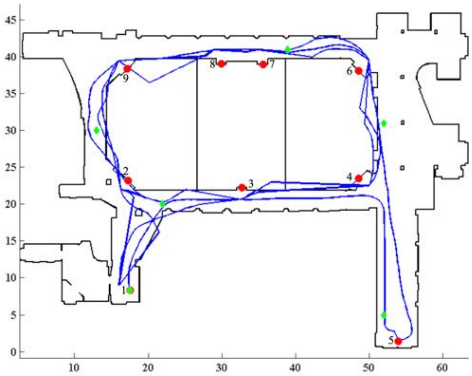
is about 5 m. The maximum error accumulated during the path is 15.6 m for user A, while user B maintains a better accuracy, having a maximum error 8.9 m.

Moreover, the heading drift is more evident for user A than for user B. This fact can be explained by noticing the different rotation strategy adopted by the two users: user A stops and suddenly turns, while user B slowly turns during motion. When a movement is too fast the dynamic quaternion is weighted more than the static one and the resulting heading estimate is affected by the gyroscope drift.

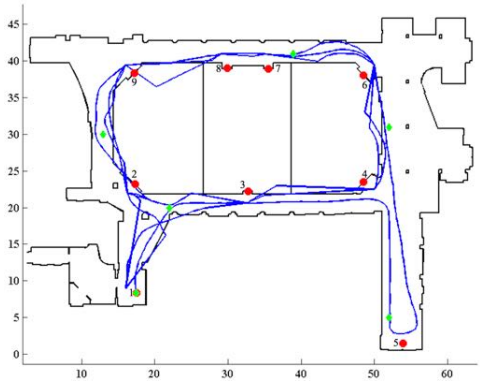
6.3.2 Passive Network Correction

In the passive network the passive OMNI-ID Dura 3000 UHF tags are used. During experimental trials, the rescuer is equipped with wearable Bluetooth UHF RFID reader on the shoulder. The reader is connected to a laptop by high speed USB and Bluetooth. The reader is able to do around 6 query to the tag during a navigation in which the user travel at $0.7m/s$ (see Appendix 7 for an evaluating of the tag-reader performance). To test the performance of the application several check points are deployed in the experiments environment. The error is computed as the distance between the check point position and the estimated one. In the following section experiments in which position correction and computation correction are discussed.

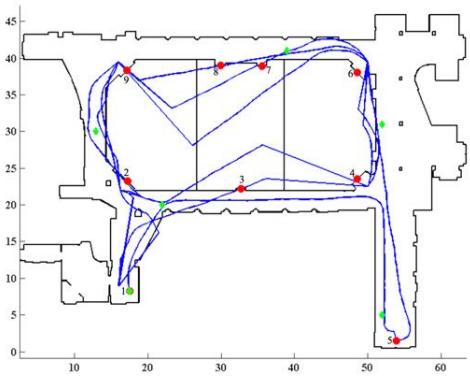
The testing environment is composed by a long ring-shaped corridor bounds by rooms. This environment has been selected for its closed-loop layout, that allows a better assessment of the performance of the localization algorithm. The sampling frequency of the iNEMO is 100 Hz, the one of RFID reader is 5 Hz, and a step is detected at $\sim$1 Hz. To this end, a synchronization procedure has been performed for data align-



(a) Estimated path using 9 tags: HIPS estimate (blue line).



(b) Estimated path using 8 tags: HIPS estimate (blue line).



(c) Estimated path using 5 tags: HIPS estimate (blue line).

Figure 6.7: Indoor results in office like environment: HIPS estimate (blue line), check points position (green stars), and tags position (red dots).

ment. In these trials, the RFID system detection area has a range of $r = 250\ cm$. A MatLab toolbox has been developed by the authors for data pre-processing and for the algorithm implementation.

The trial represents a penetrating mission along the corridor. The rescuer executes a distance of about 500 m. The objective of this trial is to analyze the improvement that the RFID technology combined with the inertial one can provide. Several configurations have been examined for assessing the impact of the RFID correction: according to this approach, a trade off between accuracy and the deployment overhead needs to be found. To this end, in the environment were deployed 9 tags according to Fig. 6.7. Depending

on the specific test performed, only a selected number of tags were used for the position correction. The results of the experiments are depicted in Fig. 6.7. Fig. 6.7a shows the system outcome by using tags for position correction: in this configuration, 9 tags are located in optimal positions and the PDR estimate is remarkably improved after the correction. Even if tag labeled as *Tag 5* (see Fig. 6.7b) is not considered the results obtained with the correction is almost near to the one obtained with the best tag deployment. In the last one result (see Fig. 6.7c) is reported the path improved using only the tags at the corners (total 5 tags) and even if the result obtained is deteriorated compared to the one with the optimal tags positions, the target performance (i.e. room-level localization accuracy) are reached.

To better understand the impact of the tags RFID position, a sub-optimal deployment of the tags in the experiment shown in Fig. 6.7 is reported in Fig. 6.8: the performance of the approach is downgraded. The tags considered are only the one placed along the corridor, and according to this tags deployment the RFID-based estimate is not able to locate the rescuer with a room-level localization accuracy.

Concerning the trial in Fig. 6.7, the objective is to determine the performance of the approach considering a closed-loops task. To this end 6 check points deployed in the environment and are considered. The Euclidean distance between the check points and the estimated ones (i.e., the positioning error) has been used as performance index.

In Tab. 6.13 the results obtained with the different tags configuration are summarized. It is possible to see how the error is related to the tags deployment. If the optimal tags configuration is considered the error is 2.62 m that is the same error obtained do not considering the *Tag 5*. On the other end it is possible to see how the results obtained using a not optimal configuration of the tags do not improve in a relevance fashion the result obtained using only the prediction, in fact the maximum error is 13.34 m. The results is improve because even if the tags positions are not the optimal one the presence of a tag, if detected, applies a correction on the path estimated. The maximum error in the correction step using the tags on the corner is around 4.28s m computed according with the performance index still fits the rescuer requirements of the room-level accuracy ($\sim$5 m error). Moreover, in this trial the impact of the correction step can be appreciated in the output, where some discontinuities arise in the estimated path. To better

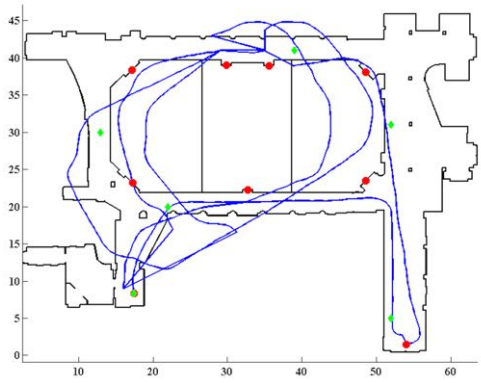


Figure 6.8: Estimated path using tags placed on the corridor (sub–optimal deployment): estimate path (blue line), check points position (green stars), and tags position (red dots).

Table 6.13: Prediction and correction Euclidean errors $[m]$

	1°	2°	3°	4°	5°	6°	1°	3°	4°	5°	6°	3°	4°	5°	6°	1°
CORRECTION ALL TAGS	0	0.56	0.28	0.72	0.38	1.96	1.65	2.11	2.62	0.18	2.4	0.41	2.06	1.60	1.34	1.65
CORRECTION ALL TAGS NOT $N5$	0	0.56	0.28	0.72	0.38	1.96	1.65	2.11	2.62	0.18	2.4	0.41	2.06	1.60	1.34	1.65
CORRECTION CORNER TAGS	0	0.56	0.28	0.72	0.47	2.82	1.65	2.53	2.62	0.18	3.58	0.95	1.72	3.35	4.28	1.65
CORRECTION CORRIDOR TAGS	0	0.56	0.28	0.72	2.05	4.8	1.65	3.30	13.34	11.12	11.19	7.84	10.7	4.54	9.43	1.65

appreciate this effect, in Fig. 6.9 the output along the x-axis and the tag activation pattern is depicted. Specifically, it highlights two different corrections occurred during the experiment. The correction step inside the A area does not change the estimate along the $x-$axis, since the output of the prediction step is accurate. On the contrary, a discontinuity can be appreciated in the correction inside the B area, since the estimate accumulated error.

It is worth noticing that in the trials the errors source depends on the heading error: the heading is not corrected by tags and a residual bias affects the estimate downgrading the overall results.

The results of the tests performed for evaluating the orientation correction. In these tests, the goal is the tracking of an user in the environment using a smart device, waist mounted, with an embedded inertial platform and the improvement of this estimate using RFID technology. The error is computed for both the experiment (User A and User B) in the two different cases: Correction using only position information, and Correction using position and orientation information.

The position computed by the algorithm is obtained by analyzing a stream of data

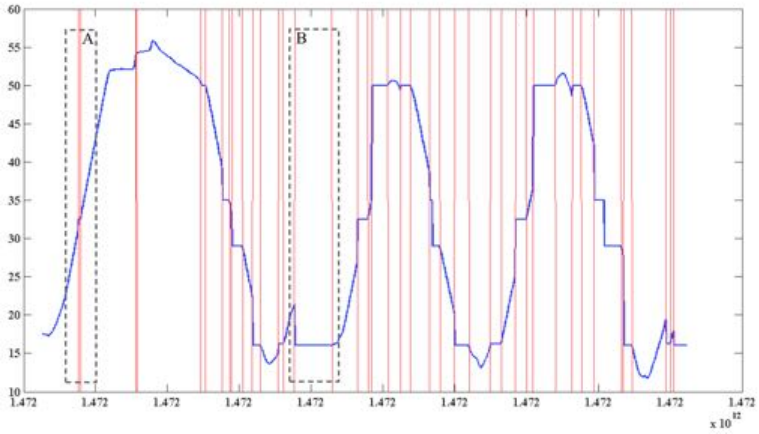
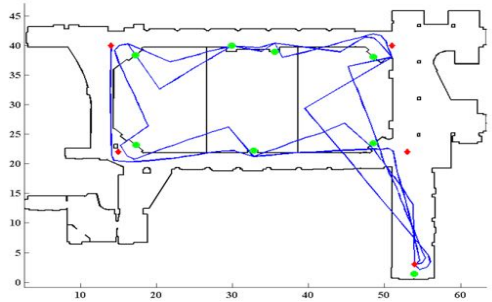


Figure 6.9: Position estimate along x-axis m (blue) and tag activation pattern (red, $ON = 60$ – $OFF = 10$) vs time ms.

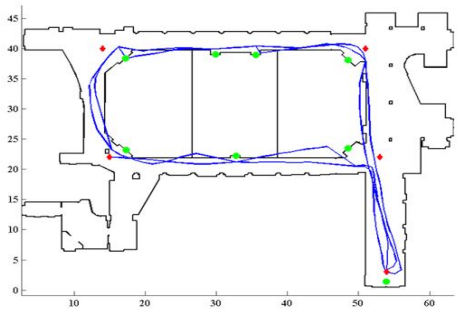
provided by the IMU. Data from accelerometer and gyroscope are available at sampling frequency 100 Hz, and data from magnetometer at 50 Hz. The data computed to produce the pat of the user are corrected using the information provided by external infrastructure composed by exteroceptive sensors.

In Fig. 6.10 results of the comparative experiment is reported. Users move along a complex path (about 305 m) characterized by straight lines and curves in order to stress the heading estimation process.

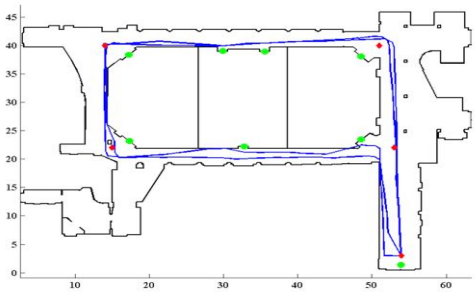
Several check points have been considered and the errors are reported in Tab. 6.14 and Tab. 6.15. In the correction using only position update the average error for the user A is 1.6 m while for user B is 1.1 m, even if the average error for both the users is around 1 m it is evident the differences between the two experiments in fact looking at the maximum errors it is possible to see how the maximum error of user A (3.9 m) it is almost the double of user B (2.62 m). Finally in the correction path using also heading updating the average error for User A (1.07 m) is almost the same of user B (0.84 m). This is explained due to nature of the error of the path of user A. This error, in fact is strictly connected to the heading of the path and using a orientation update this error is sensibly reduced and the to error are almost the same as shown also in the maximum errors that are 3.14 m for user A and 2.64 m for user B.



