University of Trento

Department of Information Engineering and Computer Science

Doctoral School in

Information and Communication Technology

Uncertainty aware localization for autonomous robots

PhD Candidate

Valerio Magnago

Advisor

Prof. Luigi Palopoli

Co-Advisor

Prof. Daniele Fontanelli

To my family

Abstract

Autonomous mobile robots are undergoing an impressive growth. They are successfully used in many different contexts ranging from service robots to autonomous vehicles. These robots are expected to move inside the environment and, in general, to perform some operation autonomously. Their reliability strongly depends on their capability to accommodate the uncertainty generated by their interaction with the physical world.

The core functionality for every autonomous mobile robots is the ability to navigate autonomously inside a known environment. The navigation task can be decomposed in identify where to go, plan and follow the route to reach the goal. In order to follow the planned path the robot needs to accommodate the actuation noise. To accommodate these noise the knowledge of the pose and speed of the robot inside the environment is needed. The more accurate the localization of the robot, the better the actuation error can be compensated for.

Localisation is the process of establishing the correspondence between a given map coordinate system and the robot local coordinate system relying on its perceptions of the environment and its motion. Sensors are affected by noise, and in time, ego-motion estimation alone diverges from the robot's true pose. Robot exteroceptive sensors can give fundamental information to reset the pose uncertainty and relocalise the robot inside the environment, hence mitigating the dead-reckoning process.

Most of the localization systems presented in the state-of-the-art focus on the maximization of the localization accuracy by leveraging the natural features of the environment. In these systems, the maximum achievable accuracy is tightly coupled with the perceivable information embedded in the different regions of the environment. Therefore, the localization uncertainty cannot be adapted to the level of accuracy desired by the users and only few approaches can provide guarantees on the localization performance.

In contrast, by infrastructuring the environment, it is possible to obtain a desired level of uncertainty. Current approaches tend to over-design the infrastructure in dimension and supported measurement frequency. They provide far more accuracy than required in most areas of the environment in order to guarantee the tightest constraints that often are required only in limited regions.

The ability to adapt to the location-dependent uncertainty is more than just a desirable property for a localisation system, since it helps in the reduction of the system consumption, in the minimization of external infrastructures and in the relaxation of the assumptions to be made on the environment. In line with the considerations above, localisation throughout this thesis is not seen as the process that always has to maximise the accuracy of the estimated robot pose. On the contrary, localisation is considered as the process that minimises an objective function related

to the infrastructure's cost, to the power consumption and to the computation time, being subject to some requirements on the localization accuracy.

Contents

Chapter 1

Introduction

"Robotics is the science of perceiving and manipulating the physical world through computer-controlled devices"[121]. Robotic systems are placed in the physical world; they perceive information about their environment through sensors and manipulate it through actuators. They are successfully used in many different contexts ranging from service robots [127] to autonomous vehicles [105]. In any conceivable application, in order to accomplish autonomous operations, robots have to be able to master the unavoidable uncertainty that exists in the physical world.

As depicted in Figure 1.1, researchers usually split the task of moving autonomous robots in three main sub-problems: *planning*, *control*, and *localization*. The *planner* is responsible for selecting a sequence of actions (e.g., schedule the places to visit) and for finding a physically executable trajectory by which the robot can reach the target location. The *control system* commands the actuators of the robot in order to ensure that its motion satisfies the planner's requirements. *Localization* is the process of establishing the correspondence between a given map coordinate system and the robot local coordinate system (i.e., to estimate the pose of the robot with respect to the map) relying on its perceptions of the environment and its motion. Localization is an active and increasing research area in robotics, constantly providing new ideas to improve robustness, precision and tools for theoretical analysis.

In the literature, the ability of an autonomous mobile system to precisely localize itself within a given map is often recognized as a fundamental prerequisite for its safe autonomous navigation and transportation. In [35], localization has been referred as *"the most fundamental problem to providing a mobile robot with autonomous capabilities"*. One of the main problems for an effective localization algorithm is the uncertainty involved along the platform motion. In fact, the robot actuators (e.g., motors) follow the desired commands with a certain level of tolerance; the internal models of the robot are approximated to provide timely responses; the onboard sensors measuring the vehicle ego-motion (e.g., inertial measurement units, wheeled robots encoders) are affected by noise. The impact of these uncertainties is accu-

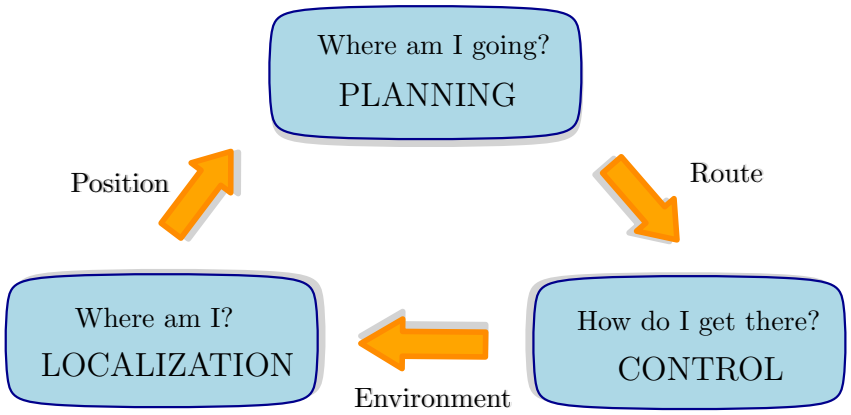


Figure 1.1: Interaction between planning, control, and localization. The localization senses the environment and computes the robot position, the planning computes the route to follow from the position and the control moves the robot in order to follow the route.

mulated in the recursive position estimation (i.e., dead-reckoning). This problem leads the localization error to grow unbounded over time. Therefore, in order to ensure the success of the autonomous operations, it is necessary to periodically reset the depicted error growth using exteroceptive measurements . Mobile autonomous robots follow different strategies to bound their position uncertainty within a map. Usually, they cannot sense their pose directly and infer it through exteroceptive sensors, whose effectiveness is limited by the nature of the measured quantity, their range and their resolution.

The main aspects that characterize the localization problem are the following: i) the knowledge that is available initially and at run-time, ii) the environment features, iii) the motion controller. Depending on the kind of knowledge available in the initial state, it is possible to distinguish 3 different problems: *position tracking*, *global localization*, and *kidnapped robot* problem. Position tracking assumes that the initial pose of the robot is known: the robot's localization can be achieved coping with the noise arising from the dead-reckoning process. It is a local problem since, from the beginning to the end of the operations, the uncertainty is local and therefore confined to a region close to the robot's true pose. In global localization, the robot ego-motion is correctly estimated by dead-reckoning, but there is no information about the initial pose of the robot; thus, no bound can be assumed on the initial pose error. The kidnapped-localization-problem refers to a situation where an autonomous robot is carried to an arbitrary location without him noticing it has been moved (i.e., kidnapped), producing an erroneous ego-motion estimate. The *kidnapping* of a robot is commonly used to test its ability to recover from localization failures. The type of environment can be static or dynamic, depending on whether the robot is the only active entity in the environment or not. Concerning the type of control, the

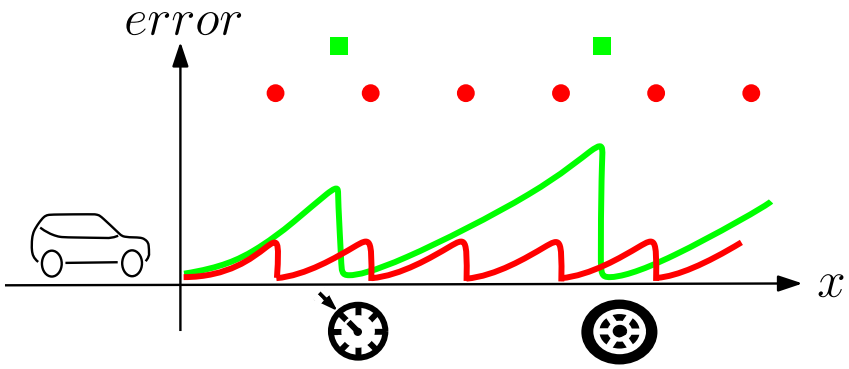


Figure 1.2: 1D example of how optimizing the re-localization can require less measurements while keeping the functionality of the system. The markers on the top represent the re-localization

localization can be classified as passive or active. In the case of passive localization the robot's operations can only be observed. Furthermore,the planning and the control of the robot are not designed to facilitate localization. In contrast in active localization the control algorithm acts in order to minimize the error of the estimated position.

The performance of a localization system usually is evaluated in terms of the average, of the root-mean-square or the 95th percentile of the localization error [101]. Relying on the proposed metrics, engineers usually consider the strictest requirements on localization accuracy while designing and selecting the system and they do not consider that different areas inside the environment may ask for different target uncertainties. This often leads to the choice of overly expensive hardware solutions and to the need for deploying a supporting infrastructure in the environment. Therefore, the ability to adapt to the location-dependent uncertainty is a very important property of a localization system. In fact, it is crucial for the reduction of the system consumption, for the minimization of external infrastructures and for the relaxation of the assumption to be made on the environment.

In the spirit of the considerations above, throughout this thesis, localization is not seen as the process that has to always maximize the accuracy of the estimated robot pose, but as the process that minimizes an objective function related to the infrastructure cost, to the power consumption and to the computation time, subject to some requirements on the localization accuracy. I will illustrate my considerations basing on the following example. This simple mono-dimensional case reported in Figure 1.2 represents a pit-stop sequence for an autonomous robot and the red line represents a possible localization solution. The solution guarantees an accurate pose estimation during all the operations, enabling the vehicle to slow down precisely at the entrance of the box and to stop in the area where the wheels can be changed correctly. This approach requires frequent re-localization in order to keep the drift

of the dead-reckoning process bounded (the red dots in the Figure 1.2 represent re-localization events). Using a high re-localization frequency significantly impacts the costs of hardware and infrastructure besides affecting the risks of the computation delays. In many cases the analysis of the specific task to be performed allows for the design of a more cost-effective localization system. In the proposed case there are two locations where the robot needs a very accurate pose estimation (i.e., before the speed limit and b efore the change tire zone) while in other areas a coarser localization precision could be enough to fulfill the autonomous operation. The green line and the green square in Figure 1.2 represent a possible optimization of the localization that preserves the functionality while requiring lower re-localization frequency. Note that in the example of Figure 1.2 the optimization allows to improve localization accuracy in the two critical locations by using a smaller number of re-localizations.

1.1 Scientific contributions of the thesis

Following the line of reasoning presented above we have conducted several researches that have been already presented in 11 scientific articles most of which are included in this cumulative thesis. Table 1.1 reports the complete list of publications done and, for the ones that are included in this work, we list the chapter where they are presented. The main contributions described in these publications are presented in 4 chapters (4 to Chapter 7). Each chapter presents the results obtained using a particular/specific localization technology:

- Chapter 4 describes an algorithm to optimize the deployment of artificial landmarks to support the re-localization process giving some guarantees on the uncertainty level. A brief introduction to the approach is given below in the Subsection 1.1.1.

- Chapter 5 presents an uncertainty aware authority sharing control algorithm that maintains its functionality even with a reduced infrastructure. This work was developed in long discussions with Marco Andreetto and the experiments with reals robot were performed in collaboration with Stefano Divan. A brief introduction to the approach is given in the Subsection 1.1.2.

- In Chapter 6 is described an Ultra-wideband (UWB) based localization system that can reduce the use of infrastructure by exploiting position dependent requirements on the localization uncertainty level. This system was developed in collaboration with Pablo Corbálan who is pursuing a PhD on UWB localization systems and is introduced in Subsection 1.1.3.

- Chapter 7 presents the first step in developing a localization system based on phase measurement of the backscattered signal from low cost UHF RF

Reference	Appearing in	Thesis main chapter
[81]	IPIN	Chapter 4
[11]	CNS	Chapter 4
[82]	TIM	Chapter 4
[77]	I2MTC	Chapter 4
[83]	ICRA	Chapter 5
[78]	IROS	Chapter 6
[80]	I2MTC	Chapter 7
[79]	TIM	Chapter 7
[13]	I2MTC	
[90]	IPIN	
[21]	EuCAP	

Table 1.1: Scientific contributions of the PhD.

tags. The pursuit of this methodology was defined in collaboration with Andrea Motroni who has RFID-based localization systems as the central focus of his PhD. A more detailed introduction to this work can be found in Subsection 1.1.4.

These chapters are preceded by a chapter of background materials (Chapter 2) and a chapter reporting the related analysis of the state of the art (Chapter 3). Ending the manuscript, Chapter 8 presents a final discussion and comments on the presented works.

1.1.1 Landmark relocalization

The system that is presented in Chapter 4 focus on the improvement of the robustness of any localization system based on any kind of contextual information (e.g., natural landmarks, environment shape) by using artificial landmarks of known pose to relocalize the robot. This approach, while attractive since it does not require any modification to the environment, it is prone to failure in inherently ambiguous or dynamic environments. In these kind of environments the reliability of localization can be improved by placing artificial landmarks, which provide better detection performance [126].