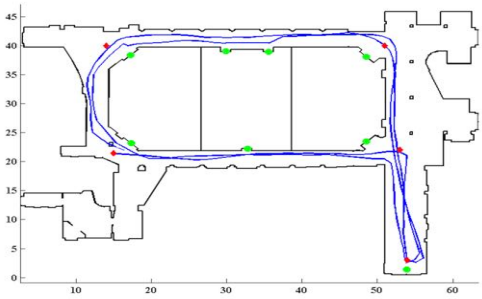
(a) Corrected path using Inertial Platform data and position information from exteroceptive sensors – User A. Axis unit: [m]



(b) Corrected Path using Inertial Platform data and position information from exteroceptive sensors – User B. Axis unit: [m]



(c) Corrected path using Inertial Platform data, position and, orientation information from exteroceptive sensors – User A. Axis unit: [m]



(d) Corrected path using Inertial Platform data, position and, orientation information from exteroceptive sensors – User B. Axis unit: [m]

Figure 6.10: Results of the two experiments. the path estimated by the algorithm (blue line) and the checkpoints where the error is measured (red diamonds). Figure 6.10a and Figure 6.10b show the path corrected using the position provided by the exteroceptive sensors and the Figure 6.10c and Figure 6.10d are the path in which also the orientation information is provided by the sensors (green dots).

Table 6.14: Euclidean Error User A [m]: Kind of localization vs check points

Euclidean Error - User A	1	2	3	4	5	1	2	3	4	5	1
Position Correction	0	1.61	1.91	0.1	2.26	1.15	1.31	3.97	0.1	2.76	2.22
Position-Heading Correction	0	1.61	1.91	0.1	1.79	0.22	0.89	3.14	0.1	1.65	0.42

6.4 Wireless Sensors Network Localization

In this section the contribution provided by the use of an active network is proposed.

In Active Network information are exchanged between rescuer or mobile agent and network. In particular there are two situation: in the first one is considered a rescuers

Table 6.15: Euclidean Error User B [m]: Kind of localization vs check points

Euclidean Error - User B	1	2	3	4	5	1	2	3	4	5	1
Position Correction	0	2.62	0.32	0.8	1.42	1.37	2.00	0.1	1.25	1.5	1.23
Position-Heading Correction	0	2.64	0.32	0.8	1.42	1.37	1.26	0.1	0.23	0.70	0.45

that travels in a network composed by range capable nodes, in the second one the same sensors are used to compose the network but WSN localization is done using the robot KHEPERA III. In the first case three anchor nodes are considered to start the localization (*Loose Cooperation*) in the second one the robot travels in the space to find the sensors that are necessary to initialize the localization algorithm (*Thight Cooperation*). The mobile agent that moves in the environment is in both the approach a unicycle robot that moves in the environment accord to the following model. The kinematic model for the mobile agent is written as

$$\begin{aligned} \dot{x_r} &= u_v \cos\varphi \\ \dot{y_r} &= u_v \sin\varphi \\ \dot{\varphi_r} &= u_\omega \end{aligned} \tag{6.2}$$

where the robot posture p_r is given by $p_r = [x_r, y_r, \varphi_r]^T$ (i.e., the position and the heading), u_v is the linear velocity of the robot and u_ω is the angular velocity. To track the robot along the path a simple static time-invariant feedback control law can be applied on each segment of the path, taking into account only the position coordinates (x_r, y_r). The tracking problem can be suitably solved by using a partial feedback linearization of the dynamic model and applying simple PD like control law in the new coordinates [23]. Supposing that a particular posture is not required when approaching the via point, the tracking problem can be relaxed into a stabilization problem considering only the position coordinates. The solution is achieved in two steps as proposed in [3]. The nonholonomic constraints are removed and a holonomic velocity u_{vh} is computed to drive the robot over the via point. Therefore, two control laws are defined: the first one is the projection of u_{vh} along the nonholonomic direction u_v and the second one steering the nonholonomic axis of the mobile agent towards the holonomic, so to reduce

the parameter β

$$u_v = K_v \parallel u_{vh} \parallel cos(\beta) \tag{6.3a}$$

$$u_\omega = K_\omega \beta \tag{6.3b}$$

where K_v and K_ω are gains. The key idea is that the eq. 6.3b independently steers the angle φ to a value that definitively let $\beta = 0$. This control fails on the via point itself, since β is undefined: to avoid hard discontinuities in the control law, the via-points have been defined as a circular region of a suitable radius r: the robot stops when it steps into this region.

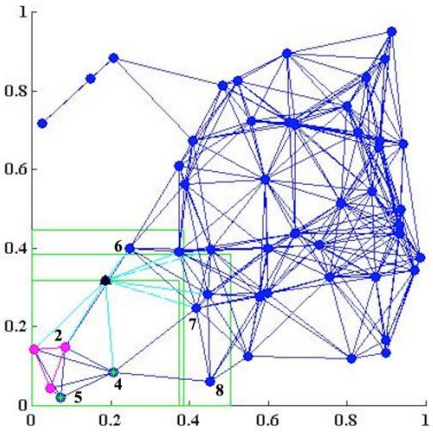
During navigation, the robot is localized by means of proprioceptive and exteroceptive sensors merged by an Extended Kalman Filter. The state transition model is given by the discrete time unicycle model in eq. (6.2), while the observation model is given by

$$d_{rj} = \parallel p_j - p_r \parallel \tag{6.4}$$

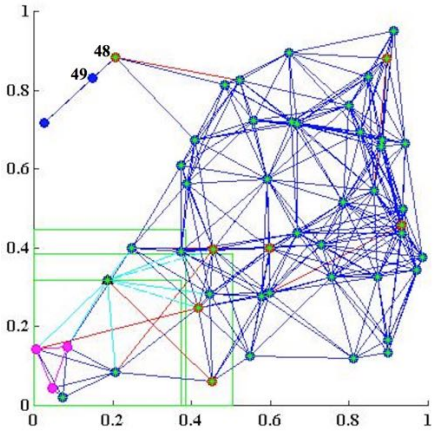
where p_r is the robot position and p_j is the position of the node σ_j, forwarded by the node during the measurement process, and $\parallel \cdot \parallel$ is the Euclidean norm in $\mathbb{R}^2$.

6.4.1 Loose Cooperation

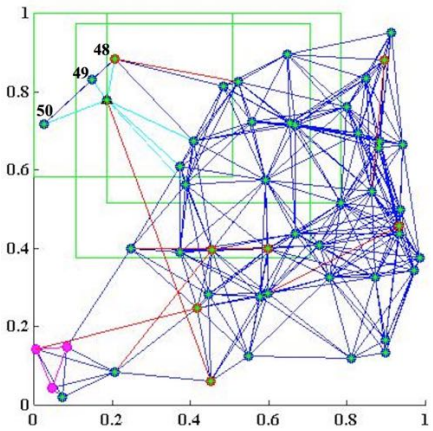
In Fig. 6.11 is shown the results of two iteration of the Safe Navigation Algorithm, over a random graph composed by 50 nodes, deployed in the square $S = [(0, 1) \times (0, 1)]$, having the range of the sensor antenna $\rho = 0.2$. Due to position of the localization seeds, only 5 nodes are localized using SELA: the topology of the graph prevents the propagation of the localization, since the majority of the nodes are far from the seeds, as shown in Fig. 6.11a. To fill the gap, the algorithm considers all the nodes connected to a single localized ones: specifically, node σ_6 is connected to σ_2 and nodes σ_7 and σ_8 to σ_4. All these nodes share the seeds $\Sigma_{6,7,8} = \{\sigma_2, \sigma_4, \sigma_5\}$, subsequently, only a candidate point is retrieved by the algorithm, as shown in Fig. 6.11b. The localization propagates in the networks and most of the nodes can compute their positions. To finally localize node σ_{49} a candidate position is selected considering as pivot the node σ_{48}. The mobile node travels to the candidate position across the network, so to bound the localization



(a) SELA output



(b) SELA output using a mobile node still in the first candidate position



(c) SELA output using mobile node still in the second candidate position

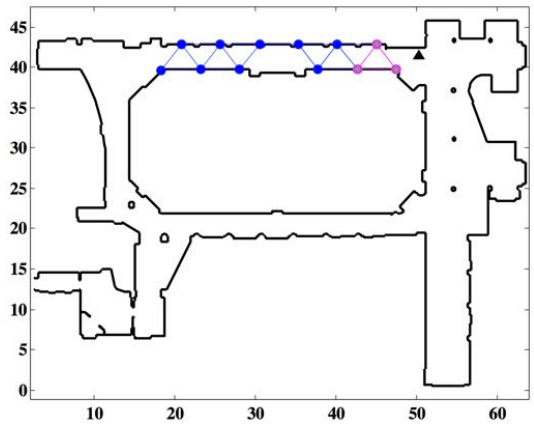
Figure 6.11: SELA output exploiting the rescuer (black triangle connected to the graph with dashed cyan line): in the graph are represented seeds nodes (magenta) in the left down corner, unlocalized nodes (blue circle), localized nodes (green star), SELA nodes (red circle), and shadow edges (red line).

error provided by PDR. When the rescuer reaches the candidate position, the SELA algorithm is able to localize nodes σ_{49} and σ_{50}, as shown in Fig. 6.11c.

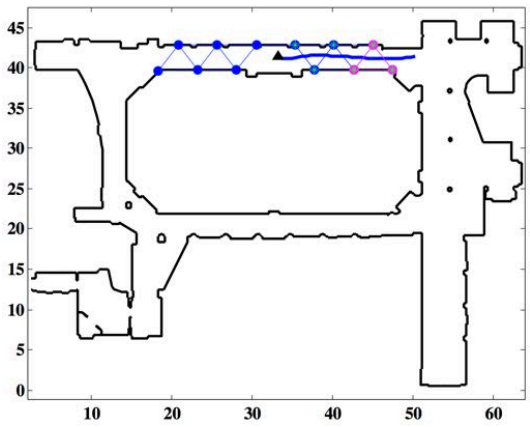
In Fig. 6.12 is shown the results of a real application of the Safe Navigation Algorithm inside a building. In particular during this experiment a sensor network composed by 12 nodes is deployed as reported in Fig. 6.12a. The range of the sensors is about 5 meters. Inside the building a rescuer walk along a path of about 40 meters. Due to position of the localization seeds, only 6 nodes are localized using SELA as shown in Fig. 6.12b. To this end in Fig. 6.12c is find the candidate position of the additional node to localize the entire network. The additional node, according to the path of the rescuer, is localized as reported in Fig. 6.12d. When the rescuer reaches the candidate position, the SELA algorithm is able to localize nodes as shown in Fig. 6.12e.

The graph is composed by ten nodes randomly deployed in the environment. The link between a pair of nodes represents the presence of communication. The number of the nodes of the network is known. The robot starts to explore the environment and find the first node and localizes it using it own position. To find the other nodes two strategies are considered. As first step it is selected the node connected to the localized one with the maximum degree, if more nodes have the same degree the ones at the minimum distance is selected. In this case the first node find by the robot is the node labeled as 3. This node have three neighbors with the same degree, according to the research strategies described above the robot in this case travels on the one at the minimum distance, node 4 and localize it. At the same way from the node 4 the robot travels through the node 5.

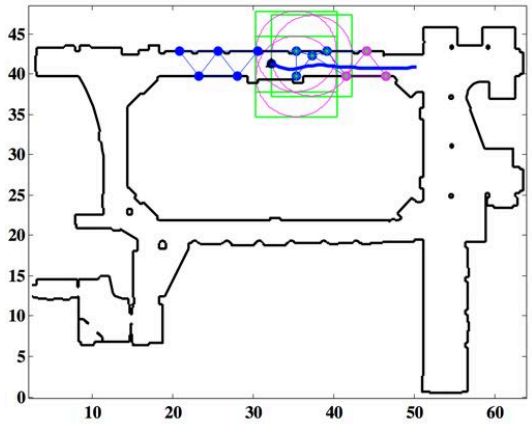
After these operations a set of three nodes $\{3, 4, 5\}$ (see Fig. 6.13a) is produced and trilateration and shadow edge can be applied to localize the network (see Fig. 6.13b). In this case, due to the topology of the network, some nodes are not localized at the end of this iteration, so the algorithm starts again and the robot looking for other nodes. The mobile agent travels to the unlocalized node connected to a localized one, in this exploration it finds the node 6 (see Fig. 6.13c) and using the choosing node strategy it finds also the node 8 (see Fig. 6.13d) and the node 7x. Now, it is possible to apply again trilateration and, this time, at the end of this strategy the network is completely localized as shown in Fig. 6.13f.



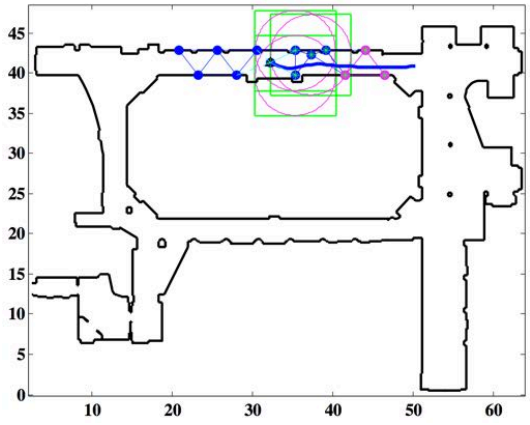
(a) Sensors Network deployed in the environment



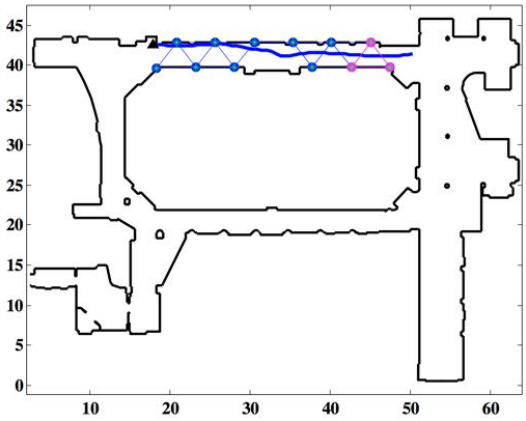
(b) Sensors Network localization



(c) SELA output using a mobile agent in the first candidate position

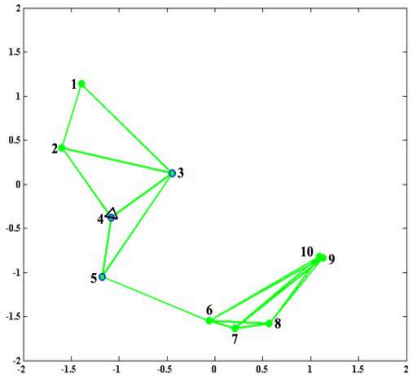


(d) Localization of the candidate

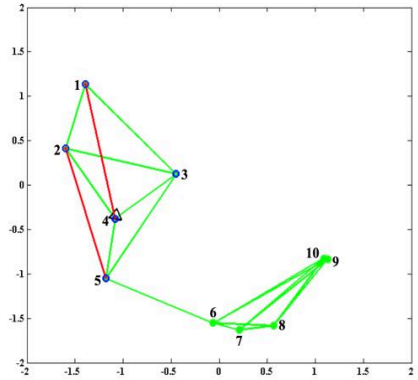


(e) Sensors Network localization

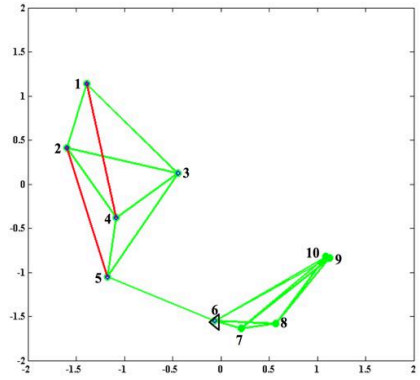
Figure 6.12: A real application of SELA output exploiting the rescuer (black triangle connected to the graph with dashed cyan line): in the graph are represented seeds nodes (magenta) in the left down corner, unlocalized nodes (blue circle), localized nodes (green star).



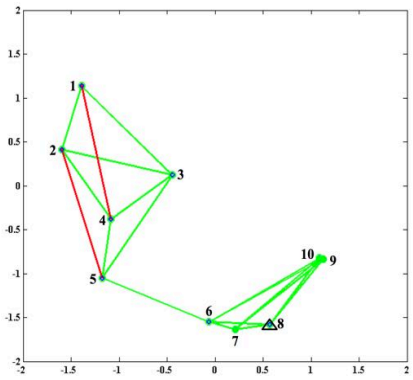
(a) Mobile Agent output



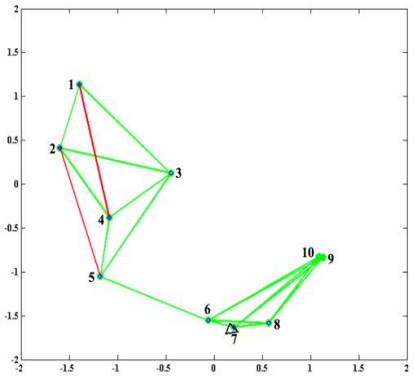
(b) Trilateration and shadow edge output



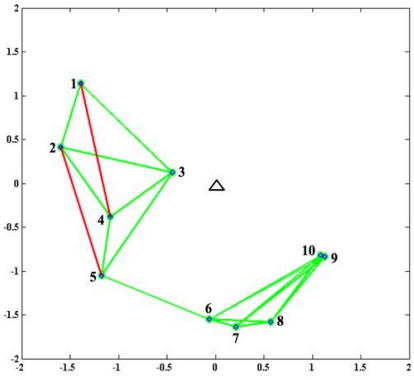
(c) Mobile Agent output in which the first unloclized node connected to the localized subgraph is find



(d) Mobile Agent output the second unloclized node is find



(e) Mobile Agent output the third unloclized node is find



(f) Trilateration and shadow edge output

Figure 6.13: The Mobile Agent (robot) explore the environment and find a set of starting nodes, trilateration and shadow edge localize the nodes connected to the nodes in the set: in the graph are represented localized nodes (blue circle), localized nodes by shadow edge (red circle), and shadow edges (red line).

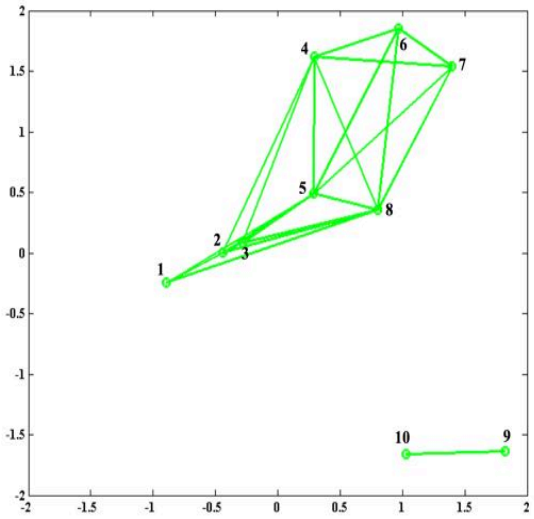


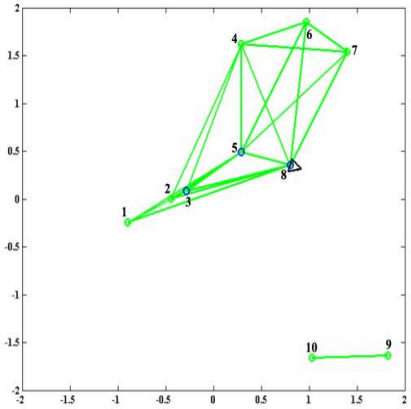
Figure 6.14: A disconnected graph with the nodes randomly deployed in which the green dots represent the nodes and the black triangle the robot.

It is now possible explain how the algorithm works on the disconnected graphs. According to Fig. 6.14 the network is composed by two subgraph. The subgraphs are localized in two steps (see Fig. 6.15).

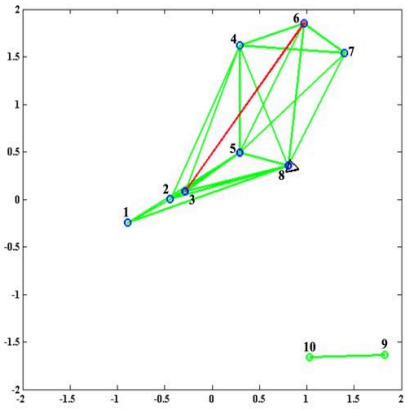
The robot, as for the connected graph, starts exploring the environment and it finds three starting nodes $\{3, 5, 8\}$ (see Fig. 6.15a). After that the trilateration and shadow edge strategies are used and the first part of the graph is localized as shown in Fig. 6.15b.

Now two nodes, labeled as 9 and 10, are still not localized. Two localized them it is necessary to connect them to the localized part of the network. We can do that as described in and create a virtual link, the *bridge link* as shown in Fig. 6.15c. Thanks to this strategy it is possible to obtain a new graph G_{d_r} that is a graph in which the virtual link is included. After that the robot localizes the disconnected node the link falls down and the network is again disconnected (see Fig. 6.15d). Anyway, according to the localization strategy it is possible to complete the localization of the network as shown in Fig. 6.15e)

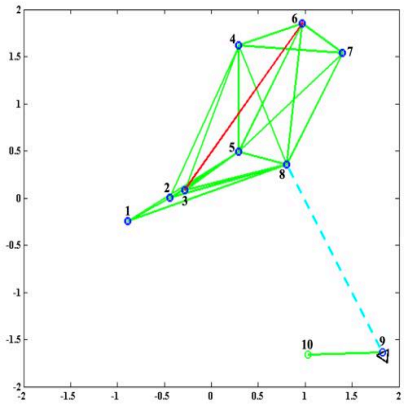
In this section the algorithm applied to a real case is considered. Here it is studied the connected case as shown the graph in Fig. 6.16a. According to the algorithm the



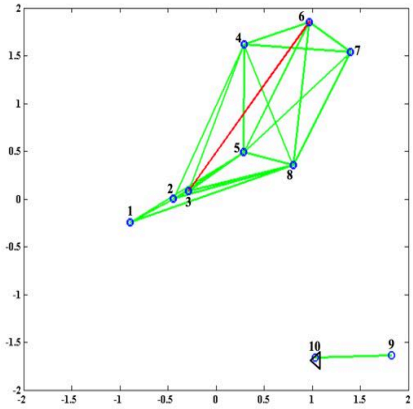
(a) Mobile Agent output



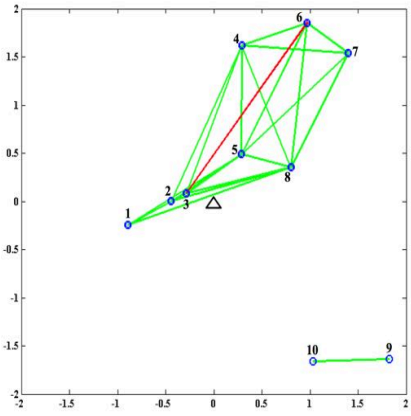
(b) Trilateration and shadow edge output



(c) Mobile Agent and *bridge link* output

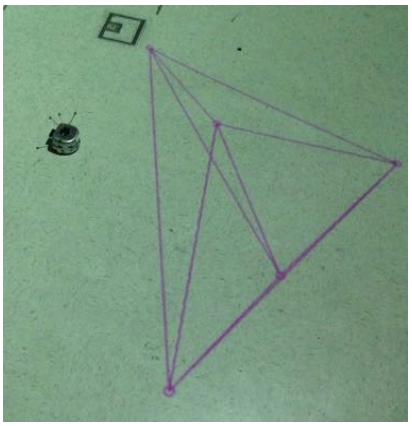


(d) Trilateration and shadow edge output

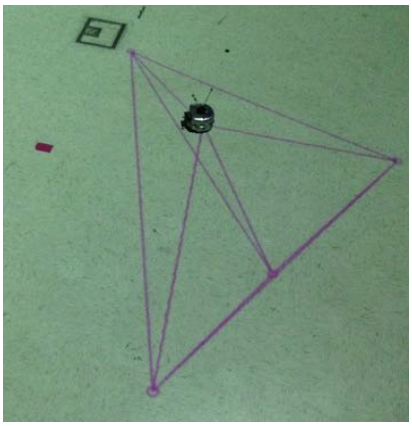


(e) Mobile Agent output

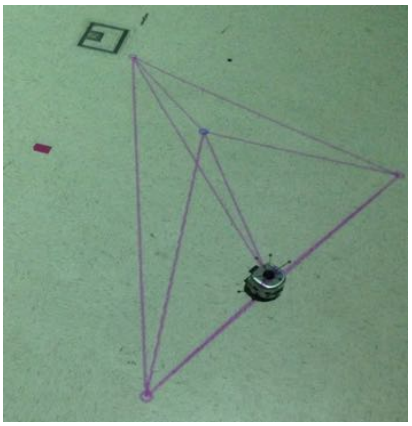
Figure 6.15: The Mobile Agent (robot) explore the environment and find a set of starting nodes, trilateration and shadow edge localize the nodes connected to the nodes in the set: in the graph are represented localized nodes (blue circle), localized nodes by shadow edge (red circle), and shadow edges (red line).