Most of the state of the art techniques focus on solving the so-called art gallery problem. In this problem the objective is to place the artificial landmarks such that at least one of them is visible from every possible pose of the robot. These approaches do not consider the level of information already present in the environment (like natural landmarks), the level of uncertainty that the user needs, and the ego-motion estimated by the robot. Therefore, the theoretical analysis tend to generate an over-sized landmarks deployment. In practice, these approaches are often based on

sensible heuristics and driven by a manual selection of the landmark positions [41]. The absence of a proper formalization prevents the provision of precise guarantees on the system performance and accuracy [26].

We show that the required localization performance can be achieved, via: a less invasive landmarks deployment, the characterization of the robot localization system, and the knowledge of the environment and the stasks the robot needs to accomplish in such environment. In these circumstances, it is possible to develop an optimization process that minimizes the required infrastructure while preserving the performance of the system. We propose a systematic approach to optimize the deployment of artificial landmarks in the environment that takes into account environmental information (e.g., shape, natural landmarks) and ensures that localization uncertainty is kept within the desired boundaries expressed as a function of the robot position and task.

We consider probabilistic models, hence the proposed approach provides stochastic guarantees: the robot localization accuracy lies within some boundaries in a user-defined metric-space with a certain probability that can also be user-defined. For example, in Chapter 4, we define the metric as the euclidean distance projected along the bigger eigenvector of the uncertainty matrix. Using this metric the user can express the requirement in terms of the maximum distance of the robot true pose from the estimated one. Independently from the metric used, the optimization problem can be reduces to a coverage problem, that, as showed by Krause and Guestrin [66], it is NP-hard.

To be solved within a reasonable computation time, some simplification and approximation are required. A first simplifying assumption is to consider a set of discrete location where the landmarks can be placed, this allows for encoding of the problem as the satisfiability of a propositional logic formula (SAT problem). Evey model of such formula corresponds to a deployment of landmarks that satisfies the accuracy requirements. The SAT formalization can be efficiently, although approximately solved using a greedy heuristic.

This concept can be applied to a broad class of drift-less, input-affine wheeled robots used in indoor environments, but in this Thesis, the approach is tailored to the *FriWalk*. The *FriWalk* is the assistive robotics platform (shown in Figure 1.3) developed within the European ACANTO project (for a brief description of the project, see section 2.4) to support the ambulation of elderly people. Due to the *FriWalk* non-linear system dynamics, the evaluation of the proposed approach is done at first through Monte Carlo simulation and then with experiments performed in a real scenario.

1.1.2 Probabilistic authority sharing

Authority sharing is a new control paradigm in which the control action is decided by seeking a trade-off between the control goals (e.g., staying on a course) and the decisions of a human interacting with the system (e.g., the driver of a car). This

Figure 1.3: *FriWalk* and user during experimental campaign validation within ACANTO [12].

paradigm can be exploited to reduce the requirements on the localization accuracy while preserving the system functionality. The key idea is to adapt the control performance to localization uncertainty. The balance between user and controller can change according to the circumstances. The approach is increasingly popular [89], and its potential application areas are many. Again in this work, we show the applicability of the proposed approach on the *FriWalk*.

As commonly done in the design of control algorithms for autonomous robots, the *FriWalk*'s authority sharing control [5, 8, 6] assumes that the localization algorithm provides an accuracy sufficient to make the error in the estimation of the pose negligible for the controller [71]. The intermittent nature of the re-localization information can lead to poor performance (possibly even to instability) of the controller if the average rate is not sufficient to compensate for the system dynamics [115, 93]. Authority sharing offers an elegant and unexpected escape from this problem by limiting the area in which we have to ensure a precise localization.

The key observation is that even a user with mild cognitive impairments is able to maintain a direction of motion when the environment does not require choices (e.g., a corridor). Only in the presence of decision points (e.g., bifurcations, cross-roads, doors), is a constant intervention of the system required. A possible way to see this is that the intelligence of the user can be used to compensate for the reduced information precision on the environment. This behavior mimics a human being

driving his/her car and overriding the suggestions coming from the navigator if the GPS localization is evidently wrong or if an unforeseen obstacle, i.e., road works, blocks the suggested way. Similarly, the autonomous driving system of modern cars gives back control to the human in case, for example in case of heavy weather [74].

We can translate this simple idea into the following design principle: support a frequent re-localization with a high level of infrastructure when close support is required for the user and a light infrastructure when we can shift the authority to the user. This natural strategy has to be complemented by a control algorithm that decides the balance of the authority according to the accuracy of the information on the system state, which is the most important contribution of Chapter 5. Specifically, we propose a hybrid control scheme with two states: robot in control and human in control. The control scheme is Lyapunov-based and gives the authority to one of the two states according to the available localization precision or when the deviation from the path becomes relevant.

The performance on the path following maximum error is experimentally characterized as a function of the uncertainty growth due to dead-reckoning. To the best of our knowledge, this is the first work that directly considers data uncertainties to rule the controller behavior, being most of the literature devoted to the compensation of parametric model uncertainties (e.g., [4]) or the disturbance rejection (e.g., [31]).

1.1.3 UWB

Ultra-wideband (UWB) radios are rapidly gaining traction, thanks to a new generation of smaller and cheaper transceivers (e.g., the DecaWave DW1000 [37]), offering decimeter-level accuracy. In comparison with other techniques (e.g., LiDAR [51], vision [42]) UWB enables both distance estimation (ranging) *and* communication among devices within the *same* radio chip. This is a major asset in system design, especially in robotics applications (e.g., drones and space exploration) where concerns about weight, form factor, and complexity dominate. Other radio-based technologies (e.g., WiFi [28]) offer the same asset but with an order of magnitude decrease in localization accuracy and significantly higher energy consumption.

The control of a robot requires the acquisition of accurate pose information at relatively high frequency, typically in the range 10–30 Hz. In this respect, the application of UWB to robotics faces two key challenges. First, positioning entails several message exchanges with nearby anchors. The commonly-used two-way ranging (TWR), described in the IEEE 802.15.4 standard [61], requires 2 packets per estimate in its simplest form (single-sided TWR, SS-TWR); and, in practice, 4 packets are often used (double-sided TWR, DS-TWR) to improve accuracy. A position estimation from N anchors, therefore, requires $2N$ or $4N$ messages, respectively. As these communication occur on the shared wireless medium, the *scalability* to scenarios with multiple robots is intrinsically at odds with the achievable update rate, i.e., tracking accuracy [27]. Further, although UWB has relatively low *energy consumption*, w.r.t. WiFi, it still consumes hundreds of mW, that are often prohibitive for

small and lightweight robots where the energy budget matters [70]. Based on these considerations, reducing the UWB sampling rate without compromising accuracy is imperative for the practical application of this localization technology.

The proposed approach tackles both challenges by combining the *speed* information acquired by low-cost, low-accuracy proprioceptive sensors (e.g., IMU or odometers) with the *distance* information obtained by high-cost, high-accuracy UWB devices. Ego-motion estimators are widely used in robotics and can usually be executed at high frequency, but unfortunately suffer from drift phenomena induced by dead-reckoning.

The key idea of this approach is to use UWB measurements *intermittently and adaptively* by triggering them only if and when needed to keep the positioning uncertainty under control. We model uncertainty directly by estimating the a-posteriori probability density function (pdf) based on an Unscented Kalman Filter (UKF). This enables us to provide *stochastic guarantees* and to empower the robot user with the ability to *intuitively specify application requirements* in terms of an accuracy threshold combined with a confidence level (e.g., positioning error $\leq$20 cm in $\geq$90% of the samples).

1.1.4 RFID

Among the various RF technologies proposed for indoor localization, the use of RF IDentification (RFID) systems based on passive tags is particularly interesting due to their low cost and the easiness of deployment in a given environment [76, 128].

Once one or more tags are detected, the position of an agent equipped with an RFID reader can be estimated given the position of such tags inside the map. Unfortunately, the commonly used methods for estimating the position based either on scene analysis methods (e.g., fingerprinting) or Received Signal Strength Indicator (RSSI) data are strongly affected by multipath propagation issues [117]. Moreover, a coarse-grained resolution is expected.

In indoor scenarios, better performance can be achieved by exploiting the phase of the tag backscattered signal [38, 24, 50, 22]. In particular, we propose to reverse the problem discussed in [90], where multiple items equipped with passive UHF-RFID tags are localized from the phase measurements collected from a reader installed on a wheeled robot which as a precise measure of its position. Our objective, instead, is to localize and track the robot motion that has been equipped with an RFID reader and with two odometers that can track the robot position even when no tags are detected (dead-reckoning).

A crucial issue of the proposed approach is the inevitable phase ambiguity of the UHF signals received by the RFID reader antenna. To address this problem, we propose an effective dynamic estimator based on an Unscented Kalman Filter (UKF), which is corroborated by a preliminary rigorous observability analysis of the localization problem and the identification of the unobservable region.

1.2 Notation

1.3 Notation

Through this thesis the following notation is used:

Notation	Meaning
x, y,...	Scalar values
$\boldsymbol{x}$, $\boldsymbol{y}$,...	Column vectors
$\boldsymbol{x}_{1:t}$	Sequence of vectors $\boldsymbol{x}_1$, $\mathbf{x}_2$,...,$\boldsymbol{x}_t$
$\boldsymbol{X}$, $\boldsymbol{Y}$,...	Vector-valued random variables
$\mathcal{A}$, $\mathcal{B}$,...	Sets
A, B,...	Matrices
$p(\boldsymbol{X} = \boldsymbol{x})$	Probability density function evaluate at $\boldsymbol{x}$
$p(\boldsymbol{X} = \boldsymbol{x}\|\boldsymbol{Y} = \boldsymbol{y})$	Conditional probability density function at $\boldsymbol{x}$ conditioned on $\boldsymbol{y}$
$E[\boldsymbol{X}]$	Expected value of $\mathbf{X}$
$Cov[\boldsymbol{X}]$	Covariance matrix of $\boldsymbol{X}$
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	Multivariate Gaussian with mean μ and covariance Σ
$\boldsymbol{x}^T$, A^T),...	Transposes of the vector $\boldsymbol{x}$ and the matrix $\boldsymbol{A}$
$\|A\|$	Determinant of the matrix A
$\langle W \rangle, \langle M \rangle, \ldots$	Reference frames

Note that, as used in the standard literature (see, e.g., Section 2.2 in [121]), in this thesis the shorthand notation $p(\boldsymbol{x}) := p(\boldsymbol{X} = \boldsymbol{x})$ is often used to denote the value of the probability density function of the random variable $\boldsymbol{X}$ evaluated at the specific value $\boldsymbol{x}$.

Through this thesis the following abbreviations are used:

Abbreviation	Meaning
EKF	Extended Kalman filter
NP	Nondeterministic polynomial time
PDF	Probability density function
RFID	Radio-frequency identification
SDA	Sensor detection area
UKF	Unscented Kalman filter
UWB	Ulta wideband

Chapter 2

General background

This chapter provides the background notions required to understand the approaches proposed in the following parts of the thesis. First, in Section 2.1 the Bayes rule is introduced and some basic concepts of probability theory are illustrated. In Section 2.2 Bayes filters are introduced starting with a brief mathematical derivation and reporting a description of the 2 most used filters of this family, namely the Kalman filters and the Particle filters. Section 2.3 reports a brief introduction to localization technologies based on radio frequency technologies with particular focus on Ultra-Wide Band (UWB) and Radio-Frequency IDentification (RFID) systems, which are the base technology used in Chapter 6 and Chapter 7, respectively. The last section of this chapter (Section 2.4) provides a general overview of the ACANTO project, which has been the main motivating force and sponsor of the research work reported in Chapter 4 and Chapter 5.

2.1 Probability theory

Robots operations are always corrupted by noise and often measures carry only a partial information of the world. In order to recover robustly the system state working with probability is fundamental. Probabilistic inference allows the mapping of the distribution and uncertainty on action and measure on the system state estimation. Usually the probability that a random value $\boldsymbol{X}$ is equal to a particular value $\boldsymbol{x}$ is defined as $p(X = \boldsymbol{x})$ and it has always a positive value

$$p(\boldsymbol{X} = \boldsymbol{x}) \geq 0.$$

To simplify the notation often $p(X = \boldsymbol{x})$ is shortened to $p(\boldsymbol{x})$. Drawing the value of $p(\boldsymbol{x})$ as a function of $\boldsymbol{x}$ we obtain the probability density function (PDF) of the random variables. By integrating the PDF over a region we find the probability that the random variable $\boldsymbol{x}$ falls in this region. Obviously by integrating $\boldsymbol{x}$ on all the domain we obtain a probability of 1 (the variable should have a value that is in

its domain by definition).

$$\int p(\boldsymbol{x})d\boldsymbol{x} = 1$$

Metrics employed often to describe probability distributions are the expectation of a random variable:

$$E[\boldsymbol{x}] = \int \boldsymbol{x}p(\boldsymbol{x})d\boldsymbol{x}$$

and its covariance:

$$Cov[\boldsymbol{x}] = E[\boldsymbol{x} - E[\boldsymbol{x}]]^2 = E[\boldsymbol{x}^2] - E[\boldsymbol{x}]^2$$

The joint probability.