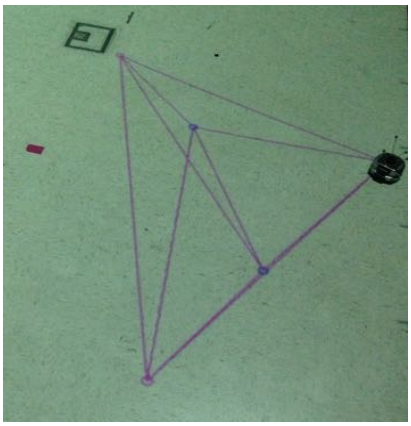
(a) Mobile Agent in starting position



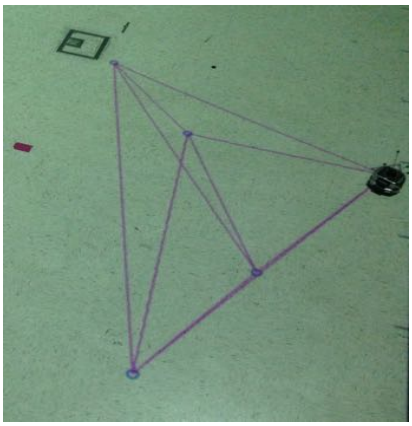
(b) Mobile Agent first node localized



(c) Mobile Agent second node localized

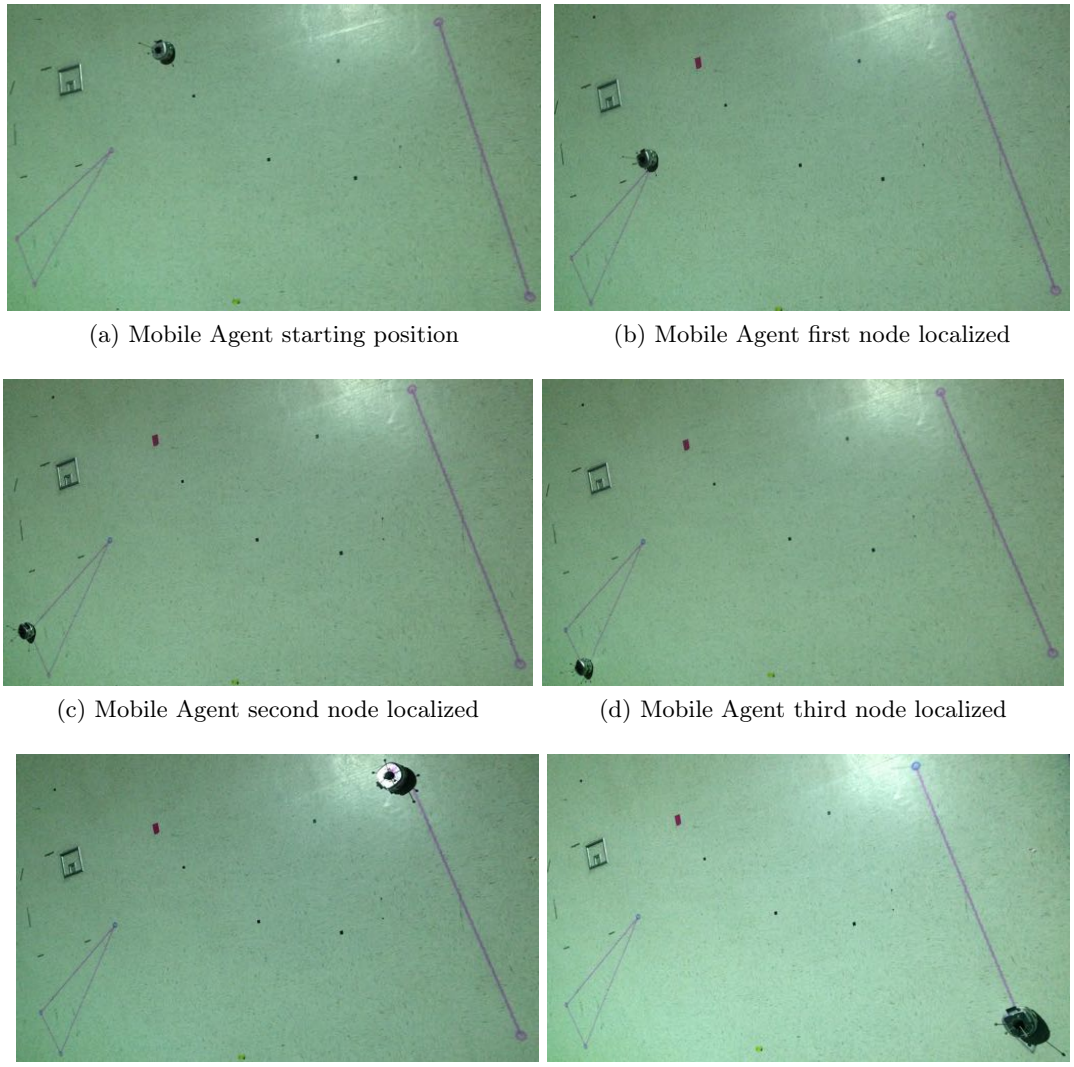


(d) Mobile Agent third node localized



(e) The remaining node is localized using trilateration and the network is completely localized

Figure 6.16: The Mobile Agent (robot) explore the environment and find a set of starting nodes and the remaining node are localized using trilateration: in the graph are represented localized nodes (blue circle), graph (magenta), unlocalized node (magenta circle).



(a) Mobile Agent starting position

(b) Mobile Agent first node localized

(c) Mobile Agent second node localized

(d) Mobile Agent third node localized

(e) Mobile Agent finds the disconnected component

(f) Mobile Agent node output

Figure 6.17: The Mobile Agent (robot) explore the environment and find a set of starting nodes and then the disconnected subgraph: in the graph are represented localized nodes (blue circle), graph (magenta), unlocalized node (magenta circle).

robot starts to explore the environment and it finds the first node (see Fig. 6.16b), and from that one is able to find the second and the third one (see Fig. 6.16c and Fig. 6.16d). Now a set of three localized node it is provide and it is possible to try to localize the remain of the network using trilateration or shadow edge. Here the use of trilateration is enough in fact the remaining nodes to localize had both three localized neighbors. As shown in Fig. 6.16e the network is completely localized.

Now the disconnected case is proposed. According to Fig. 6.17a the graph analyzed is composed by two parts the first one composed by three nodes and the second one composed by two nodes. As shown Fig 6.17 the robot starts to localize first the set composed by three nodes. The process of this operation is detailed in Fig. 6.17b, Fig. 6.17c, and Fig. 6.17d in which the sequence of the node that are discovered is reported.

Once that this part of the network is localized the robot travels to the second part of the graph (see Fig. 6.17e) to find the remaining unlocalized nodes as described in Section 5.4 and the robot localizes the last one node (see Fig. 6.17f).

Chapter 7

Conclusion

Location awareness is an important aspect in disaster response. Although many efforts have been carried out both in academy and industry, to date, there is not an off-the-shelf solution for the localization of first responders during emergency.

To this aim, a novel architecture for localization system that foresees the use of hybrid team, i.e., team composed by human rescuers and robots, has been designed. In more details, it exploits the complementary abilities of human and robot through a common framework: in this way, the localization representation of human can cope with the robotic representation.

The framework, HRTLS, is based on Bayesian filtering: the localization is regarded as the problem of finding the position and the heading of a mobile agent, exploiting data acquired by wearable sensors and a smart environment. To this end, in this thesis the design and implementation of a localization algorithm for human rescuer, based on pedestrian dead reckoning and context-aware sensors (i.e., range free nodes) has been performed. The adopted approach is quite common in robotic literature, however, it is not used in indoor positioning systems, where the localization is mostly obtained by fingerprinting or trilateration. The performed research results in a set of algorithms for activity recognition and heading estimate, that have deeply tested in long lasting experiments in large environments. The results retrieved in a real scenario, by using low cost devices, outperform the performances of state of the art localization systems while achieving (and overtaking) the requirements of the firefighter scenario.

Concerning the smart environment, the HRTLS foresees a cooperation between the mobile agents and the infrastructure pre-deployed in the emergency area. Specifically, different approaches for improving sensor localization and mobile agent tracking are proposed. The thesis focuses on node positioning, addressing this topic in a theoretical perspective: a simple network model is adopted (i.e., the disk graph) and conditions on the localizability of the network are derived exploiting information about the availability of communication link between nodes.

It is useful to underline that the proposed architecture can be applied to several application scenarios. As well known, smartphones are equipped nowadays with many sensors and they are worldwide spread. There in many new services can benefits form the proposed architecture, i.e. smart guides in museum or stores, location based hints, etc.

Beyond the results obtained, there is still room for improvement and there are topics that still need to be properly addressed. In indoor localization for rescuers, more activities need to be considered and different sensors (e.g., Bluetooth Low Energy tag) could be used for improving the localization accuracy. Moreover, more complex filters can be adopted to cope with sensor able to provide only context aware-information.

Concerning the smart infrastructure, there is the need to consider the impact of the noise over the range measurement: here it is considered only in the refinement step, however the localizability of the network can be affected by noisy measurements. Another future direction can be the investigation of different models for the networks that take into account the real radiation pattern of the radio antenna.

Appendix A

The drawbacks of the angle/axis representation can be overcome by a different four-parameter representation; namely, the unit quaternion. Euler parameters, defined as $\mathcal{Q} = \{\eta, \epsilon\}$ where:

$$\eta = cos\frac{\theta}{2} \tag{7.1}$$

$$\epsilon = [\epsilon_x, \epsilon_y, \epsilon_z]^T = sin\frac{\theta}{2}r; \tag{7.2}$$

are called the scalar and the vector part of the quaternion respectively. They are constrained by the condition:

$$\eta^2 + {\epsilon_x}^2 + {\epsilon_y}^2 + {\epsilon_z}^2 = 1, \tag{7.3}$$

hence, the name *unit* quaternion. It is worth remarking that, unlike the angle/axis representation, a rotation by $-\theta$ about r gives the same quaternion as that associated with a rotation by θ about r; this solves the above nonuniqueness problem.

The rotation matrix corresponding to a given quaternion takes on the form:

$$R(\eta, \epsilon) = \begin{bmatrix} 2(\eta^2 + \epsilon_x^2) - 1 & 2(\epsilon_x\epsilon_y - \eta\epsilon_z) & 2(\epsilon_x\epsilon_z - \eta\epsilon_y) \\ 2(\epsilon_x\epsilon_y - \eta\epsilon_z) & 2(\eta^2 + \epsilon_y^2) - 1 & 2(\epsilon_y\epsilon_z - \eta\epsilon_x) \\ 2(\epsilon_x\epsilon_z - \eta\epsilon_y) & 2(\epsilon_y\epsilon_z - \eta\epsilon_x) & 2(\eta^2 + \epsilon_z^2) - 1 \end{bmatrix}. \tag{7.4}$$

If it is desired to solve the *inverse problem* to compute the quaternion corresponding to a given rotation matrix

$$R(\eta,\epsilon) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{7.5}$$

the following result is useful:

$$\eta = \frac{1}{2}\sqrt{r_{11}+r_{22}+r_{33}+1} \tag{7.6}$$

$$\epsilon = \frac{1}{2}\begin{bmatrix} sgn(r_{32}-r_{23})\sqrt{r_{11}-r_{22}-r_{33}+1} \\ sgn(r_{13}-r_{31})\sqrt{r_{22}-r_{33}-r_{11}+1} \\ sgn(r_{21}-r_{12})\sqrt{r_{33}-r_{11}-r_{22}+1} \end{bmatrix}, \tag{7.7}$$

where conventionally $sgn(x) = 1$ for $x \leq 0$ and $sgn(x) = -1$ for $x < 0$.