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(X = \boldsymbol{x}\, and\, Y = y)$$

expresses the probability that $\boldsymbol{X} = \boldsymbol{x}$ and jointly $\boldsymbol{Y} = \boldsymbol{y}$. If the random variable $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent:

$$p(x, y) = p(x)p(y) \tag{2.1}$$

The conditional probability instead expresses the distribution of $\boldsymbol{X}$ knowing the value of the random variable $\boldsymbol{Y}$.

$$p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{X} = \boldsymbol{x}\, given\, \boldsymbol{Y} = \boldsymbol{y}) \tag{2.2}$$

It can be extracted from the joint probability as:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{y})}$$

Notice that if the variables are independent (eq. 2.1) the joint probability does not add any information to the distribution of $\boldsymbol{X}$.

$$p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{x})$$

Knowing $p(\boldsymbol{x}|\boldsymbol{y})$ and $p(\boldsymbol{y})$ it is possible to use the theorem of total probability to obtain the not conditioned distribution of the random variable $\boldsymbol{X}$:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})d\boldsymbol{y}$$

Instead if we want to "reverse" $p(\boldsymbol{x}|\boldsymbol{y})$ to $p(\boldsymbol{y}|\boldsymbol{x})$ we can use the Bayes rule:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})} = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{\int p(\boldsymbol{y}|\boldsymbol{x}')p(\boldsymbol{x}')d\boldsymbol{x}'} \tag{2.3}$$

Considering that the denominator is not a function of $\boldsymbol{x}$ it can be considered as a normalizer and the the Bayes rule it is often shortened as:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \eta p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$$

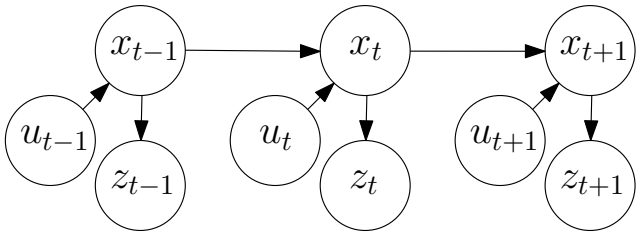


Figure 2.1: Evolution of control, state and measure represented as Baesyan Network.

This rule is the basis for the most used filters in robotics.

When estimating the state of a robot at time t ($\boldsymbol{x}_t$) usually a set of measurements ($\boldsymbol{z}_{1:t}$) and a set of control actions ($\boldsymbol{u}_{1:t}$) are available. Using the definition of conditional probability (eq.(2.2)), the state estimation can be expressed as estimating the probability distribution of the current state conditioned to the measures, controls and past states distribution:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}).$$

Using the same approach we can estimate the distribution of the predicted measure at time t as $p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})$. If $\boldsymbol{x}$ is a complete state representation of the system (i.e. the state is the best predictor of the future), the process can be modeled as a Markov chain (see Fig. 2.1) and the dependency from the past measurements and control action can be removed:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)$$
$$p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) = p(\boldsymbol{z}_t|\boldsymbol{x}_t)$$

The most up to date belief estimation of the state collecting all information up to time t (i.e., $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t})$) is called posterior and usually indicated as $bel(\boldsymbol{x}_t)$. Instead the distribution given from all the control up to time t and the past measures up to time $t-1$ (i.e., $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})$) it is called prediction and indicated as $\overline{bel}(\boldsymbol{x}_t)$. Computing $bel(\boldsymbol{x}_t)$ from $\overline{bel}(\boldsymbol{x}_t)$ it is usually called correction or measurement update since it leverage the measure distribution given from the sensors ($p(\boldsymbol{z}_t)$).

One of the most used and studied PDF is the Gaussian and its general form is:

$$p(\boldsymbol{x}) = \det(2\pi\Sigma)^{0.5} \exp\{-0.5(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma(\boldsymbol{x}-\boldsymbol{\mu})\}$$

where $\Sigma = Cov[\boldsymbol{x}]$ is a semidefinite positive matrix and $\boldsymbol{\mu} = E[\boldsymbol{x}]$. The Gaussians are important in statistic and they are recurrent in nature, partially due to the central limit theorem [104].

2.2 Bayes Filters

The most general algorithm for calculating beliefs is the Bayes Filter. The Bayes Filter calculates the belief in a recursive way from the distribution of the measure

Algorithm 1: BayesFilter($bel(\boldsymbol{x}_{t-1})$, $\boldsymbol{u}_t$, $\boldsymbol{z}_t$)

Result: $bel(\boldsymbol{x}_t)$

1 $\overline{bel}(\boldsymbol{x}_t) = \int p(\boldsymbol{x}_t|\boldsymbol{u}_t, \boldsymbol{x}_{t-1})\, bel(\boldsymbol{x}_{t-1})\, d\boldsymbol{x}_{t-1}$;

2 $bel(\boldsymbol{x}_t) = \eta\, p(\boldsymbol{z}_t|\boldsymbol{x}_t)\, \overline{bel}(\boldsymbol{x}_t)$;

and control following the steps reported in Algorithm 1. In the first line the distribution of the control action $p(\boldsymbol{u}_t)$ is used to predict the current state from the previous estimation and this is done by integrating the product of the state transition model probability $p(\boldsymbol{x}_t|\boldsymbol{u}_t, \boldsymbol{x}_{t-1})$ by the probability distribution of the previous state $bel(\boldsymbol{x}_{t-1})$ over all the state space. The second step (line 2) is called measurement update since it improves the state estimation using the measures from sensors. It is given by the product of the measurement likelihood ($p(\boldsymbol{z}_t|\boldsymbol{x}_t)$) and the predicted state ($bel(\boldsymbol{x}_{t-1})$). This recursive filter can be applied successfully only if the initial belief $bel(\boldsymbol{x}_0)$ is known. If we know the exact initial state $bel(\boldsymbol{x}_0)$ we will impose a picky probability distribution centered on the known state, if we do not have any information on the initial belief the PDF will be homogeneously spread on all the possible states. As reported here the Bayes Filter formulation is not implementable to work on-line in general estimation cases. To obtain a solution suitable for real-time applications some assumptions and approximations are needed. A commonly used approximation consist in constraining the PDF shape obtaining fast closed form solution of the filter and/or reducing the state space to a finite domain reducing the integration domain of the filter . For the sake of completeness we now report a summary of the main mathematical steps necessary to obtain the recursive form of the Bayes Filter followed by a quick overview of the principal instance of Bayesian filter used in the robotics community namely the Kalman Filter and the Particle Filter.

2.2.1 Mathematical derivation

In this subsection we want to demonstrate how it is possible to obtain the recursive formulation of the Bayes filter reported in Algorithm 1. The base assumption behind the Bayes filter is to work with a Markov process and therefore we can make predictions for the future state of the system based solely on its present state.

Applying the Bayes rule reported in eq. (2.3):

$$bel(\boldsymbol{x}_t) = p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}) \tag{2.4}$$

$$= \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})}{p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})} \tag{2.5}$$

$$= \eta p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) \tag{2.6}$$

Since $\boldsymbol{x}_t$ is a complete state (Markov assumption) we can remove the dependence

on the past measure and control action:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) = p(\boldsymbol{z}_t|\boldsymbol{x}_t). \tag{2.7}$$

By substituting eq. (2.7) in eq. (2.4) we get:

$$bel(\boldsymbol{x}_t) = \eta\, p(\boldsymbol{z}_t|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) \tag{2.8}$$

Note that the last term of the equation is the predicted belief ($\overline{bel}(\boldsymbol{x}_t)$).

$$bel(\boldsymbol{x}_t) = \eta\, p(\boldsymbol{z}_t|\boldsymbol{x}_t)\, \overline{bel}(\boldsymbol{x}_t). \tag{2.9}$$

With this equation we have demonstrate the second step reported in Algorithm 1 (i.e., the update step), but we still need to demonstrate the prediction step (i.e., the first step). This can be done by expanding the definition of posterior belief and using the Markov assumption:

$$\begin{aligned}
\overline{bel}(\boldsymbol{x}_t) &= p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) && (2.10)\\
&= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) d\boldsymbol{x}_{t-1} && (2.11)\\
&= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}\boldsymbol{u}_t) p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) d\boldsymbol{x}_{t-1} && (2.12)\\
&= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}\boldsymbol{u}_t) p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t-1}) d\boldsymbol{x}_{t-1} && (2.13)
\end{aligned}$$

Note that dependency on $\boldsymbol{u}_t$ was removed under the consideration that a future action can not influence the value of current state of the system. Using the definition of posterior belief eq. 2.13 can be written as:

$$\overline{bel}(\boldsymbol{x}_t) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}\boldsymbol{u}_t) bel(\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1},$$

This is exactly the demonstrate prediction step reported in the Algorithm 1 completing the mathematical derivation.

2.2.2 Kalman filter

Invented by Swerling (1958) and Kalman (1960) as a filtering tool for linear Gaussian system, the Kalman filters are probably the most popular baesyan recursive filters. The assumptions of the Kalman filters are:

- the system is a Markov process (inherited from baesyan filters);
- the system dynamics is linear and the input noise (ε_t) is a normally distributed random variable with mean 0 and covariance R_t ($N(0, R_t)$).

$$\boldsymbol{x}_{t+1} = A_t\boldsymbol{x}_{t-1} + B_t\boldsymbol{u}_t + \boldsymbol{\varepsilon}_t$$

Given a linear dynamic system the transition probability is gaussian distributed and can be expressed in closed for as:

$$p(\boldsymbol{x}_t|\boldsymbol{u}_t, \boldsymbol{x}_{t-1}) = \det(2\pi R_t)^{-0.5} \exp\{-0.5(\Delta\boldsymbol{x}_t)^T R_t^{-1}(\Delta\boldsymbol{x}_t)\}$$

where $\Delta\boldsymbol{x}_t = \boldsymbol{x}_t - A\boldsymbol{x}_{t-1} - B\boldsymbol{u}_t$.

- the system measure is linear and is characterized by a noise δ_t that is zero mean normally distributed $N(0, Q_t)$.

  $$\boldsymbol{z}_t = C_t\boldsymbol{x}_{t-1} + \boldsymbol{\delta}_t$$

  Under this assumption the measure probability is distributed as

  $$p(\boldsymbol{z}_t|\boldsymbol{x}_t) = \det(2\pi Q_t)^{-0.5} \exp\{-0.5(\Delta\boldsymbol{z}_t)^T Q_t^{-1}(\Delta\boldsymbol{z}_t)\}$$

  where $\Delta\boldsymbol{z}_t = \boldsymbol{z}_t - C\boldsymbol{x}_t$.

- the initial belief is gausian distributed and can be characterized by its 2 moments $\boldsymbol{\mu}_0$ and Σ_0

  $$bel(\boldsymbol{x}_0) = \det(2\pi\Sigma_0)^{-0.5} \exp\{-0.5(\boldsymbol{x}_0 - \boldsymbol{\mu}_0)^T \Sigma_0^{-1}(\boldsymbol{x}_0 - \boldsymbol{\mu}_0)\}$$

Given these assumptions $bel(\boldsymbol{x}_t)$ is gaussian distributed $\forall t \geq 0$ and the Kalman filter (KF) is the best possible estimator of the state (i.e., the one that give us the state belief with greater accuracy). Even if the KF is grounded on linear system, introducing some approximation, have been proposed to apply the filter to nonlinear state estimation. The most known approaches inside the Kalman Filter family that can deal with nonlinear systems are the Extended Kalman Filter (EKF) and the Unscended Kalman filter (UKF). Even if they are not guaranteed to converge they are the best possible linear estimators since they achieve the highest possible accuracy among them. An additional limitation of Kalman filters derives from the assumption of Gaussian distribution. The unimodal bell shape is well suited to the tracking problem, but does not generalize well to the multiple hypothesis tracking that often characterizes global location problems.

The steps of the classic KF are reported in Algorithm 2. The inputs, as for the standard Bayes filter (see Algorithm 1), are the initial state belief and the input and measure distribution; they are expressed by their first and second order momentum underlying their gaussian distribution. The first two lines of the Algorithm translate the prediction step of the classic recursive Bayes filter and the others compute the posterior belief through the measurement update process. In this second part some auxiliary variables are defined; particularly important is K_t called kalman gain. It expresses the relative weight between measurement and prediction: low values of K_t imply that the measure is unreliable compared to the predicted state thus more trust will be given to the predicted state, otherwise if K_t assumes high values more

Algorithm 2: KalmanFilter(μ_{t-1}, Σ_{t-1}, u_t, R_t, z_t, Q_t)

Result: $\boldsymbol{\mu}_t$, Σ_t

1 $\bar{\boldsymbol{\mu}}_t = A_t \boldsymbol{\mu}_{t-1} + B_t \boldsymbol{u_t}$;
2 $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + B_t R_t B_t^T$;
3 $K_t = \bar{\Sigma}_t C_t^T \left(C_t \bar{\Sigma}_t C_t^T + Q_t \right)^{-1}$;
4 $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + K_t(\boldsymbol{z_t} - C_t \bar{\boldsymbol{\mu}}_t)$;
5 $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$;

trust will be given to the measure. Line 4 of the Algorithm 2 uses the kalman gain to weight the innovation (i.e., the difference between the actual and the predicted measurement $\boldsymbol{z}_t - C_t \bar{\boldsymbol{\mu}}_t$)

The derivation of Algorithm 2 from Algorithm 1 leverages the simplification of gaussian PDFs multiplication as sum of exponents. This semplification is nor reported in this work, a clear mathematical derivation can be found in [121]. The complexity of the algorithm usually is dominated by the inversion of the matrix in the update phase (line 3 of the algorithm) that is $O(d^{2.4})$ where d is the dimension of the output vector $\boldsymbol{z_t}$. If the measure space dimension d is much lower that the state dimension n the algorithm complexity is dominated by the $O(n^2)$ operations.

We report now a brief description of the 2 most used members of the KF family: the Extended Kalman filter and the Unscented Kalman filter. Those filters extend the classic KF adding a certain degree of adaptability to non-linear systems generally modeled as:

$$\begin{cases} \boldsymbol{x}_t &= g(\boldsymbol{u}_t, \boldsymbol{x}_{t-1}) + \boldsymbol{\epsilon}_t \\ \boldsymbol{z}_t &= h(\boldsymbol{x}_t) + \boldsymbol{\delta}_t \end{cases} \tag{2.14}$$

Extended Kalman filter

The Extended Kalman Filter is based on the first order Taylor linearizion of the non-linear system reported in eq. (2.14). It follows the same steps of the KF Algorithm 2 but the predicted state and measure are retrieved directly using the nonlinear function and the uncertainty propagation are done using the following linearizion:

$$A_t = \frac{\partial g(\boldsymbol{u}_t, \boldsymbol{x}_{t-1})}{\partial \boldsymbol{x}_{t-1}}$$
$$B_t = \frac{\partial g(\boldsymbol{u}_t, \boldsymbol{x}_{t-1})}{\partial \boldsymbol{u}_t}$$
$$C_t = \frac{\partial h(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}$$

The EKF approximations do not work well when the PDF of the belief is spread over a large part of the nonlinear function causing the linear approximation to fail.

Algorithm 3: UnscentedKalmanFilter(μ_{t-1}, Σ_{t-1}, u_t, R_t, z_t, Q_t)

Result: μ_t, Σ_t

1 $\chi_{t-1} = \begin{pmatrix} \mu_{t-1} & \mu_{t-1} + \gamma\sqrt{\Sigma_{t-1}} & \mu_{t-1} + \gamma\sqrt{\Sigma_{t-1}} \end{pmatrix}$;
2 $\bar{\chi}_t^* = g(u_t, \chi_{t-1})$;
3 $\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\chi}_t^{*[i]}$;
4 $\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} \left(\bar{\chi}_t^{*[i]} - \bar{\mu}_t\right) \left(\bar{\chi}_t^{*[i]} - \bar{\mu}_t\right)^T + R_t$;
5 $\bar{\chi}_t = \begin{pmatrix} \bar{\mu}_t & \bar{\mu}_{t-1} + \gamma\sqrt{\bar{\Sigma}_t} & \bar{\mu}_{t-1} + \gamma\sqrt{\bar{\Sigma}_t} \end{pmatrix}$;
6 $\bar{\mathcal{Z}}_t = h\left(\bar{\chi}_t\right)$;
7 $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$;
8 $S_t = \sum_{i=0}^{2n} w_c^{[i]} \left(\bar{\mathcal{Z}}_t^{*[i]} - \hat{z}_t\right) \left(\bar{\mathcal{Z}}_t^{*[i]} - \hat{z}_t\right)^T + Q_t$;
9 $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} \left(\bar{\chi}_t^{*[i]} - \bar{\mu}_t\right) \left(\bar{\mathcal{Z}}_t^{*[i]} - \hat{z}_t\right)^T$;
10 $K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}$;
11 $\mu_t = \bar{\mu}_t + K_t\left(z_t - \hat{z}_t\right)$;
12 $\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$;

Unscented Kalman filter

Algorithm 3 reports the pseudocode for the Unscented Kalman filter which computes the posterior belief relying on the Unscented transform. This filter shows better performance compared to EKF on nonlinear systems and it does not require the computation of the first order derivatives. At first a symmetric set of sigma points are generated from the distribution of the state belief (μ_{t-1},Σ_{t-1}). The parameter γ, involved in this operation, is a user defined constant that determines the spread of the sigma points around μ_{t-1}: the larger γ the wider the points distribution and $w_m^{[}i]$ is the weight of the i-th sigma point given from the unscented transform that is function of γ and other 2 user defined constants that are used to incorporate prior knowledge on the state distribution. The application of the nonlinear motion function to the sigma points (line 2) allows for the estimation of the mean (line 3) and covariance (line 4) of the prediction. From line 5 to 10 an analogous procedure is applied to estimate the mean and covariance of the predicted measure and the correlation matrix of the predicted state and the measure distributions. Knowing these matrices it is possible to compute the Kalman gain K and update the estimate of mean and covariance of the state using the standard linear Kalman filter procedure. Usually, to speed up the filter and slightly increase the accuracy, the predicted sigma points are directly used in the measurement function without recomputing the sigma points from the restored predicted mean and covariance.

Algorithm 4: ParticleFilter(χ_t, u_t, z_t)

Result: χ_t

1 $\bar{\chi}_t = \chi_t = \varnothing$;
2 **for** *m = 1 to M* **do**
3 | sample $x_t^{[m]} \sim p(x_t|u_t, x_{t-1}^m)$;
4 | $w_t^{[m]} = p(z_t|x_t^{[m]})$;
5 | add $\langle x_t^{[m]}\, w_t^{[m]} \rangle$ to $\bar{\chi}_t$.;
6 **end**
7 **for** *m = 1 to M* **do**
8 | draw i with probability $\propto w_t^{[i]}$;
9 | add $x_t^{[m]}$ to χ_t.;
10 **end**

2.2.3 Particle filter

The Particle filter is a baesyan filter that approximates the posterior $bel(\boldsymbol{x}_t)$ by using a set of random state samples drawn from this posterior. This representation is nonparametric and does not require any assumption on the shape of the distribution of the stochastic variable influencing the system. The greater the number of particles the better the PDF will be approximate. The density of the particles is proportional to the probability expressed by the PDF. The particle filter algorithm is reported in Algorithm 4. On line 4 a set of hypothetical states $x_t^{[m]}$ is generated sampling from the motion distribution. The resulting sample is then associated with an importance factor ($w_t^{[m]}$) based on the measure probability associated to that particle. The particles are then re-sampled by their importance factor generating the new particle distribution χ_t that represents the posterior belief $bel(x_t)$.

2.3 Radio frequency localization

Radio technology like IEEE 802.11 (WiFi), Bluetooth, Zigbee, RFID, and Ultra-Wideband (UWB), are commonly used in indoor localization systems. Different techniques can be used to determine the position of an agent exploiting radio wave transmissions: *Proximity location*, *Received Signal Strength Indicator*, *Fingerprinting*, *Time of Flight*, and *Angle of Arrival*.

Proximity location is the simplest positioning method. When the receiver detect a signal from a transmitter or from a transponder of known location the system know that is nearby it. The estimated position accuracy depends on the communication range that is proportional to the radiated power: lower waves power imply smaller transmission range and therefore a more accurate localization. The estimation can be further improved by intersecting the detection range of multiple transmitters as

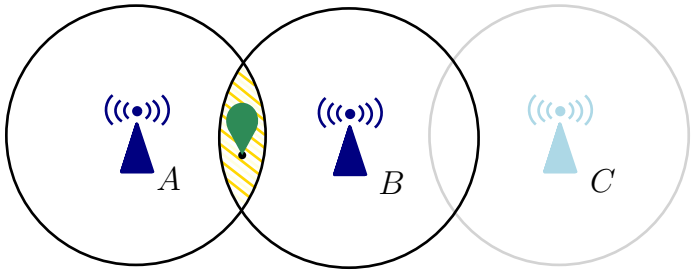


Figure 2.2: *Proximity location.* The green receiver detects the signal from the transmitters labeled as A and B while C is out or range. The yellow area is the intersection of the detection areas of transmitter A and B, and is the set of possible receiver location.

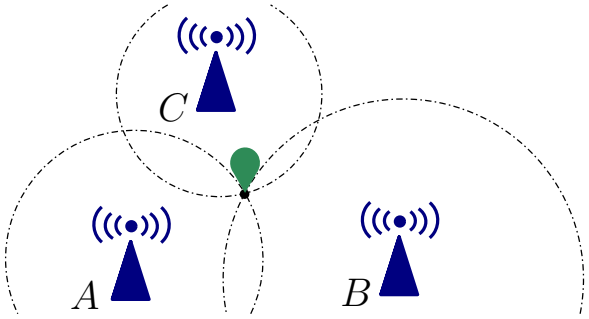


Figure 2.3: Location estimation using trilateration based on range measurement from 3 transmitters of know position.

represented in Figure 2.2. In the figure the transmission range is represented with a circular area, but in general propagation irregularities of electromagnetic waves can generate very complex shape.

Received Signal Strength Indicator (RSSI) is a measure of the signal power at the receiver and can give a better location than proximity. It is possible to link the $RSSI$ value with the distance between the transmitter and the receiver, using a simple propagation model [69]: $RSSI = -10\,n\,log_{10}(d) + A$, where n is the path loss exponent, and A is the RSSI value at a reference distance from the receiver. Multilateration can be used to determine the 2-D location of the receiver, measuring the RSSI from at least 3 nodes (see Figure 2.3). Usually, this approach is simple and cost-efficient, but it suffers from poor localization accuracy: the noise and the frequent wave reflections (multipath fading) characterizing indoor environment generate severe RSS fluctuation, and the attenuation of the signal passing through walls and significant obstacles is challenging to model and predict.

Fingerprinting can improve significantly the precision of multilateration approaches based on RSSI [58]. These methods estimate the position of the robot

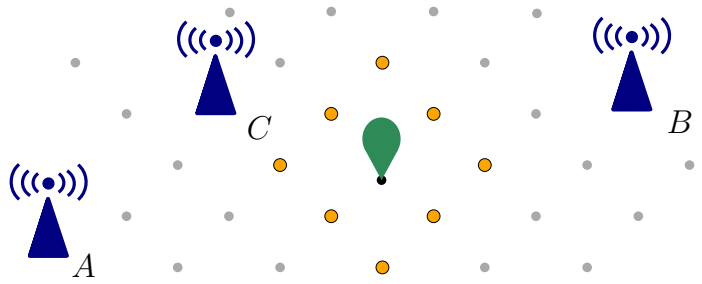


Figure 2.4: *Fingerprinting*. The dots correspond to the database location at which the levels of RSSI of the 3 transmitters (A, B, C) are registered. The 8 yellow locations are extracted, comparing the measured levels of RSSI with the database. The receiver position can be estimated averaging these 8 neighbors.

by matching the measured values of RSSI coming from different transmitters with the ones previously stored in a database. The closest values are then averaged to find the estimate location (see Figure 2.4). The need of a constantly updated database add a significant overhead and rigidity in the system deployment and management.

Time of Flight (ToF) methods use wave propagation time to measure the distance and indirectly estimate the location. They can be subdivided in 3 categories *Time of Arrival*, *Time Difference of Arrival*, and *Phase of Arrival*. *Time of Arrival* (ToA) exploits the signal propagation time to calculate the distance between transmitter and receiver. The receiver can compute the propagation time subtracting the transmission time (usually stored in the message) from the ToA. By multiplying the propagation time by the speed of light, it is possible to obtain the transmitter-receiver distance. *Two-Way Ranging* ToA (TWR) time synchronization is not required and only arrival and departure time difference on each terminal is measured. TWR measures the round-trip signal propagation time to estimate the distance between the 2 terminals (see Figure 2.5). By subtracting the time that the terminal 1 took to retransmit the message from the total time elapsed between transmission and reception of the signal on the terminal 2, it is possible to estimate 2 times the time of flight from one terminal to the other. This technique is called *Single Side-Two Way Ranging* (SS-TWR) since the distance is virtually estimate only on one side. By adding a third transmission from terminal 2 to terminal 1 it is possible to add a further estimation of the time of flight with a symmetric procedure as the one reported above obtaining leading to a more precise distance estimation. This methods it is called *Double Side-Two Way Ranging* (DS-TWR) and is represented in Figure 2.5. Usually in ToA methods the 2D position of a target is determined from distance measurements knowing the position of the reference station. At least 3 reference are needed to estimate without ambiguity the 2D target location from the intersection of circles that usually is done considering the distance measures uncertainties and searching the position with maximum likelihood. The

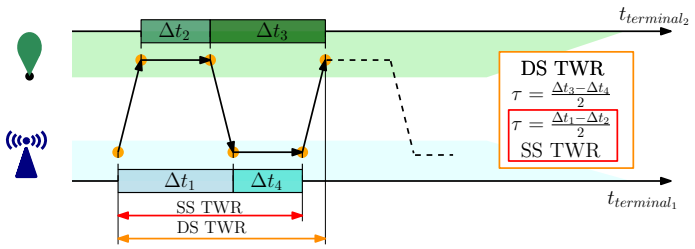


Figure 2.5: Scheme of the message exchanged in Single Side and Double Side *Two-Way Ranging* for distance estimation. τ is the time of flight of the wave from the terminal 1 to the terminal 2.

critical factors that affect ToA estimation accuracy are the signal bandwidth and the sampling rate. High sampling rate, complemented from frequency domain super-resolution techniques, and large spectrum of the signal generate a precise distance estimation. Even if these techniques can be very precise they are prone to error when the transmitter and the receiver are not in direct line of sight. *Time Difference of Arrival* (TDoA) method does not require a signal retransmission. TDoA exploits the delay in signal propagation times from different transmitters measured at the receiver. The TDoA between the signal coming from node i and node j ($\Delta t_{i,j}$) can be converted into a physical distance knowing the speed of light: $\Delta d_{i,j} = c\Delta t_{i,j}$. $\Delta d_{i,j}$ defines a hyperboloid of possible receiver positions (see Figure 2.6), and the intersection of 3 hyperboloids can determine a unique position of the receiver. As for ToF, the accuracy of TDoA depends on the signal bandwidth, sampling rate, and the existence of a direct line of sight between transmitters and receiver. TDoA does not require any receiver-transmitters synchronization.