Notice that in eq. 7.6 it has been implicitly assumed $\eta \leq 0$; this corresponds to an angle $\theta \in [-\pi, \pi]$, and thus any rotation can be described. The quaternion extracted from $R^{-1} = R^T$ is denoted as $\mathcal{Q} - 1$, and can be computed as

$$\mathcal{Q}^1 = \{\eta, -\epsilon\}. \tag{7.8}$$

Let $\mathcal{Q}_1 = \{\eta_1, \epsilon_1\}$ and $\mathcal{Q}_2 = \{\eta_2, \epsilon_2\}$ denote the quaternions corresponding to the rotation matrices R_1 and R_2, respectively. The quaternion corresponding to the product $R_1 R_2$ is given by

$$\mathcal{Q}_1 * \mathcal{Q}_2 = \{\eta_1\eta_2 - {\epsilon_1}^T\epsilon_2, \eta_1\epsilon_2 + \eta_2\epsilon_1 + \epsilon_1 \times \epsilon_2\} \tag{7.9}$$

where the quaternion product operator "$*$" has been formally introduced. It is easy to see that if $\mathcal{Q}_2 = {\mathcal{Q}_1}^T - 1$ then the quaternion $\{1, 0_{3\times 1}\}$ is obtained from 7.6 which is the identity element for the product.

Appendix B

Inertial Platform Calibration

For the calibration of the accelerometer and gyroscope was used the standard *IEEEstd517-1974(R2010)* (the *six-position static and rate tests* [2]).

Accelerometr Calibration

The six-position static test method (used for the accelerometer calibration) requires the inertial system to be mounted on a leveled table with each sensitive axis pointing alternately up and down. For a triad of orthogonal sensors this results in a total of six positions. The bias (b_{acc}) and scale (S_{acc}) factors can then be calculated using the following equations:

$$b_{acc} = \frac{l_{up} + l_{down}}{2} \tag{7.10}$$

$$S_{acc} = \frac{l_{up} + l_{down} - 2K}{2K} \tag{7.11}$$

where l_{up} is the mean value of sensor measurement when the sensitive axis is pointed upward, l_{down} is the measurement when the sensitive axis is pointed downwards and the value K is the known reference signal. For accelerometers, K is the local gravity constant and for gyroscopes K is the magnitude of the earth rotation rate at the given latitude. However, the earth rotation rate can only be used for navigation and tactical grade gyroscopes, since low grade gyroscopes such as MEMS suffer from bias instability and noise levels that can completely mask the earth s reference signal. The six-position calibration accuracy depends on how well the axes are aligned with the vertical axes

of the local level frame. For accurate results a perfect cube shaped mounting frame is required. This standard calibration method can be used to determine the bias and scale factors of the sensors, but cannot estimate the axes misalignments (non-orthogonalities). To estimate the non-orthogonalities an improved six-position test can be performed which takes into account all three types of errors.

Gyroscope Calibration

For the Gyroscope calibration the Angle Rate Test is used. Rate tests are typically done using a precision rate turntable. By rotating the unit through precisely known angles the reference turning rates can be developed. By comparing the outputs of the IMU to these references, the biases, scale factors and non-orthogonalities of the gyroscopes can be estimated using the same principle as the six-position method. This is typically accomplished by rotating the table through a defined angle in both the clockwise and counter clockwise directions as given by equation:

$$S_{gyro} = \frac{\omega_{clockwise} + \omega_{anticlockwise}}{2\omega}, \tag{7.12}$$

where $\omega_{clockwise}$ represents gyroscope output on rotating the turntable by ω in the clockwise direction while $\omega_{anticlockwise}$ represents gyroscope output on rotating the turntable by ω in the $\omega_{anticlockwise}$ direction and S_{gyro} gyroscope scale factor error. These methods provide an estimate of the deterministic errors only and hence stochastic error models are required to determine random errors.

Thermometer Calibration

Even if in this work I don't use the temperature sensor I have calibrated it with the Methods of Thermal Testing. The purpose of thermal testing is to determine the variation of the basic sensor parameters when operated under different temperatures. There are two main approaches for thermal testing:

1. Allow the IMU enclosed in the thermal chamber to stabilize at a particular temperature corresponding to the temperature of the thermal chamber and then record the data. This method of recording the data at specific temperature points is called the Soak method.

2. In the second method, called the Thermal Ramp method, the IMU temperature is linearly increased or decreased for a certain period of time (Titterton and Weston, 1997).

In this work is used the Soak method in which thermal calibration was conducted to calculate the biases and scale factors variations of the sensors over a range of temperature from $10°C$ to $40°C$. The intermediate value is given by the equation:

$$p(t) = \frac{t1 - t}{t_1 - t_0} p_0 + \frac{t - t_0}{t_1 - t_0} p_1, \tag{7.13}$$

where t is the required temperature point and $p(t)$ is the calculated value at that point.

Magnetometer Calibration

To calibrate the magnetometer what has been done is: holding the smart IMU has been played an eight as the same way that the figure Fig. 1.

After creating the semi-sphere representing the magnetic field is searched to the sphere that best approximates with the best distance between the approximate and hopefully all samples collected during the movement.

In the figure Fig. 2 is reported the sphere that represented the magnetic field of the magnetometer.

Figure 1: Movement as Möbius strip for the magnetometer calibration.

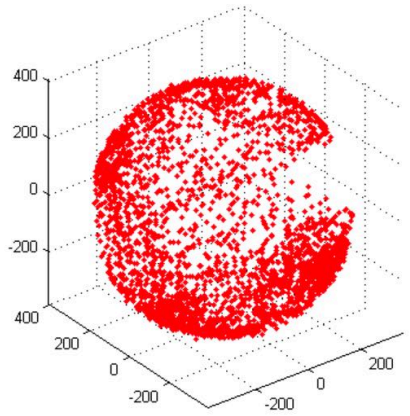


Figure 2: Magnetic field of the magnetometer sphere.

RFID Calibration

The use of the RFID system for ILP tasks is challenging since it only provides information about the presence of tags in the environment. Moreover, most of the current RFID systems are developed for logistics applications, where high power antennas are used to detect tags equipped with small antennas. In the REFIRE framework, the opposite approach is foreseen, due to design constraints. The reader antenna is carried by the rescuer: it is small and low power to extend the battery life. The tags are embedded in the environment and their size has been increased to host a big antenna. Specifically, the reader is the qID R1240I, a mobile UHF RFID reader composed by two directional antennas (horizontal and vertical polarization) able to implement a semi-circular radiation lobe. The tag is the rugged Omni-ID Ultra passive equipped with a large reflective antenna.

To better understand the feasibility and the performance of the approach, a large number of trials have been carried out to outline the behavior of the system under real operating scenarios since the tags detection performances depends on environmental conditions (i.e., electromagnetic reflections, humidity, temperature, etc.). Here, only few are reported for the sake of space, however, a complete assessment for the adopted system has been obtained.

The following parameters are used to define a specific operating scenario, as shown in Fig 3:

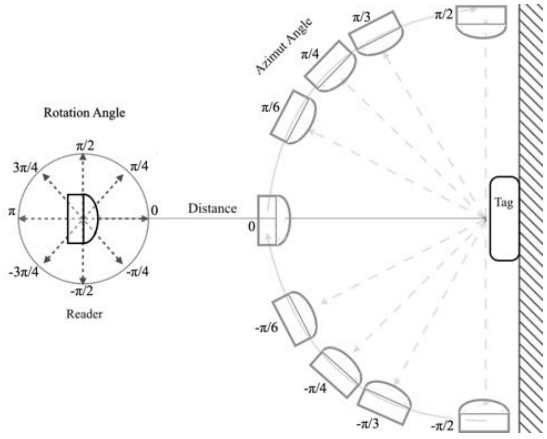


Figure 3: Parameters for static and dynamic tests.

d It represents the distance between the tag and the reader. It is set in the range $[0 \div 350]\, cm$;

ϑ It represents the azimuth between the two devices. When tag and reader are faced $\vartheta = 0$, the following azimuth angles are considered $\vartheta \in \left\{0, \pm\frac{\pi}{6}, \pm\frac{\pi}{4}, \pm\frac{\pi}{3}, \pm\frac{\pi}{2}\right\} rad$;

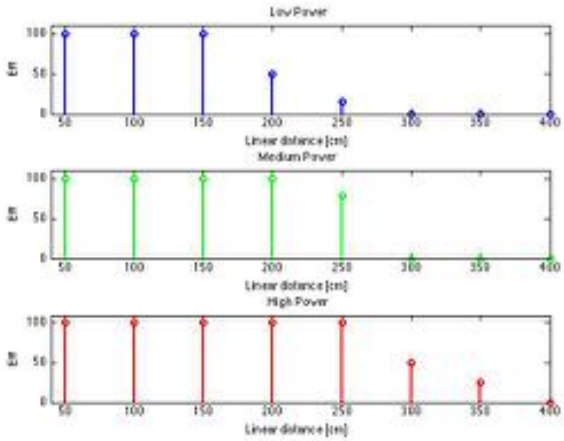
φ It represents the rotation angle of the reader with respect to its vertical axis. Also for these parameters, the following value are considered $\varphi \in \left\{0, \pm\frac{\pi}{4}, \pm\frac{\pi}{2}, \pm\frac{3\pi}{4}, \pm\pi\right\} rad$;

h_T It represents the height of the RFID tag; three different heights are considered, so $h_T \in \{149, 187, 220\}\ cm$, according to the heights of the emergency signs;

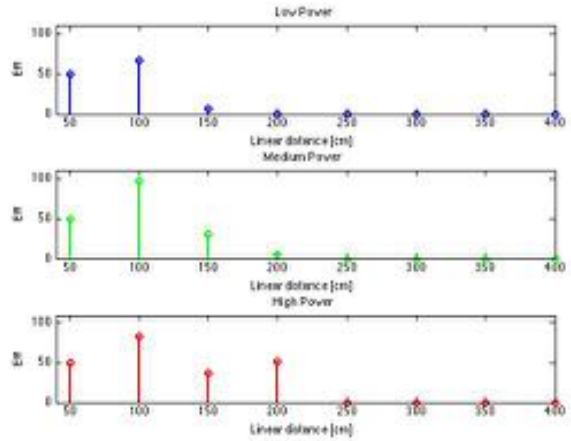
h_R It represents the height of the RFID reader; several heights are considered according within the users;

e It represents the emission power of the reader. It can be set via the reader software with respect to the maximum power $12.5\, mW$ to high power (100%), medium power (66%) and low power (33%).

The tests performed can be divided into two classes: *static* and *dynamic* tests. In these tests different indexes are used to evaluate the efficiency of the system, as detailed below.



(a) RFID system efficiency for $h_T = 149\,cm$ and different values of e.



(b) RFID system efficiency for $h_T = 220\,cm$.

Figure 4: RFID efficiency vs distance d using different power level e (blue, green, and red are low, medium, and high power respectively).

Static Tests

The static tests aim at setting the performance of the system into indoor environments, considering the reader stationary. The main outcome of these tests is the definition of the radiation lobe of a tag in real operating conditions: tests, indeed, were performed in office like environments, where reflections warp the nominal radiation lobe.

During trials, the reader was fixed on a mobile platform and located in different position near a tag: tests performed in similar operating conditions have been averaged. In all the trials, a mean shoulder height is considered, so $h_R = 140\,cm$.

The efficiency η is defined as the percentage of the echoes over the queries sent:

$$\eta = \frac{n_e}{n_q} 100 \tag{7.14}$$

where n_e is the number of echoes detected and n_q the total number of queries.

The first set of trials aims at evaluating the performance of the system, using different power levels. The results of these trials are reported: they are obtained by locating the reader in front of the tag (i.e., $\vartheta = 0$ and $\varphi = 0$) and moving forward and backward the mobile platform, so to change the parameter d. Two different heights for the tag have been considered (i.e., $h_T \in \{149, 220\}\,cm$). The outcome of the trials are reported in Fig. 4: in both cases the same efficiency is obtained considering high and medium power levels, while the performance are downgraded using low power level. To limit the power consumption, the power level needs to be set to medium power. It is worth also noticing that the higher the difference $|h_T - h_R|$, the lower the efficiency.

The second set of trials aims at defining the efficiency of the RFID system, considering different rotation angle φ. Tab. 1 presents the results having set $e = 66\%$, $h_T = 149\,cm$, $\vartheta = 0\,rad$, and $d = 150\,cm$.

Looking at Tab. 1, the first column ($\varphi = 0\ rad$) reports the same results of the test in Fig. 4a, since the devices are perfectly faced; the efficiency quickly decreases in the same way when the reader is rotated clockwise or counter-clockwise. Some outliers are obtained when the reader is completely rotated ($\varphi = \frac{3\pi}{4}$ and $\varphi = \pi$) due to reflections.