Phase of Arrival (PoA) uses the relative phase, instead of time, combined with the speed of propagation of the electromagnetic wave to estimate the distance. The propagation distance can be expressed as $d = \lambda(\theta/(2\pi) + n)$ where θ is the measured phase, n is the number of whole wavelengths between sender and receiver and λ is the wavelength that can be computed dividing the light speed by the wave frequency (c/f). In order to estimate the distance it is necessary to remove the ambiguity n from the equation. This is commonly done leveraging the phase measured using multi-frequency waves.

Angle of Arrival (AoA) estimates the 2D position of the agent by measuring the angle of the radio waves coming from at least 2 different reference nodes, as shown in Figure 2.7. These angles are estimated through directional antennas, which usually is an expensive solution. AoA provides an accurate estimation when the transmitter-receiver distance is small. Instead, its accuracy deteriorates, increasing their distance: in situations where the transmitter is far from the receiver, a small error in the angle of arrival calculation maps to a considerable position estimation error.

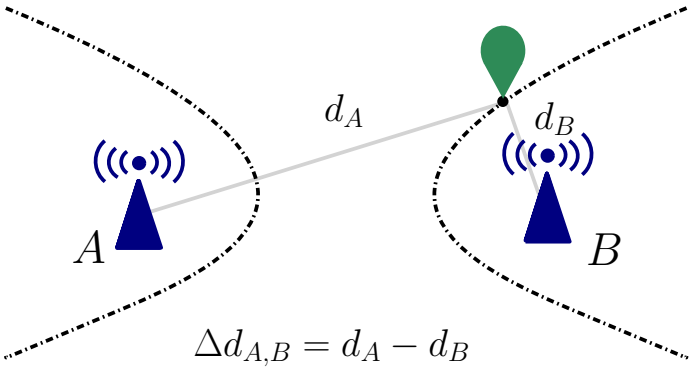


Figure 2.6: *Time Difference of Arrival.* The black dashed line represents the hyperboloid defining all the 2D location characterized with a constant distance difference $\Delta d_{A,B}$ from the 2 blue antennas.

2.3.1 UWB

Ultra-wide band (UWB) localization is based on radio frequencies transmissions. Introduced in the early 1990s [111, 125], UWB communications have received wide interest after the U.S. Federal Communications Commission (FCC) allowed the use of unlicensed UWB communications [45]. In August 2007 ultra-wideband (UWB) physical layer was included in the standard IEEE 802.15.4a among the Low Rate Wireless Personal Area Network. In contrast with conventional radio transmission that transmits information by varying the power level, frequency, and/or phase of a sinusoidal wave, UWB transmissions encode the information as a series of energy pulses in a large band of radio frequencies. In general the series is composed by a time-hopping sequence of very short pulses [87], typically ≤ 2 ns, spreading the signal energy across a bandwidth ≥ 500 MHz. Their short duration assures the resistance against multipath effects and gives a supreme time resolution. To prevent interference on radio signals the UWB has a very strict power emission limits thus UWB can be used only on short-range applications (e.g., indoor).

The first commercial systems, developed in the context of the IEEE 802.15.3a standardization process, were intended for high data rate, short range personal area networks (PANs) [88, 106]. More recent emerging applications of UWB are for sensor networks. Such networks combine low to medium rate communications with positioning capabilities. The very large bandwidth of UWB provides excellent time resolution and allows UWB radios to accurately measure the time of arrival (TOA) of a signal. Moreover, measuring and analyzing the channel impulse response allow

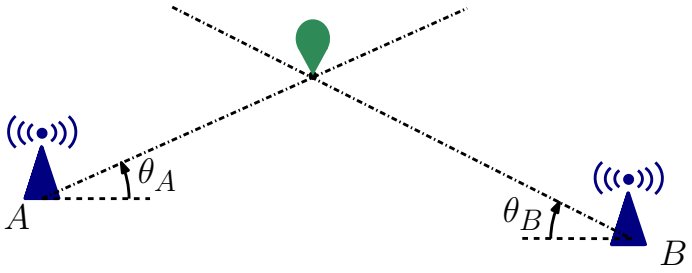


Figure 2.7: *Angle of Arrival.* The black dot-dashed line represents the estimated wave direction, and the position of the green marker is estimated from the intersection of these lines.

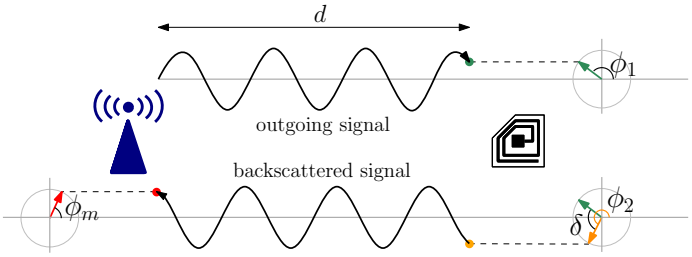


Figure 2.8: Backscattered phase from UHF-RFID tag. ϕ_1 is the phase delay generated in one way travel ($\phi_1 = 2\pi\frac{d}{\lambda} \pmod{2\pi}$); δ is the phase shift characteristic of the tag and the frequency of the wave; ϕ_m is the measured phase at the reader ($\phi_m = \pi\frac{d}{\lambda} + \delta \pmod{2\pi}$).

to distinguish the direct path from multipath components. As a result, time-based positioning techniques like TWR, ToA and TDoA are commonly used in UWB radios localization system, allowing centimeter-level accuracy in ranging, and a low-power and low-cost implementation of communication systems.

2.3.2 RFID

Radio-frequency identification (RFID) uses electromagnetic-radio waves to identify and track tags. Usually an RFID localization system is composed by several tags acting like transponder and an antenna working as a radio receiver and transmitter. The fundamental different with UWB system is that tags have to be interrogated to provide measurements. Normally the transmitted signal activate the transponder that reflects back the signal shifted by a phase value that a characteristic of the tag (as shown in Figure 2.8). The received signal is attenuated and has a different phase if compared to the transmitted signal and these difference are function of the distance that separate the antenna with the tag.

There are two type of tags: passive tags and active tags. Passive tags are powered

by the radio waves transmitted from the RFID reader have a limited range but are very cheap. Active tags are powered by a battery thus can be read at a greater range from the RFID reader (up to hundreds of meters) but are more expensive. Another categorization of RFID system is near field ($\leq$ 14 MHz) and far field ($\geq$ 400 MHz). Near field communication maximum range is around 1 m and typically is based on inductive coupling. Typical RFID far field frequencies are in the UHF bands and its range is greater that the near field.

Ultra High Frequency Radio Frequency IDentification (UHF-RFID) based on passive tags is a promising solution by virtue of its low cost, its detection range and easiness of installation. UHF-RFID have a larger reading range (i.e. several meters) if compared with RFID techniques working of lower ranges. Moreover, the wide-beam reader antennas allows to manage many tags at the same time thanks to the anti-collision algorithm implemented in the EPC Global Class1 Gen2 protocol.

The common measure exploited on UHF RFID-based positioning system is the RSSI measuring the amplitude of the signal backscattered by a tag. This measure is strongly dependent on the environment and tag characteristics (e.g. material, type, orientation) and it suffers from multi-path propagation. Therefore, the definition of a reliable path loss model linking the distance and RSSI is complex and the expected localization error is in the order of a few meters. Tedious fingerprinting technique can be used to characterize UHF RFID RSSI to improve this performance.

Better accuracy can be achieved exploiting PDoA of the backscattered signal. A big limitation to PDoA is that the phase accumulated by the electromagnetic wave propagation in round trip between reader's and tag's antennas, has a circular periodicity as can be seen from Figure 2.8. Every phase measurement have a 2π periodicity hence the same measurement is retrieved from all position that have a radial distance of $\lambda/2$ (where λ is the wave length). Nevertheless the measured phase (ϕ_m) is characterized by two main components: $\phi_m = \phi + \delta$, where ϕ is the phase accumulated during the round trip of the signal and δ is a nearly constant phase shift that is function of the tag and the frequency.

Some approaches calibrate δ for each tag in the system before system operation. Others use measures taken from different position or leverage multi-frequency electromagnetic wave to filter out the spurious contribution from the phase measurements. Frequency based methodologies are limited by the regulation which limit the bandwidth of RFID systems, especially in Europe (ETSI band 865-868 MHz). Another possibility to remove the unwanted contribution is a time differentiation of the signal, but in this way it is the radial relative speed antenna-tag is estimated but not their relative position.

2.4 ACANTO

The scientific and technological development of modern world has increased the life expectancy, producing a constant aging of the population. Europe is definitely the

area with the oldest population. Predictions for 2050 state that almost 35% of European population, i.e., more than one individual every three, will be 60 years old or over. Population aging influences the rise of deambulation impediments, increasing the need of mobility devices. For instance, 24% of adults aged 65 or over used at least one mobility aid in 2011 in United States [52]. Since the population in this age range is increasing, especially in advanced countries, the importance of mobility aids is expected to grow in the next years.

A promising way to face these nefarious effects of reduced mobility is the combination of deambulation aids and modern robotic technologies. The European project ACANTO [2] developed a prototype of an assistive robotic walker, called *FriWalk*, capable of physically and cognitively supporting the assisted person. The *FriWalk* is similar to a standard four-wheeled rollator (see Figure 1.3), but it is endowed with actuators, sensors, and computing abilities to localize itself in the environment [93, 48], and to compute [32] and follow [9, 7, 10, 46] safe and comfortable paths guiding the user towards a desired location. A possible use case was that the user asks the *FriWalk* (using a tablet-based interface) to be guided toward the restrooms inside a building and the walker, after computing the fastest and safest path, is able to guide the user to the target location. During the motion the robot offers physical support and helps the user to reach the restrooms, ensuring that he/she moves safely, e.g., by avoiding obstacles [18].

Moreover, ACANTO emphasizes the importance of social interaction among the users. Since social interaction is foundamental for correct aging, a dedicated social network was development, e.g., to bring together two people with similar profiles. Overall, the workflow of ACANTO is depicted in Figure 2.9.

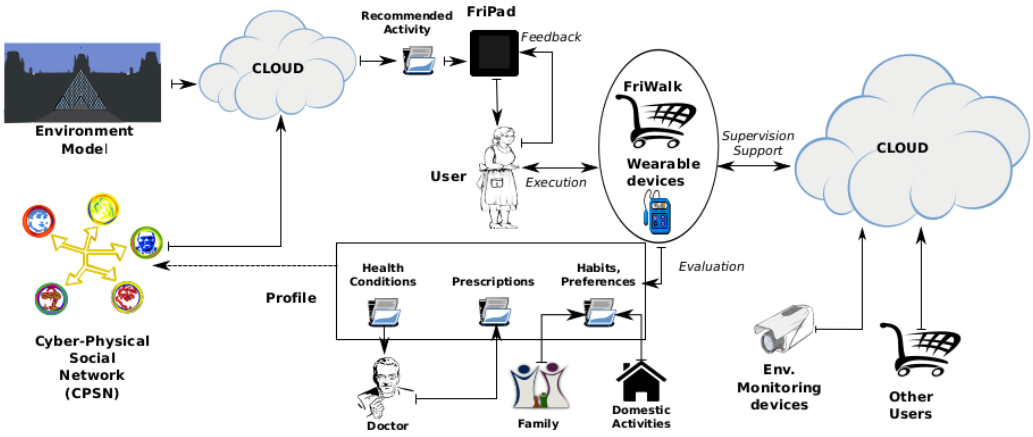


Figure 2.9: Workflow of ACANTO, from the ACANTO proposal.

Chapter 3

State of the art

This Chapter presents the relevant works in the state of the art that are related to the techniques proposed in this thesis. A brief overview of the placement problem, described in Chapter 4, is presented in Section 3.1. Section 3.2 summarizes the state of the art in assistive robots used to support human deambulation. They provide the basis for the control algorithm described in Chapter 5. Finally, Section 3.3 and Section 3.4 report the current approaches for the fusion of inertial information and radio localization technologies. These approaches are used in Chapter 6 and Chapter 7.

3.1 Landmark Placement

The problem of visual landmark selection has been widely investigated in the scientific literature from several points of view. The classic landmarks selection techniques attempt to decompose the environment into a minimal number of maximally sized regions, such that a minimum set of landmarks is visible from any position of a given region, thus ensuring continuous robot re-localization.

One of the first formalizations of this, proposed by Klee in 1970s [97], is the Art Gallery problem. The Art Gallery problem aims at determining the minimum number of guardians that, in an art gallery, can together observe the whole gallery. This problem has been shown to be NP-hard by Lee and Lin in [72], and in [30], Chvatal derived an upper bound of the optimal number of guards as a function of the number of walls in the gallery.

In 1998 Salas and Gordillo [109] generalize the Art Gallery problem to the landmarks placement problem and restrict the perceptual capabilities of the guards to a circular arc with finite minimum and maximum radius. They cast the problem as a coverage problem: find the minimum number of sensors that can cover the maximum possible area. In [108], the problem is extended by requiring that from every position in the map, at least k landmarks are observable. Nazemzadeh et al. [91] solve the art gallery placement for non-omnidirectional sensors, and they derive a

closed-form expression for the optimal distance between landmarks in open space with a triangular detection area of the sensor. In [107], Rupp and Levi propose to further reduce the number of artificial landmarks by using contextual information such as walls and corners. They define a confidence value that numerically describes the expected reliability and the accuracy of the position estimation at a given point. The landmarks configuration is optimized by maximizing the mean confidence value.

In [86], Meyer-Delius et al. present one of the first attempts to ground the landmarks deployment on a solid probabilistic localization method. In particular, they propose to use artificial landmarks to statically resolve estimation ambiguities of the Monte Carlo localization system arising from featureless and symmetrical environments. By resolving the ambiguities, they ensure that a global re-localization is always achievable from every robot position. All the approaches reported above guarantee that the robot's exteroceptive sensors can always gather enough information to localize the robot statically but do not consider that ego-motion estimation given from the proprioceptive sensors can mitigate a temporary lack of external information.

Vitus and Tomlin [122] are among the first to consider both exteroceptive and proprioceptive information for landmarks deployment. They rely on an omnidirectional sensor to detect landmarks and they fix the number of landmarks to be deployed in the environment and the robot trajectory. They propose an incremental approach for placing the landmarks under these assumption: at each iteration, they select, among a set of possible positions sampled around the given trajectory, the one that minimizes the average trace of the uncertainty matrix. Once the procedure places the desired number of landmarks, they refine the position of the selected landmarks via a global optimization procedure that ensures the convergence to an (at least local) optimal solution. A similar solution is presented by Beinhofer et al. in [15]. They suppose to know in advance the trajectory of the robot and the maximum number of landmarks that can be deployed in the environment. They discretize the possible position of landmarks and select using a greedy approach the subset that maximize the conditional mutual information of the states of the robot given the landmark observations. In a subsequent work [16] Beinhofer et al. propose a technique to find the minimum number of landmarks for which a bound on the maximum deviation of the robot from its desired trajectory can be guaranteed with high confidence. Their proposed method incrementally places landmarks while continuously simulating the robot following a single trajectory utilizing a circular landmark detection area. While the approaches reported above can be classified as offline methods, since they place the landmarks before the robot operations, some researchers also focus on online approaches. In contrast to offline, online approaches decide whether to utilize the observed landmarks during operation. Examples of this approach are the work of Thrun [120] and the work of Strasdatet al. [118].

3.2 Autonomous Walkers

Several works in the literature focus on guiding the user inside an environment without sacrificing their perceived freedom using autonomous walkers. In 1990 Goswami et al. [55] introduce the concept of passive robots. Passive robots do not have actuators that can give power to the system, and for this reason they cannot harm the user leaving him the responsibility of the locomotion.

Hyrata et al. were among the first to apply the concept of passive robots to intelligent walkers [59]. They use servo-brakes on the back wheels of the robot to guide the user by changing the apparent dynamics of the system. In [60], the authors use a system similar to [59], but they aim to support user intention. They recognize the motion intention of the user from the user-applied force and hip rotation to generate proper braking torque. Fontanelli et al. in [47], present an optimized braking control that minimize the applied torques to enhance the user comfort. In [6] the authors propose to use a simplified bang-bang policy to control the brake when the robot needs to correct the direction of the user and use an hybrid controller to regulate human-robot interaction. Andreetto et al. [7] propose a path following algorithm for passive assistive robotic walkers based on front steering wheels. This system based on low cost hardware can gently guide the user on the path but does not have the possibility to stabilize users' walking speed for example, to prevent a fall.

Some works abandon the concept of passive robotics to increment the motion capability of the system. A larger set of of maneuvers increases the number of operation that the robot can perform to support the user and improves the comfort. The presence of actuation disrupts the system passivity, with potential safety problems, which can be dealt with using direct or indirect user interfaces [84]. In direct methods the user commands/intentions are directly communicated to the device through joysticks [85], force sensors [56, 75, 124], turn buttons, and voice [68]. Indirect interfaces, instead, recognize the user's movements and/or intents without requiring her/his input. For instance, the user's intention are estimated from the shin position extracted from a laser scanned sensor [73], a depth camera sensor [119], or from the forced applied on the walker extrapolated from rotational encoders on the wheels and current sensors on the actuators [63]. In [5], the authors simulate passive behavior of the walker by leaving the control of the forward velocity to the user. The robot linear speed follows the motion of the user that has the impression of being in control of the motion but with an hybrid scheme the robot can control the angular speed of the system and guide the human operator on the desired path.

3.3 UWB

In [112], Schroeder et al. report the architecture and implementation of a low-cost UWB positioning system. Their system is based on TDoA, and they measured a mean position error of 50 cm with 95 % of errors below 1.5 m. A ToA based system

evaluation is presented in [129]. The system achieves a 2D root mean square error (RMSE) of 7.4 cm by moving the receiver inside a 2.5×2 m rectangular area covered by 4 UWB antennas. In [114], the authors exploit a real-time UWB based 3D positioning system based on 802.15.4a compliant wireless node. They do not apply any filtering technique, and they reported a mean accuracy in the 3D space of 11 cm.

By using a proper filter, the performance can be improved, as reported in [20]. They propose a system composed of 3 antennas capable of tracking a low-speed moving platform with high precision. The platform did not have any sensor for ego-motion estimation, but the reconstruction was done leveraging UWB ranging measurements and using a constant velocity model. They evaluated 3 different filtering approaches: Average Filter, Kalman Filter, and Particle Filter. The best performance was achieved using the KF, whose reported root mean square error (RMSE) is 35 mm.

Several works propose to combine UWB measurements with ego-motion information to deal with more dynamic motions. In [34], Corrales et al. fuse dead-reckoning information from an inertial motion capture system with global position measurements given from a external UWB infrastructure. The accuracy in tracking the human motion that this system can achieve is in the order of a few centimeters. Therefore this system can be used in situations where accurate human body tracking is necessary (e.g., human-machine collaboration).

In [54], Gonzalez et al. combine a UWB and GPS to localize a mobile platform in a mixed indoor-outdoor environment. They fuse the information with odometry reconstruction using a Particle Filter, and they report a maximum error in the indoor section of 20 cm. In [130], Stancic et al. present a UWB based localization for a semi-autonomous floor scrubber. The authors propose to fuse UWB information with encoder measurements to overcome the low sampling rate of the UWB system (usually around 2-10 Hz). The low position measurement frequency leads to inaccurate estimation of the orientation that does not allow proper control of the robot. A good performance of the controller can be obtained by fusing the encoder measurements that usually comes with an high frequency. In [95], the authors present a localization system of a miniature spherical robot based on IMU, encoders, and UWB measurements. The rate of the UWB system was 5 Hz, while IMU and Odometry information were collected at 200 Hz and 125 Hz, respectively. The authors reported an RMSE of 100 mm when using only UWB information and of 60 mm when fusing all information with a KF.

Most of the works in the literature use the UWB system at a fixed sampling rate. To the best of our knowledge, the only adaptive combination of a dead-reckoning system and UWB is presented in [49]. They use a DS-TWR scheme, and the UWB samples are triggered based on the maximum eigenvalue of the covariance matrix estimated via an Error-state Kalman Filter (ESKF) to track a Drone motion. They use a Vicon motion capture system as ground truth, and the lower achievable RMSE

from the proposed system is 7 cm but is adaptable based on global user requirements.

3.4 RFID

Early solution in UHF RFID positioning systems are based on ranging estimation from RSSI. Typically, the accuracy of RSSI based methods are in the order of 1 m as reported in [43]. The accuracy is improved from proximity based methods which exploit only the ID information of the read tags. [57, 99] report centimeter level accuracy, using proximity based methods coupled with an high density deployment of short range RFID tags. In contrast Boccadoro et al. registered an accuracy of 0.1 m using a low long range UHF tags density, in special condition (i.e., with a constrained path of the robot) [19].

In [39], DiGiampaolo et al. obtained state of the art performance but requiring lower tag density, by fusing proximity information with odometry reconstruction through an EKF. They obtained an average error of 0.1 m with 9 UHF RFID tags deployed on the ceiling of a 3×3 m room. Niktin et al. in [96] improve the state of the art accuracy in distance estimation from the signal backscattered from UHF RFID by measuring the phase of the wave.

DiGiampaolo et al. in [40] propose a filter to track an indoor mobile vehicle based on the fusion of encoder information with the phase measured from the signal backscattered from passive UHF RFID. They calibrate the system for the phase delay introduced by the tags and overcame the ambiguity generated from the periodicity of the phase measurements using multiple instance of EKF. They named their filter as particle extended Kalman filter (PEKF). In this approach the maximum tags detection range of 1 m is used to limit the region where the particle are initialized and to toughen the recursive weighting step of the particles. They achieve a mean position error of 4 cm, moving the robot on a 2×1 rectangular path in both a slightly cluttered and a noisy environments. This confirmed the noise robustness and sensitivity of phase measurements if compared to RSSI based systems. Nevertheless, this performance were achieved by only using 2 tags obtaining a significant improvement with respect to other methods available in the literature in terms of accuracy versus tag density. In order to achieve this result, their approach requires a tedious calibration of the characteristic phase delay of each tag.

In [25], Buffi et al. were able to localize moving object on a mono-dimensional conveyor belt at fixed speed with centimeter accuracy, leveraging the time series of phase measurement collected from a static RFID antenna. The phase delay introduced by the tags is removed working with difference of distance, i.e., subtracting the first phase measurement from the time series. Subsequently in [23], the same authors propose a theoretical analysis of the observability of a constant velocity mono-dimensional motion using a UHF RFID system. A train moving at a known speed on a railway is presented as the case of study in their work. In this scenario

they consider that the antenna is fixed on the moving train and used that 3 tags are deployed at known and fixed location in the environment in order to track the train. The signal emitted from the moving antenna is backscattered from the tags deployed in the environment. The tag phase delay is removed from the measured phase subtracting the estimations between them under the assumption that tags share the same offset. The phases subtraction operation combined with a mono dimensional constant speed motion model, make it possible to find a number of candidates location of the train. The most probable position of the train is determined by finding the maximum density point of these solutions.

Chapter 4

Landmark relocalization

Indoor applications based on vehicular robotics require accurate, reliable, and efficient localization. Reasonable trade-offs can be achieved by fusing multiple proprioceptive and exteroceptive sensor data. Especially on low-cost robots, the informations captured from the environment and processed by the localization filter are not always sufficient to ensure an accurate localization. In such cases, a possible solution is to add more information/features in the environment [126]. The process of adding more features in the environment is not standardized and often it is based on common-sense approaches [65].

In this Chapter, we address the problem of minimizing the number of features that have to be deployed inside the environment while ensuring that localization uncertainty is kept within the desired boundaries. The developed approach relies upon the following key elements: a dynamic model describing agents' motion, a model predicting the agents' paths within a given environment and, finally, a formalization in propositional logic of the optimization problem, which can be efficiently (although approximately) solved by a greedy algorithm. The effectiveness of the proposed landmark placement technique is first demonstrated through simulations in a variety of conditions, and then it is validated through experiments on the field.

4.1 Models Overview

This section describes the dynamic models used to formalize the landmark placement optimization problem presented in Section 4.2. In particular, two kinds of kinematic models are considered, i.e., first a very general robot model and then a more specific model belonging to the same class, but tailored to better describe the dynamic of the *FriWalk* (introduced in Section 2.4). The general model in Subsection 4.1.1 emphasizes the fact that the optimal landmark placement technique can be applied to a broad class of drift–less, input–affine wheeled robots used in indoor environments. Indeed, the only underlying assumptions are the presence of a landmark detector with a limited Sensor Detection Area (SDA) and a localization

system based on a dead reckoning position tracking system. The specific model described in Subsection 4.1.2 is instead just a special case of the general one, and it is needed to validate the proposed approach in a practical case study, as shown in Subsection 4.5. To obtain a tractable problem, in the following, we assume that the location and attitude data measured anytime a landmark is detected are used directly to adjust agent position, i.e., without relying on the fusion with data collected from other sensors (e.g., odometers). With this approximation, we are limiting the minimum uncertainty to the measurement uncertainty, but this is good unless we do not require an uncertainty lower than the one offered from the measurement. The corresponding uncertainty analysis is described in Section 4.1.3. Under these conditions, the landmark placement results are indeed expected to be more conservative than the results obtained when a Bayesian filter (e.g., an Extended Kalman Filter – EKF) is used. In fact, if a given selection of landmarks is able to keep positioning uncertainty below given boundaries using only raw sensor data, it is reasonable to assume that the same constraints can be even more safely met when some data fusion algorithm is used, as it is shown in Section 4.5.2.