Finally, several tests have been carried out to characterize the efficiency behavior when the value of the azimuth angle ϑ changes. In this trial the reader height is set to $h_R = 140\,cm$, the tag height to $h_T = 149\,cm$, and the power to $e = 66\%$. A result of these tests is reported in Fig. 5: the tag has a fixed location (i.e., the origin of the reference frame) and orientation, while the reader moves in the surroundings, changing the distance $d \in [0 \div 350]\,cm$.

The performed tests pointed out that in static conditions, the maximum range of the radiation lobe of the RFID system can be assumed to $r = 250\,cm$ with a maximum

Table 1: Efficiency values [%]: distance [cm] vs rotation angle [rad]

φ	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π	$-\frac{3\pi}{4}$	$-\pi$	$-\frac{\pi}{4}$
50	100	100	0	100	75	0	0	100
100	100	100	0	100	30	0	0	100
150	100	75	0	50	50	0	0	20
200	100	0	0	30	0	0	0	0
250	80	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0

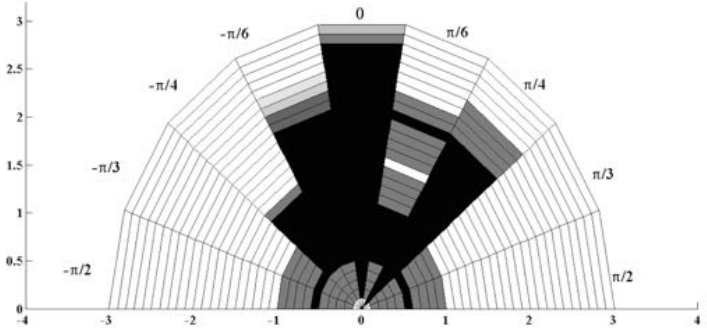


Figure 5: Percentage of successful readings.

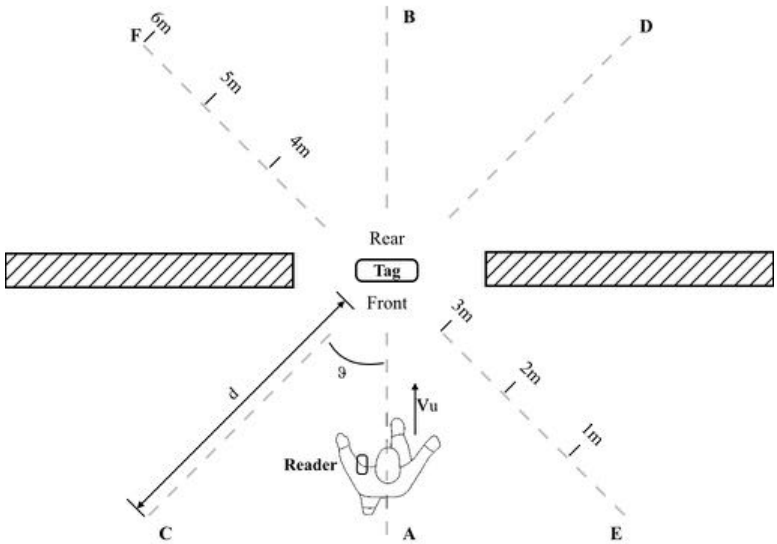


Figure 6: Door crossing tests.

angle of $\alpha = \pm\frac{\pi}{3}$: the direction of the lobe strongly depends on the orientation of the tag.

Table 2: Tag detections in door crossing tests

(a) $e = 66\%$ $v_u = 1\ m/s$

Path	**1** m	**2** m	**3** m	**TOT**
AB	0	0	1	1
CD	0	1	2	3
EF	0	1	2	3
BA	0	0	1	1
DC	0	0	2	2
FE	0	0	0	0

(b) $e = 66\%$ $v_u = 2\ m/s$

Path	**1** m	**2** m	**3** m	**TOT**
AB	0	0	1	1
CD	0	0	2	2
EF	0	0	1	1
BA	0	0	0	0
DC	0	0	0	0
FE	0	0	0	0

Dynamic Tests

The aim of the dynamic tests consists in evaluating the performance of the RFID system during the user's motion. To this end, different scenarios are considered: in the first set of trials, the door crossing test is considered, in the second and third the user walking in an office-like environment is addressed.

In all the trials, the power is set to $e = 66\%$, while the remaining parameters change due to the walking dynamics; also the user speed v_u is considered.

In this tests the efficiency η is measured as:

$$\eta = n_e \tag{7.15}$$

since the reader is not always in a radiation lobe of a tag during motion, so there is no need to consider the number of queries.

In the first trial, the reader is placed on the left shoulder of a user walking across a door along three different paths, as depicted in Fig. 6. The results of these tests are averaged and approximated to the lower integer over 10 path executions. They are reported in Tab. 2: the number of detections during the forward walking is collected,

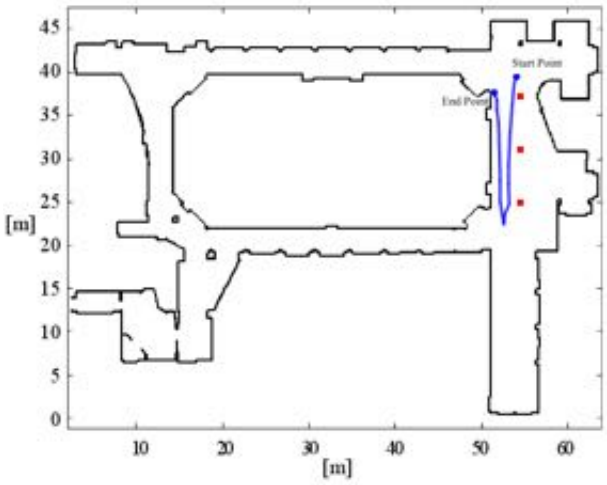


Figure 7: Corridor test: user path (blue solid line) and tags (red squares).

Table 3: Tag detections in corridor test

FORWARD			BACKWARD		
TAG n.1	**TAG n.2**	**TAG n.3**	**TAG n.3**	**TAG n.2**	**TAG n.1**
2	9	0	3	9	0

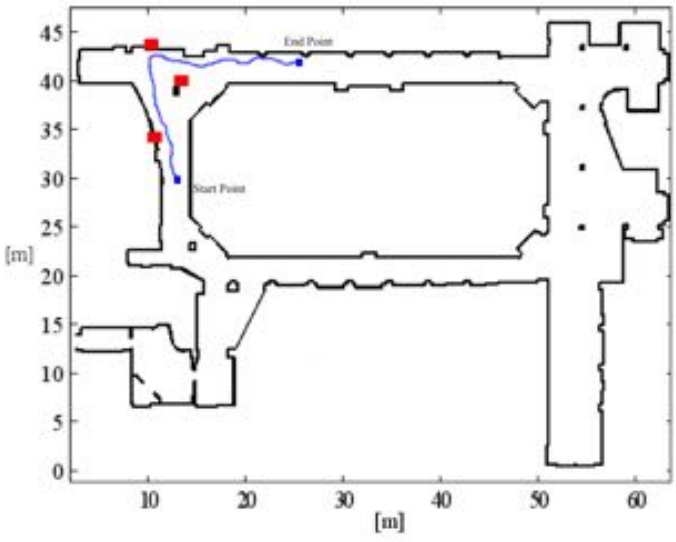


Figure 8: Open space test: user path (blue solid line) and tags (red squares).

Table 4: Tag detections in the open space test

TAG n.1	**TAG n.2**	**TAG n.3**
11	0	5

considering two different speed, $v_u = 1\,m/s$ and $v_u = 2\,m/s$. As it is clear from the results, a larger number of detections is retrieved walking at lower speed, since under this operating condition the reader is in the tag radiation lobe for a longer time, however some detections are available also at higher speed. It is worth noticing that the results obtained during the crossing door trials are compliant with the static ones.

In the second trials, the user walks along a corridor, forward and backward, as shown in Fig. 7: the overall length of the path is $l = 34\,m$, the user speed is about to $v_u = 1\,m/s$. The reader is placed on the left shoulder of the user, considering $h_R = 140\,cm$. The three tags used in the experiment are located on the columns, their height is $h_T = 220\,cm$ and their orientation with respect to the vertical z-axis of the navigation reference frame is $\varphi_T = \frac{\pi}{16}$, $\varphi_T = \frac{\pi}{12}$, and $\varphi_T = \frac{\pi}{6}$ for the first, second and third tag, respectively.

In Tab. 3, the detections of the reader during the test are reported. It is worth noticing that the detections number of the second tag is greater with respect to the others, due to the presence of a metal door reflecting the electromagnetic signal.

In the third dynamic test a forward path in an open space is performed as shown in Fig. 8. The user walks along a $26\,m$ long path at a constant speed $v_u = 1\,m/s$, and wears the RFID reader on the left shoulder. Three tags are considered: they have the same orientation with respect the z-axis of the navigation reference frame $\varphi_T = 0$, however they are located at different height, $h_T = 149\,cm$, $h_T = 187\,cm$, and $h_T = 220\,cm$ for the first, second and third tag, respectively.

The number of echoes retrieved during the experiment from each tag is reported in Tab. 4. The second tag is not detected since the radiation lobe of the RFID reader does not intersect the radiation lobe of the tag itself.

Appendix C

Background service

A Background Service is a component that allows to execute operations in background even if the application that states them is not in action. In other cases another component of the application can starts the Background Service and it can continue to work in background even if another application is used. It is also possible that another component can interact whit it. Usually are useful in background operation as: file download, music execution, and geo-localization.

The Background Service doesn't have a graphic interface which the user can interact but at the same time they should not be confused with processes or threads, intact a Background Service is mange in a different way and has a particular life cycle. Moreover they can use the interfaces provided by Android integrating them self with the system.

The Background Service is useful every time that we want to do something in background without occupy the screen.

It is important now to stress what a background service is not: it is not a separated process. If it is not specified it is executed in the same process in which is integrated, As previously said is not a thread but that does not mean that it works outside the main thread, in fact it is executed in the main thread of the host process, A SB provides two main characteristics:

- A structure for the application to tell the system what you want to do in the background (even when the user does not interact directly with the application);
- A structure for the application to expose some functionality to other applications.

In Fig.9 the life cycle of a Background Service is sketched.

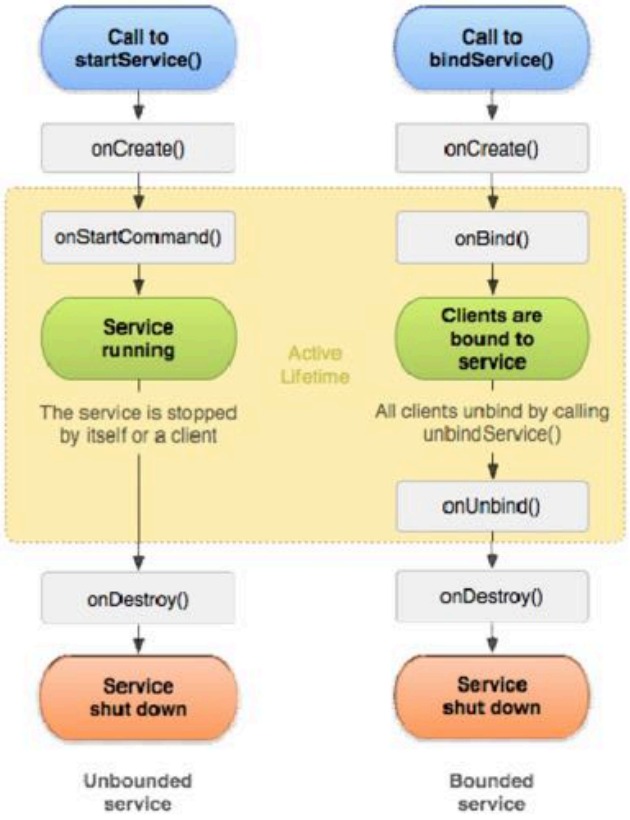


Figure 9: Life cycle of a Background Service

Activity

In each application, it is possible to find one or more fundamental elements, but it is not sure that it has it all. However in most cases, the applications include al least an Activity. According with the Android documentation an Activity is defined as: "a single and precise thing that the user can do". Obviously that assumes that the user to do something has to interact with the device.