4.1.1 General model

The fixed, right-handed reference frame for platform localization is referred to as $\langle W \rangle = \{O_w, X_w, Y_w, Z_w\}$, and it is shown in Fig. 4.1. The robotic vehicle is regarded as a rigid body $\mathcal{B}$ moving in the plane $X_w \times Y_w$. If t_s denotes the sampling period common to all onboard sensors, the generalized coordinates of the robot at time kt_s are $\boldsymbol{p}_k = [x_k, y_k, \theta_k]^T$, where (x_k, y_k) are the coordinates of the origin of frame $\langle \mathcal{B} \rangle = \{O_b, X_b, Y_b, Z_b\}$ attached to the rigid body, while θ_k represents the angle between X_b and X_w. The kinematic model of a generic drift–less, input–affine wheeled robot can be represented by the following discrete–time system, i.e.,

$$\begin{cases} \boldsymbol{p}_{k+1} = \boldsymbol{p}_k + G_k(\boldsymbol{p}_k, \boldsymbol{q}_k + \boldsymbol{\epsilon}_k) \\ \boldsymbol{z}_k = h(\boldsymbol{p}_k) + \boldsymbol{\eta}_k \end{cases} \tag{4.1}$$

where $\boldsymbol{q}_k$ is the piecewise input vector of the system between $(k-1)t_s$ and kt_s, ϵ_k is the additive zero-mean uncertainty term affecting input quantities and $G_k(\cdot)$ is the input vector function. Furthermore, $\boldsymbol{z}_k$ (namely the vector of output quantities that can be observed at time kt_s) is given by the sum of $h(\boldsymbol{p}_k)$ (i.e., a generic nonlinear output function of the state) and $\boldsymbol{\eta}_k$, which represents the vector of zero-mean uncertainty contributions when output quantities are measured. If the agent position is estimated through dead reckoning, the accumulation of random contributions $\boldsymbol{\epsilon}_k$ unavoidably leads to large position and orientation uncertainty after a while. If instead, the robot detects, at least sporadically, some artificial landmarks placed at known positions in $\langle W \rangle$, the positioning uncertainty is kept bounded. Consider that, in general, the SDA (denoted as $s(\boldsymbol{p}_k)$ in Fig. 4.1) of any landmark detector exhibits a finite range and a limited angular aperture. However, both detection range and angular aperture may depend on robot position $\boldsymbol{p}_k$.

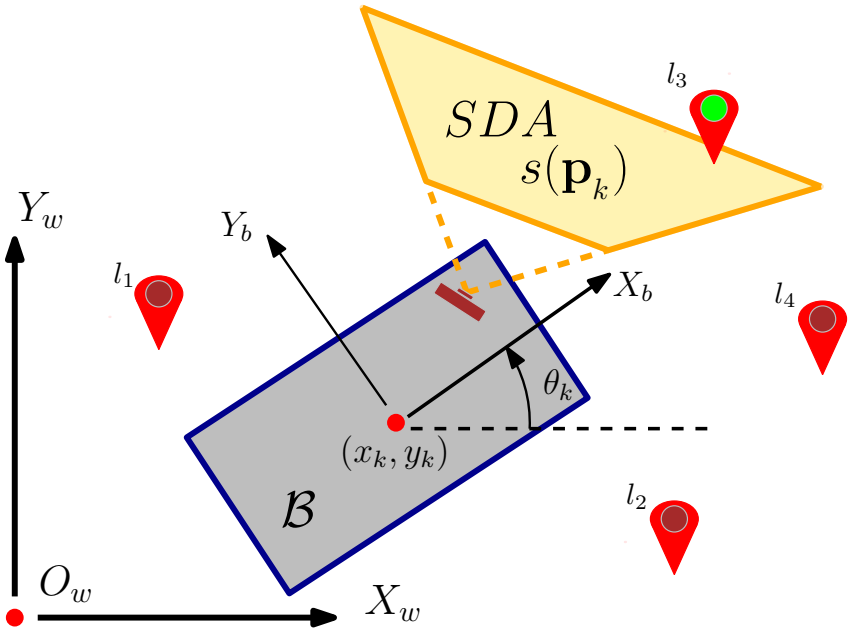


Figure 4.1: Generic representation of a robot to be localized in reference frame $\langle W \rangle$. Landmarks l_1, l_2, l_3 and l_4 are also represented. In particular, l_3 lies inside the SDA of the landmark detection sensor when the robot is located in $s(\boldsymbol{p}_k)$.

4.1.2 A more specific model: the *FriWalk* case

The *FriWalk* is equipped with relative encoders on the rear wheels and with a front monocular camera used to detect specific landmarks (i.e., Aruco codes) placed at known positions in $\langle W \rangle$ (e.g., on the floor) and with a given orientation with respect to X_w [92]. The kinematic model of the *FriWalk* is a unicycle [98]. In this case, the robot planar coordinates (x_k, y_k) (namely, the origin of the body frame O_b with axis X_b pointing forward) refer to the mid-point of the rear axle (see Fig. 4.2). Observe that, with reference to Fig. 4.1, the robot generalized coordinates are still $\boldsymbol{p}_k = [x_k, y_k, \theta_k]^T$. The camera measures the relative position and orientation of the robot with respect to every detected Aruco code. Absolute position and orientation in $\langle W \rangle$ are then estimated as described in [92]. The main parameters of the SDA (which in this case coincides with the field of view of a front camera) are: the maximum and minimum detection ranges (denoted as R and r, respectively) and the camera aperture angle α, as shown in Fig. 4.2. It is worth noting that the analysis are done considering a symmetric trapezoidal SDA generated from a camera mounted with a negligible roll angle detecting landmarks placed on the ground, but is not limited to this: landmark can be placed anywhere and any SDA shape is

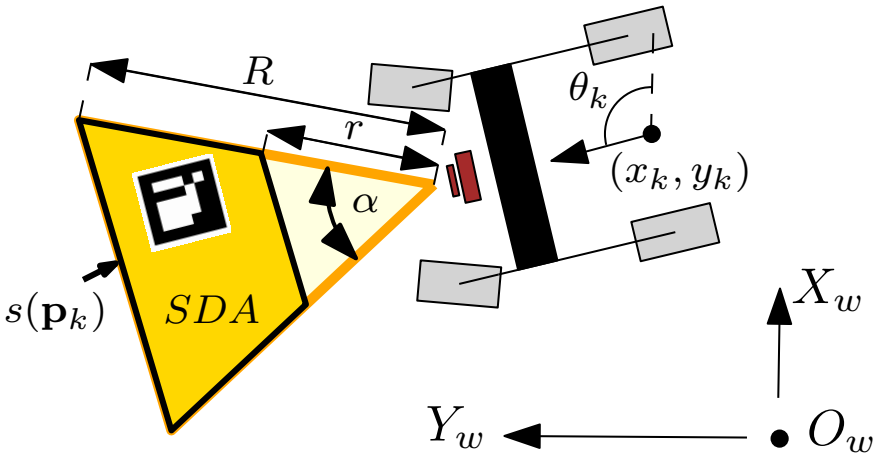


Figure 4.2: Geometrical parameters of the *FriWalk* model and of the SDA of the landmark detection sensor.

supported. With reference to the general model described by expression (4.1), in the case of the *FriWalk* the input vector function of the system is

$$G_k(p_k, q_k + \epsilon_k) = \begin{bmatrix} (v_k + \epsilon_{v_k}) t_s \cos\theta_k \\ (v_k + \epsilon_{v_k}) t_s \sin\theta_k \\ (\omega_k + \epsilon_{\omega_k}) t_s \end{bmatrix}. \tag{4.2}$$

where input vector $\boldsymbol{q}_k = [v_k, \omega_k]^T$ includes the angular and linear velocities of the robot (denoted as ω_k and v_k, respectively) at time t_k. The additive input noise $\boldsymbol{\epsilon}_k = [\epsilon_{v_k}, \epsilon_{\omega_k}]^T$ due to the finite resolution and tick reading errors of both wheels encoders can be reasonably assumed to be white and normally distributed. As a consequence, the covariance matrix of the noise associated with v_k and ω_k is

$$E = \begin{bmatrix} \sigma_v^2 & \sigma_{v\omega} \\ \sigma_{v\omega} & \sigma_\omega^2 \end{bmatrix},$$

where σ_v^2 and σ_ω^2 represent the variances of v_k and ω_k, respectively, while $\sigma_{v\omega}$ is the covariance between them. Finally, the output function $h(\boldsymbol{p}_k)$ of system (4.1) just coincides with the state of the system itself, i.e. $h(\boldsymbol{p}_k) = \boldsymbol{p}_k$. Therefore, $\boldsymbol{z}_k = \boldsymbol{p}_k + \boldsymbol{\eta}_k$, where $\boldsymbol{\eta}_k$ is the vector of uncertainty contributions associated with the measurement of position and orientation based on landmark detection (see Section 4.5 for further details). In particular, the covariance matrix of $\boldsymbol{\eta}_k$ is

$$N = \begin{bmatrix} \sigma_{c_x}^2 & \sigma_{c_{xy}} & \sigma_{c_{x\theta}} \\ \sigma_{c_{xy}} & \sigma_{c_y}^2 & \sigma_{c_{y\theta}} \\ \sigma_{c_{x\theta}} & \sigma_{c_{y\theta}} & \sigma_{c_\theta}^2 \end{bmatrix} \tag{4.3}$$

where $\sigma_{c_x}^2$, $\sigma_{c_y}^2$ are the variances associated with the camera-based measurements of the robot planar position along axis X_w and Y_w, $\sigma_{c_\theta}^2$ is the variance of the orientation

measurements with respect to X_w, and terms $\sigma_{c_{xy}}$, $\sigma_{c_{x\theta}}$ and $\sigma_{c_{y\theta}}$ represent the covariances between pairs of measured quantities.

4.1.3 Uncertainty Analysis

If a non-Bayesian estimator is used and one landmark is detected at time kt_s, the covariance matrix $P_k \in \mathbb{R}^{3\times3}$ of the state estimation error simply coincides with (4.3), i.e., $P_k = N$. In such conditions, the positioning uncertainty depends on the metrological features of the vision system used to measure the relative position and orientation of the robot with respect to the landmark lying in the SDA of the camera. However, when no landmarks are detected, positioning uncertainty tends to grow due to the accumulation of the noise introduced by dead reckoning (e.g., due to the wheels encoders used for odometry, as explained in Section 4.1.2). In this case, the evolution of P_k as a function time can be obtained, to a first approximation, from the linearization of the state equation of system (4.1) around the estimated state. Thus, assuming that $\boldsymbol{\epsilon}_k$ and $\boldsymbol{p}_k$ are uncorrelated $\forall k$, it follows that

$$\begin{aligned} P_{k+1} \approx & \left(I + \frac{\partial G_k(\boldsymbol{p}_k, \boldsymbol{q}_k)}{\partial p_k}\right) P_k \left(I + \frac{\partial G_k(\boldsymbol{p}_k, \boldsymbol{q}_k)}{\partial \boldsymbol{p}_k}\right)^T + \\ & \frac{\partial G_k(\boldsymbol{p}_k, \boldsymbol{\epsilon}_k)}{\partial \boldsymbol{\epsilon}_k} E \frac{\partial G_k(\boldsymbol{p}_k, \boldsymbol{\epsilon}_k)}{\partial \boldsymbol{\epsilon}_k}^T. \end{aligned} \tag{4.4}$$

Expression (4.4) can be regarded as an application of the law of propagation of uncertainty in the multivariate case [103]. Moreover, assuming that the initial state of the system is known, it is reasonable to set $P_0 = N$.

Consider that, since P_k is a 3×3 matrix, a scalar uncertainty parameter is preferable to monitor and to keep positioning uncertainty under control. Therefore, in the rest of this Chapter the following function will be used to evaluate positioning uncertainty, i.e.,

$$u_p(P_k) = \sqrt{\max \mathrm{Eig}\left(P_k^{x,y}\right)}, \tag{4.5}$$

where $P_k^{x,y}$ refers to the upper 2×2 matrix of P_k, i.e.,

$$P_k^{x,y} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}, \tag{4.6}$$

and operator $\mathrm{Eig}(\cdot)$ returns the eigenvalues of the argument matrix.

The rationale for choosing function (4.5) to set uncertainty constraints is threefold. First of all, it is simple to apply. Secondly, even if $u_p(\cdot)$ might include the orientation contribution, in practice just the uncertainty associated with planar position is typically of interest [122, 16]. Finally, the use of function (4.5) is conservative because, from the geometrical point of view, it can be regarded as the radius of a circle centered in the estimated position and circumscribing the ellipse representing the actual positioning uncertainty in the plane $X_w \times Y_w$. In particular,

$u_p(P_k) \in [u_p^-, u_p^+]$, where $u_p^- = \max(\sigma_x, \sigma_y)$ if the correlation coefficient ρ in (4.6) is equal to 0, while $u_p^+ = \sqrt{\sigma_x^2 + \sigma_y^2}$ if $|\rho| = 1$.

Observe that, using a non-Bayesian estimator, the minimum positioning uncertainty is achieved anytime a landmark is detected. Therefore, $u_p(P_k) \geq u_p(N)\ \forall k$.