Exists a similarity between the concept of window, in a desktop system and the concept of Activity in Android. We can say that the Activity is the component in an application that uses the display and interacts with the user, Usually the activity are presented to the used as full screen window but they can be used also ad mobile window incorporated one in another Activity. The Activity of Android have exclusivity. That means that is possible to execute more Activity concurrently, but only one can occupies

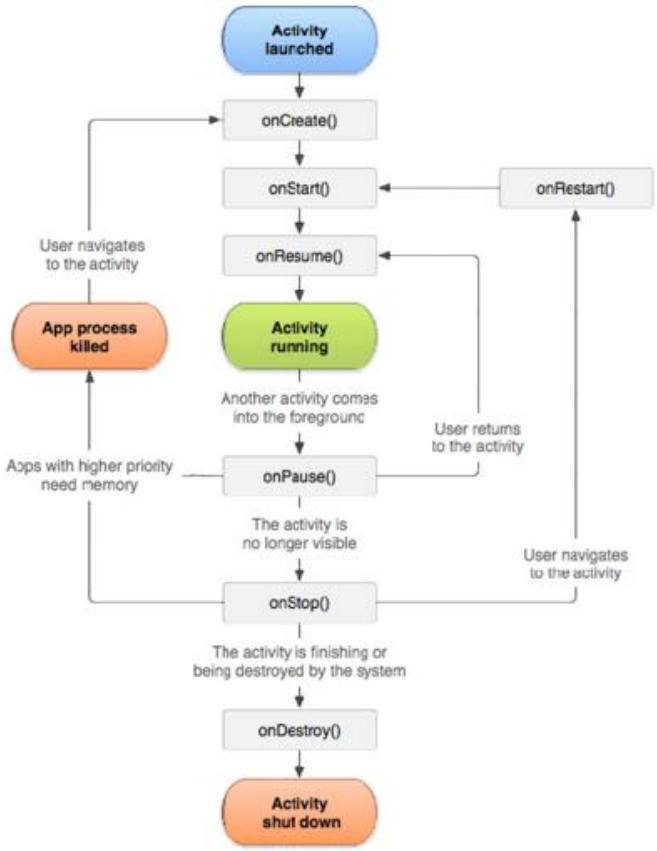


Figure 10: Life cycle of an Activity

the display.

The Activity can be close spontaneously if they have finished their tasks. The cases in which an Activity can arrested can be two:

- the Activity is hibernated and the system arbitrarily decide it is no longer useful and therefore destroys it.
- the system is out of memory, he needs to reclaim space and kills Activity in the background.

The Fig.10 illustrates the sequence of calls to methods Activity performed during the changes in status of the Activity.

The rectangles represent the callback methods you can implement to perform operations when the Activity you move between states, while the ovals represent the states in which an Activity can be.

Sensor Log

As stressed in the previously sections, the use of the BS allows the user to sent in background the application with out compromise the operation, consequently the information will be still collected.

These data are taken by the inertial sensors (accelerometer, gyroscope, and magnetometer) and visualize on a view available on the main screen. For each sensors the information in the x, y, z coordinates is visualized each 10 milliseconds and will be automatically saved on the devices. Using specific classes for the management of the sensors it is possible to obtain the information about the data. The type of sensors is defined by some static constant as:

- TYPE ACCELEROMETER
- TYPE GYROSCOPE
- TYPE ORIENTATION
- TYPE MAGNETIC FIELD

Now, it is necessary to access at the data detected by the sensors. The approach is asynchrony, that means that using a listener and according with the delay the values detected by the system are notified. To this end a listener that is invoked each time that the value of the sensors change is implemented. the frequency in which the data are updated is 10 ms.

The application implemented in Android use as input the data processed by the inertial sensors and gives as output a text file, as a consequence there text files are created, one for each sensors invoked.

At the beginning each sample was visualize with relative data and time, but if on one side this strategies helps to the legibility of the data by a new user, on the other

side the complexity of the next procedure of data process increase. So a new label for the data was used:

$$TIMESTAMP\ X\ Y\ Z \tag{7.16}$$

Bibliography

[1] M. H. Afzal, V. Renaudin, and G. Lachapelle. Assessment of indoor magnetic field anomalies using multiple magnetometers. *in ION GNSS*, 2010.

[2] M. H. Afzal, V. Renaudin, and G. Lachapelle. Magnetic field based heading estimation for pedestrian navigation environments. *Test*, 2011.

[3] M. Aicardi, G. Cannata, G. Casalino, and Indiveri. On the stabilization of the unicycle model projecting a holonomic solution. *Proc. of 8th Int. Symp. on Robotics with Application, Maui, HI*, 2000.

[4] D. Ampeliotis and K. Berberidis. Energy-based model-independent source localization in wireless sensor networks. *in Proceedings of the 16th European Signal Processing Conference, Lausanne, Switzerland*, 2008.

[5] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, B. D. Anderson Y. R. Yang, and P. N. Belhumeur. A theory of network localization. *Mobile Computing, IEEE Transactions*, 5(12):1663–1678, 2006.

[6] A. Bel, J. L. Vicario, and G.Seco-Granados. Node selection for cooperative localization: efficient energy vs. accuracy trade-off. *in Proceedings of the 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC '10), Modena, Italy*, 2010.

[7] A. Bel, J. L. Vicario, and G.Seco-Granados. Real-time path loss and node selection for cooperative localization in wireless sensor networks. *in Proceedings of the 21st IEEE International Symposium on Personal Indoor and Mobile Radio Communications Workshops (PIMRC '10)*, pages 283–288, 2010.

[8] J. E. Bertram and A. Ruina. Multiple walking speed-frequency relations are predicted by constrained optimization. *J. theoretical biology*, 209(4):445–53, 2001.

[9] D. Blatt and A. O. Hero. Energy-based sensor network source localization via projection onto convex sets. *IEEE Transactions on Signal Processing*, 54(9):3614–3619, 2006.

[10] P. Bragatto, F. De Cillis, F. Inderst, F. Pascucci, R. Setola, and M. Tesei. Improving the safety and the operational efficiency of emergency operators via on-field situational awareness. *IEEChemical Engineering Transactions*, 57(2):331–336, 2016.

[11] R. G. Brown and P. Y. C. Hwang. Introduction to random signals and applied kalman filtering. *Wiley*, 1996.

[12] J. Calusdian, X. Yun, and E. Bachmann. Adaptive-gain complementary filter of inertial and magnetic data for orientation estimation. *Robotics and Automation (ICRA), 2011 IEEE International Conference*, pages 1916–1922, 2011.

[13] N. Castaneda and S. Lamy-Perbal. An improved shoe-mounted inertial navigation system. *in 2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–6, 2010.

[14] F. Cavallo, A. Sabatini, and V. Genovese. A step toward gps/ins personal navigation systems: real-time assessment of gait by foot inertial sensing. *in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1187–1191, 2005.

[15] L. Cheng, C. Wu, Y. Zhang, H. Wu, M. Li, and C. Maple. A survey of localization in wireless sensor network. *Hindawi Publishing Corporation International Journal of Distributed Sensor Networks*, 2012.

[16] L. Cheng, C. D. Wu, Y. Z. Zhang, and Y. Wang. Indoor robot localization based on wireless sensor networks. *IEEE Transactions on Consumer Electronics*, 57(3):1099–1104, 2011.

[17] C. T. Chiang, P. H. Tseng, and K. T. Feng. Hybrid unified kalman tracking algorithms for heterogeneous wireless localization systems. *IEEE Transactions on Vehicular Technology*, 61(2):702–715, 2012.

[18] F. De Cillis, L. Faramondi, F. Inderst, S. Marsella, M. Marzoli, F. Pascucci, and R. Setola. Hybrid indoor position system for first responders. *Submitted to IEEE Systems, Man, and Cybernetics*.

[19] F. De Cillis, J. Hernantes, J. M. Sarriegi, F. Pascucci, F. Inderst, M. Tesei, and R. Setola. Rising: Radio frequency identification and inertial navigation for indoor localization in emergency management and building maintenance. *The International Emergency Management Society 2015 Annual Conference*, 2015.

[20] F. De Cillis, F. De Simio, L. Faramondi, F. Inderst, F. Pascucci, and R. Setola. Improving situational awareness for first responders. *9th International Conference on Critical Information Infrastructures Security*, 2014.

[21] R. Connelly. Generic global rigidity. *Discrete & Computational Geometry*, 33(4):549–563, 2005.

[22] J. A. Costa, N. Patwari, and A. O. Hero III. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Trans. Sensor Networks*, 2(1):39–64, 2006.

[23] B. d'Andrèa Novel, G. Champion, and G. Bastin. Control of nonholonomic wheeled mobile robots by state feedback linearization. *Int. J. of Int. J. of Robotic Reasearch*, 14:543–559, 1995.

[24] K. Das and H. Wymeersch. Censored cooperative positioning for dense wireless networks. *in Proceedings of the 21st IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC '10), Istanbul, Turkey*, pages 262–266, 2010.

[25] B. Denis, M. Maman, and L. Ouvry. On the scheduling of ranging and distributed positioning updates in cooperative ir-uwb networks. *in Proceedings of IEEE Inter-*

national Conference on Ultra-Wideband (ICUWB '09), Vancouver, Canada, pages 370–375, 2009.

[26] M. Dippold. Personal dead reckoning with accelerometers. *in Third International Forum on Applied Wearable Computing IFAWC2006*, 2006.

[27] S. Dong, P. Agrawal, and K. Sivalingam. Localization error evaluation in heterogeneous sensor networks. *in Proceedings of IEEE Global Telecommunications Conference, New Orleans, La, USA*, pages 1–5, 2008.

[28] X. Du, D. Mandala, W. Zhang, C. You, and Y. Xiao. A boundary-node based localization scheme for heterogeneous wireless sensor networks. *in Proceedings of IEEE Military Communications Conference (MILCOM '07), Orlando, Fla, USA*, 2007.

[29] T. Eren, O. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur. Rigidity, computation, and randomization in network localization. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 4:2673–2684, 2004.

[30] M. Erol, L. F. M. Vieira, and M. Gerla. Localization with dive'n'rise (dnr) beacons for underwater acoustic sensor networks. *in Proceedings of the 2nd Workshop on Underwater Networks (WuWNet '07), Montreal, Canada*, 2007.

[31] L. Fang, P. Antsaklis, L. Montestruque, M. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie. Design of a wireless assisted pedestrian dead reckoning system the navmote experience. *IEEE Trans. Instrum. Meas.*, 54(6):2342–2358, 2005.

[32] R. Faragher, C. Sarno, and M. Newman. Opportunistic radio slam for indoor navigation using smartphone sensors. *in Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, 2012.

[33] L. Faramondi, F. Inderst, F. Pascucci, and R. Setola. Indoor positioning system using walking pattern classification. *MED*, 2014.

[34] L. Faramondi, F. Inderst, F. Pascucci, R. Setola, and U. Delprato. An enhanced indoor positioning system for first responders. *IPIN*, 2013.

[35] L. Faramondi, F. Inderst, F. Pascucci, and S.Panzieri. Hybrid map building for personal indoor navigation system. *AIM 2014, IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2014.

[36] R. Feliz, E. Zalama, and J. G. García Bermejo. Pedestrian tracking using inertial sensors. *J. Physical Agents*, 3(1):35–43, 2009.

[37] L. Filardo, F. Inderst, and F. Pascucci. C-hips: an hybrid indoor positioning system solution. *submitted to IEEE Transactions on Instrumentation and Measurement.*

[38] L. Filardo, F. Inderst, and F. Pascucci. C-hips: Context aware hybrid indoor positioning system. *submitted to IEEE Transactions on Industrial Electronics.*

[39] L. Filardo, F. Inderst, and F. Pascucci. C-ips: a smartphone based indoor positioning system. *IPIN*, 2016.

[40] C. Fischer, K. Muthukrishnan, M. Hazas, and H. Gellersen. Ultrasound-aided pedestrian dead reckoning for indoor navigation. *Proc. Int'l Workshop Mobile Entity Localization and Tracking in GPS-Less Environments, ACM Press*, pages 31–36, 2008.

[41] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46, 2005.

[42] R. Fujiwara, K. Mizugaki, T. Nakagawa, D. Maeda, and M. Miyazaki. Toa-tdoa hybrid relative positioning system using uwb-ir, san diego, calif, usa. *in Proceedings of IEEE Radio and Wireless Symposium (RWS '09)*, 2009.

[43] M. R. Gholami, S. Gezici, M. Rydstrom, and E. G. Strom. A distributed positioning algorithm for cooperative active and passive sensors. *in Proceedings of the 21st IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC '10), Instanbul, Turkey*, 2010.

[44] S. Godha, G. Lachapelle, and M. E. Cannon. Integrated gps/ins system for pedestrian navigation in a signal degraded environment. *ION GNSS 2006 Fort Worth TX*, 2006.

[45] D. K. Goldenberg, P. Bihler, and M. Cao et al. Localization in sparse networks using sweeps. *in Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06), Los Angeles, Calif, USA*, 2006.

[46] P. Goyal, V. J. Ribeiro, H. Saran, and A. Kumar. Strap-down pedestrian dead-reckoning system. *in 2011 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7, 2011.

[47] J. Gribben, A. Boukerche, and R. Pazzi. Scheduling for scalable energy-efficient localization in mobile ad hoc networks. *in Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '10), Boston, Mass, USA*, 2010.

[48] Y. Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Commun. Surveys Tutorials*, 11(1):13–32, 2009.

[49] P. Gupta and T. Dallas. Activity recognition using single tri-axial accelerometer worn on waist. *IEEE Healthcare Technology Conference: Translational Engineering in Health and Medicine*, 2012.

[50] Y. E. M. Hamouda and C. Phillips. Metadata based optimal sensor selection for multi-target tracking in wireless sensor networks. *International Journal of Research and Reviews in Computer Science*, 2(1):189–200, 2011.

[51] R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communication Survey & Tutorials*, 15(3), 2013.

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

[53] F. Inderst, G. Oliva, F. Pascucci, and R. Setola. Augmenting rescuer safety using wireless sensor networks. *IPIN*, 2015.

[54] F. Inderst, F. Pascucci, and M. Egerstedt. Localization of sparse wireless sensor networks using mobile anchors. *submitted to Ad Hoc Network*.

[55] F. Inderst, F. Pascucci, and M. Santoni. Indoor positioning system using multi agent correction. *submitted to IEEE /ASME Transactions on Mechatronics*.

[56] F. Inderst, F. Pascucci, and M. Santoni. 3d pedestrian dead reckoning and activity classification using waist-mounted inertial measurement unit. *IPIN*, 2015.

[57] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. *in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, 4:2652–2661, 2004.

[58] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 37–42, 2009.

[59] Y. Jin, H. S. Toh, W. S. Soh, and W. C. Wong. A robust dead-reckoning pedestrian tracking system with low cost sensors. *PerCom*, 2011.

[60] L. M. Kaplan. Global node selection for localization in a distributed sensor network. *IEEE Transactions on Aerospace and Electronic Systems*, 42:113–135, 2006.

[61] L. M. Kaplan. Local node selection for localization in a distributed sensor network. *IEEE Transactions on Aerospace and Electronic Systems*, 42:136–146, 2006.

[62] L. M. Kaplan, Q. Le, and P. Molnar. Maximum likelihood methods for bearings-only target localization. *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, Utah, USA*, 5:3001–3004, 2001.

[63] U. A. Khan, S. Kar, and J. M. F. Moura. Distributed sensor localization in random environments using minimal number of anchor nodes. *IEEE Transactions on Signal Processing*, 57(5):2000–2016, 2009.

[64] J. W. Kim and C. Park. A step , stride and heading determination for the pedestrian navigation system. *Technology*, 3(1):273–279, 2005.

[65] H. Kopka, P. W. Daly, N. Wang, E. Ambikairajah, B.G. Celler, and N.H. Lovell. Accelerometry based classification of gait patterns using empirical mode decomposition. *ICASSP*, 2008.

[66] B. Krach and P. Roberston. Cascaded estimation architecture for integration of foot-mounted inertial sensors. *in Position Location and Navigation Symposium 2008 IEEEION*, 1:112–119, 2008.

[67] Q. Ladetto. On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive kalman filtering. *ION GPS*, 2000:1735–1740, 2000.

[68] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4:331–340, 1970.

[69] C. Laoudias, M. P. Michaelides, and C. G. Panayiotou. Fault detection and mitigation in wlan rss fingerprint-based positioning. *Journal of Location Based Services*, 6(2):101–116, 2012.

[70] T. Levi and R. Judd. Dead reckoning navigational system using accelerometer to measure foot impacts. 1995.

[71] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Trans. Syst. Man Cybern. C. Appl. Rev.*, 37(6):1067–1080.

[72] K. Lu, X. Xiang, D. Zhang, R. Mao, and Y. Feng. Localization algorithm based on maximum a posteriori in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2012.

[73] M. P. Michaelides, C. Laoudias, and C. G. Panayiotou. Fault tolerant localization and tracking of multiple sources in wsns using binary data. *IEEE Transactions on Mobile Computing*, 13((6)):1213–1227, 2014.

[74] T. Moder, P. Hafner, K. Wisiol, and M. Wieser. 3d indoor positioning with pedestrian dead reckoning and activity recognition based on bayes filtering. *PIN*, 2014.

[75] B. G. Celler N. Wang, E. Ambikairajah and N. H. Lovell. Accelerometry based classification of walking patterns using time-frequency analysis. *EMBS*, 2007.

[76] NIOSH. Fatality assessment and control evaluation investigation report # f2007-19. *available at http://www.cdc.gov/niosh/fire/pdfs/face200619.pdf*, June 2007.

[77] NIOSH. Fatality assessment and control evaluation investigation report # f2007-18. *available at http://www.cdc.gov/niosh/fire/pdfs/face200718.pdf*, February 2009.

[78] L. Ojeda and J. Borenstein. Non-gps navigation for security personnel and first responders. *Navigation*, 60(3):391–407, 2007.

[79] J. Park, Y. Kim, and J. Lee. Waist mounted pedestrian dead-reckoning system. *Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference on. IEEE*, 2012.

[80] S. Park. Pedestrian inertial navigation with gait phase detection assisted zero velocity updating. *in 2009 4th International Conference on Autonomous Robots and Agents*, 2001.

[81] X. Qu and L. Xie. Source localization by tdoa with random sensor position errors - part i: static sensors. *in Proceedings of the 15th International Conference on Information Fusion*, 2012.

[82] X. Qu and L. Xie. Source localization by tdoa with random sensor position errors - part ii: mobile sensors. *in Proceedings of the 15th International Conference on Information Fusion, Singapore*, 2012.

[83] M. G. Rabbat, R. D. Nowak, and J. Bucklew. Robust decentralized source localization via averaging. *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), Philadelphia, Pa, USA*, 5:V1057–V1060, 2005.

[84] A. Anshul Rai, K. K. Chintalapudi, P. Venkat, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. *18th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2012.

[85] C. Randell, C. Djiallis, and H. Muller. Personal position measurement using dead reckoning. *Seventh IEEE International Symposium on Wearable Computers 2003 Proceedings*, pages 166–173, 2003.

[86] H. Ren and M. Q. Meng. Power adaptive localization algorithm for wireless sensor networks using particle filter. *IEEE Transactions on Vehicular Technology*, 58(5):2498–2508, 2009.

[87] J. Saarinen, J. Suomela, S. Heikkila, M. Elomaa, and A. Halme. Personal navigation system. *2004 IEEERSJ International Conference on Intelligent Robots and Systems IROS IEEE Cat No04CH37566*, pages 212–217, 2004.

[88] A. M. Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 53(7):1346–1356, 2006.

[89] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly np-hard. *Carnegie-Mellon University, Department of Computer Science*, 1980.

[90] R. Setola, F. Pascucci, F. De Cillis, F. Inderst, U. Delprato, M. Marzoli, S. Marsella, M. Petracca, M. Vari, and M. Cristaldi. Refire a reference implementation for deep indoor localization'. *8th Future Security Conference 2013, Berlin, Germany*, 2013.

[91] Y. Shang and W. Ruml. Improved mds-based localization. *in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, 4:2640–2651, 2004.

[92] C. Shen and D. Pesch. A heuristic relay positioning algorithm for heterogeneous wireless networks. *in Proceedings of the 69th IEEE Vehicular Technology Conference, Barcelona, Spain*, 2009.

[93] K. F. Ssu, C. H. Ou, and H. C. Jiau. Localization with mobile anchor points in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 54(3):1187–1197, 2005.

[94] R. Stirling, J. Collin, K. Fyfe, and G. Lachapelle. An innovative shoe-mounted pedestrian navigation system. *in GNSS 2003 (Graz, Austria, 2225 April)*, 2003.

[95] G. Sun, J. Chen, W. Guo, and K. Liu. Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs. *IEE Signal Processing Mag.*, 22(4):12–23, 2005.

[96] V. Tam, K. Cheng, and K. Lui. Using micro-genetic algorithms to improve localization in wireless sensor networks. *Journal of Communications*, 1(4):1–10, 2006.

[97] Y. Tian, H. Wei, and J. Tan. An adaptive-gain complementary filter for real-time human motion tracking with marg sensors in free-living environments. *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, 21(2):254–265, 2013.

[98] C. Toth, D. A. Grejner-Brzezinska, and S. Moafipoor. Pedestrian tracking and navigation using neural networks and fuzzy logic. *in 2007 IEEE International Symposium on Intelligent Signal Processing*, 2007.

[99] Q. Wang, X. Zhang, X. Chen, R. Chen, W. Chen, and Y. Chen. A novel pedestrian dead reckoning algorithm using wearable emg sensors to measure walking strides. *in 2010 Ubiquitous Positioning Indoor Navigation and Location Based Service*, pages 1–8, 2010.

[100] H. WeinBerg. An-602: Using the adxl202 in pedometer and personal navigation applications. *Analog Devices, Tech. Rep.*, 2002.

[101] Y. Weng, W. Xiao, and L. Xie. Total least squares method for robust source localization in sensor networks using tdoa measurements. *International Journal of Distributed Sensor Networks*, 2011.

[102] Widyawan, M. Klepal, and S. Beauregard. A backtracking particle filter for fusing building plans with pdr displacement estimates. *Proc. 5th Workshop Positioning, Navigation and Communication (WPNC 08) IEEE Press*, pages 207–212, 2008.

[103] J. Wilson, V. Bhargava, A. Redfern, and P. Wright. A wireless sensor network and incident command interface for urban firefighting. *Proc. 4th Ann. Int'l Conf. Mobile and Ubiquitous Systems: Networking & Services IEEE Press*, pages 1–7, 2007.

[104] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. *in Proc. 10th international conference on Ubiquitous computing ACM*, pages 114–123, 2008.

[105] O. J. Woodman. An introduction to inertial navigation. *University of Cambridge Computer Laboratory, Tech. Rep. 696*, 2007.

[106] M. Worrell and A. MacFarlane. *Phoenix Fire Department Radio System Safety Project, tech. report, City of Phoenix Fire Dept.*, 2004.

[107] H. Wymeersch, J. Lien, and M. Z. Win. Cooperative localization in wireless networks. *Proceedings of the IEEE*, 97(2):427–450, 2009.

[108] S. Yang and Q. Li. Ambulatory walking speed estimation under different step lengths and frequencies. *in 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 24(17):658–663, 2010.

[109] X. J. Yang, K. Y. Xing, K. L. Shi, and Q. Pan. Dynamic collaborative algorithm for maneuvering target tracking in sensor networks. *Acta Automatica Sinica*, 33(10):1029–1035, 2007.

[110] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, M. Schiek, S. Leonhardt, T. Falck, P. Mäḧoñen, and R. Magjarevic. 4th international workshop on wearable and implantable body sensor networks (bsn 2007). *ser. IFMBE Proceedings, S. Leonhardt, T. Falck, and P. Mäḧoñen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg*, 2007.

[111] C. You, Y. Chen, J. Chiang, P. Huang, H. Chu, and S. Lau. Sensor-enhanced mobility prediction for energy-efficient localization. *in Proceedings of the 3rd Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks, Reston, Va, USA*, 1:565–574, 2006.

[112] M. Youssef and A. Agrawala. The horus wlan location determination system. *Proc. 3rd Int'l Conf. Mobile Systems, Applications, and Services, ACM Press*, pages 205–218, 2005.

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

[114] W. S. Zhang and G. H. Cao. Dctc: dynamic convoy tree-based collaboration for target tracking in sensor networks. *IEEE Transactions on Wireless Communications*, 3(5):1689–1701, 2004.

[115] T. Zhao and A. Nehorai. Information-driven distributed maximum likelihood estimation based on gauss-newton method in wireless sensor networks. *IEEE Transactions on Signal Processing*, 55(9):4669–4682, 2007.

[116] Y. Zhu and L. M. Ni. Probabilistic wakeup: adaptive duty cycling for energy efficient event detection. *in Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07), Chania, Greece*, 2007.