4.2 Problem Formulation

Let $\mathcal{P} \subseteq \mathbb{R}^2 \times [0, 2\pi)$ be the set of all configurations reachable by an agent inside the environment, so that $p_k \in \mathcal{P}\ \forall k$. If $\mathcal{D}$ denotes the detectable area (namely the set of points lying in the SDA for at least one of the possible positions of the robot), i.e.

$$\mathcal{D} = \left\{(x, y) \in \mathbb{R}^2 \mid \exists \boldsymbol{p}_k \in \mathcal{P}, (x, y) \in s(\boldsymbol{p}_k)\right\},$$

then $\mathcal{L}_p \subseteq \mathcal{D}$ can be referred to as the set of points where landmarks can be actually deployed. Let $\xi(\boldsymbol{p}_k)$ be the maximum wanted (or target) positioning uncertainty. Note that, in general, $\xi(\boldsymbol{p}_k)$ can be a function of the current robot position (e.g., because locations close to walls require more accurate localization to avoid collisions). Observe also that $\mathcal{L}_p$ has an infinite cardinality. Therefore, to make the landmark placement problem tractable, a finite–element set $\mathcal{L}_f \subseteq \mathcal{L}_p$ should be defined, to ensure that the minimum possible target uncertainty is always achieved, i.e., $u_p(P_k) = u_p(N) \leq \xi(\boldsymbol{p}_k)$, $\forall k$. This condition holds true if, in every position of the chosen environment, at least one landmark lies within the SDA, i.e., $\mathcal{L}_f \cap s(\boldsymbol{p}_k) \neq \emptyset, \forall \boldsymbol{p}_k \in \mathcal{P}$. Of course, the cardinality of set $\mathcal{L}_f$ (denoted with symbol $|\cdot|$ in the rest of this Chapter) should be as little as possible to minimize the search space of possible landmark positions. The resulting minimization problem can be formulated as follows:

Problem 1. *Given $\mathcal{P}$ and $s(\cdot)$, find*

$$\mathcal{L}_f = \arg\min_{\mathcal{L}_x} |\mathcal{L}_x| \ s.t.$$
$$\forall \boldsymbol{p}_k \in \mathcal{P},\ \mathcal{L}_x \cap s(\boldsymbol{p}_k) \neq \emptyset \wedge \mathcal{L}_x \subseteq \mathcal{L}_p.$$

A geometry-based closed-form optimal solution to Problem 1 is reported in [91]. The set $\mathcal{L}_f$ thus obtained is indeed the starting point for the placement optimization problem addressed in this Chapter.

In this respect, to refine the search for optimal solutions, not only a model describing the growth of positioning uncertainty (like the one described in Section 4.1.3) is needed, but also some knowledge of the possible paths followed by the agents is essential. If fully autonomous vehicles are considered, usually the set of possible paths has a finite cardinality and it is well defined. In this case the predefined path can be augmented, considering the localization accuracy required and using a Monte Carlo approach, to consider the uncertainty generated by controlling the system [15]. If human beings are involved instead (like in the case of

the *FriWalk*, the set of possible trajectories is infinite, but the regions of space that are explored with highest probability (i.e., the most likely paths) can be derived statistically from empirical observations [17, 110]. Even if a path $T_i \in \mathcal{T}$ (where $\mathcal{T}$ is the set of all available paths) ideally consists of an infinite number of points, in practice it can be discretized by using the elements of $\mathcal{L}_f$. Indeed, $\forall i, k$, $\exists p_k \in T_i : \mathcal{S}_{i,k} = s(p_k) \cap \mathcal{L}_f \neq \emptyset$. Of course, the mapping between $p_k \in T_i$ and $\mathcal{S}_{i,k}$ is *not* bijective since multiple landmarks can be potentially detected by the same robot. Thus, the landmark placement optimization problem addressed in the next section can be formalized as follows, i.e.,

Problem 2. *Given $\mathcal{P}$, $\mathcal{L}_f$, $\mathcal{T}$ and $\xi(\boldsymbol{p}_k) \geq u_p(N)$, $\forall \boldsymbol{p}_k \in \mathcal{P}$, find:*

$$\mathcal{L} = \arg\min_{\mathcal{L}_x} |\mathcal{L}_x| \; s.t.$$
$$\mathcal{L}_x \subseteq \mathcal{L}_f,$$
$$\forall i \;\; T_i \in \mathcal{T}, \forall k \;\; \boldsymbol{p}_k \in T_i, \;\; u_p(P_k) \leq \xi(\boldsymbol{p}_k).$$

Observe that $\mathcal{L} \subseteq \mathcal{L}_f$. Therefore, the problem is well-posed since at least one solution (i.e., $\mathcal{L}_f$) certainly exists.

4.3 Optimal Landmark Placement

In this section we discuss how to solve Problem 2 by casting it into a binary programming problem, which can be tackled with different solution strategies.

4.3.1 CNF Problem Representation

To represent the problem, we associate to each possible landmark location $l_i \in \mathcal{L}_f$ a boolean variable a_i, such that

$$a_i = \begin{cases} 1, \text{ if a landmark is placed in } l_i, \\ 0, \text{ otherwise.} \end{cases}$$

Thus, a landmark deployment corresponds to an assignment to the boolean variables. The objective is to find a *least* assignment, i.e., an assignment such that the minimum number of variables is assigned the value 1, which satisfies the uncertainty constraints. We model the constraints by identifying all the partial assignments to the variables that lead to a violation. Consider a position $\boldsymbol{p}_s \in T_i$, and assume $u_p(P_s) = u_p(N)$, i.e., the minimum uncertainty in our setting. We simulate the trajectory and compute the evolution of $P_{s+1}, P_{s+2}, \ldots$ along T_i. At the same time, let $\mathcal{S}_{i,j}$ represent the landmark positions within the field of view of the landmark detector at time j along path i. If at time $k+1 > s$, $u_p(P_{k+1}) > \xi(\boldsymbol{p}_{k+1})$, then we have a violation since $\xi(\boldsymbol{p}_{k+1})$ is the maximum position uncertainty allowed at point $\boldsymbol{p}_{k+1}$ (see Fig. 4.3 at time $k+1 = 14$).

In order to avoid it, at least one landmark must be present in one of the positions $\cup_{j=s}^{k} \mathcal{S}_{i,j}$ in view. This condition can be expressed as follows

$$\gamma_{i,s} = \bigvee_{j=s}^{k} \mathcal{S}_{i,j},$$

where, with a slight abuse of notation, the boolean variables associated with the landmark positions are denoted with $\mathcal{S}_{i,j}$. Clearly, a landmark deployment $\mathcal{L}$ that does not satisfy $\gamma_{i,s}$ cannot be a solution of Problem 2, since between $\boldsymbol{p}_k$ and $\boldsymbol{p}_{k+1}$ the uncertainty constraint would be violated. We can repeat this analysis for all starting positions and all trajectories, and collect the clauses in a set Γ. For the problem to be satisfied, it is necessary and sufficient that all the generated clauses evaluate to true. Thus, the function

$$\varphi(a_i, \ldots, a_n) = \bigwedge \Gamma = \bigwedge_{i,s} \gamma_{i,s}$$

evaluates to true for all and only those assignments to the boolean variables $a_1, \ldots, a_n$ which correspond to a correct deployment. Given its form, φ is expressed in Conjunctive Normal Form (CNF). For example, with reference to Figure 4.3, from $\boldsymbol{p}_4 \in T_3$, the platform sees 5 landmarks before $u_p(P_{14}) > \xi(\boldsymbol{p}_{14})$.

The set of landmarks in view is given by $\bigvee_{j=4}^{13} \mathcal{S}_{3,j} = \{l_1, l_3, l_5, l_7, l_{10}\}$, the corresponding clause $\gamma_{3,4}$ is:

$$\gamma_{3,4} = a_1 \vee a_3 \vee a_5 \vee a_7 \vee a_{10}.$$

Since the clauses represent a disjoint operation, a cardinality reduction of the set Γ is convenient. For example, for the following two clauses

$$\begin{aligned} \gamma_{3,4} &= a_1 \vee a_3 \vee a_5 \vee a_7 \vee a_{10}, \\ \gamma_{3,5} &= a_3 \vee a_5 \vee a_7 \vee a_{10}, \end{aligned}$$

we have that $\gamma_{3,5} = 1 \Rightarrow \omega_{3,4} = 1$ but $\gamma_{3,4} = 1 \not\Rightarrow \gamma_{3,5} = 1$. Thus, only $\gamma_{3,5}$ is of relevance for the placement, while $\omega_{3,4}$ can be safely removed and hence reduce the complexity.

A compact representation of Γ is given by a coverage matrix whose columns are the possible landmarks locations $l_i \in \mathcal{L}_f$ and rows are the clauses $\Gamma_{i,s}$. The entry in position (r, c) of such a matrix has 1 if the r-th clause is satisfied by the c-th landmark, or 0 otherwise. An example is shown in Table 4.1. To summarise, Problem 2 has been cast into a binary programming optimisation problem, whose objective is to find $\arg\min_{\mathcal{A}} \sum_{a_i \in \mathcal{A}} a_i$ subject to the set of clauses Γ.

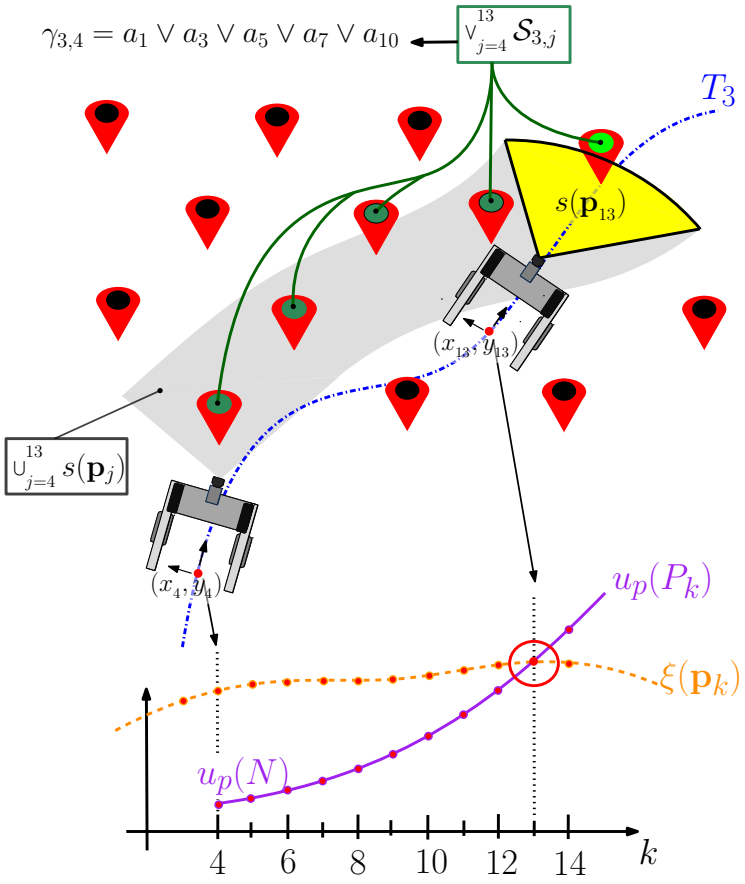


Figure 4.3: Example of uncertainty growth along a sample trajectory. The dashed line represents the uncertainty threshold $\xi(p_k)$, while the solid line is the uncertainty growth when no landmark is in the SDA starting with minimum uncertainty N at step $k = 4$. At step $k = 14$ we have $u_p(P_{14}) > \xi(\boldsymbol{p}_{14})$, therefore there is a violation. On the upper part of the figure, the shaded band represent the union of the areas observed by the camera from step 4 to 13. The placeholder icon stand for possible landmarks position and those belonging to the shaded band will generate the clause $\gamma_{3,4}$.

Table 4.1: Coverage matrix expressing the clause as disjunction of boolean variables: $\gamma_{2,3} = a_1 \vee a_2 \vee a_8 \vee a_9$; $\gamma_{4,1} = a_2 \vee a_3 \vee a_6$; $\gamma_{3,2} = a_2 \vee a_4$; $\gamma_{3,4} = a_1 \vee a_3 \vee a_5 \vee a_7 \vee a_{10}$; $\gamma_{3,5} = a_3 \vee a_5 \vee a_7 \vee a_{10}$.

	1	2	3	4	5	6	7	8	9	10
$\gamma_{2,3}$	1	1	0	0	0	0	0	1	1	0
$\gamma_{4,1}$	0	1	1	0	0	1	0	0	0	0
$\gamma_{3,2}$	0	1	0	1	0	0	0	0	0	0
$\gamma_{3,4}$	1	0	1	0	1	0	1	0	0	1
$\gamma_{3,5}$	0	0	1	0	1	0	1	0	0	1

4.3.2 Optimal Placement

As discussed, to optimise the placement we need to find the least satisfying assignment, i.e., an assignment to the variables $a_1, \ldots, a_n$ such that φ is true and the least number of variables is assigned value 1. There are several ways to formally solve this problem. One approach is to cast it as a logic optimisation problem, and look for a minimum term cover of φ. Observe that the conjunction of the true variables of a satisfying assignment is an implicant of φ. For instance, let $I = \{i_1, \ldots, i_t\}$ be the indices of the true variables of a satisfying assignment. Then, the product term $a_{i_1} \cdot a_{i_2} \cdots a_{i_t}$ logically implies φ, that is, the product term "covers" some of the ones of φ. A *minimal* deployment (i.e., one in which no landmark can be removed without violating the constraints) corresponds to a *prime* implicant of φ. The *minimum* deployment is therefore the *largest* prime implicant.

We thus use a logic optimisation program to find a minimum 2-level cover of φ. Each term of the resulting cover corresponds to a minimal deployment, and we choose the one with the least number of variables. This approach has the advantage that it provides several alternative solutions, corresponding to the various terms of the cover. In our experiments we have used the SIS optimisation software [113]. While this strategy gives us the best solution, the downside lies in its computational complexity, which is exponential in the number of variables and in the number of prime implicants. Our experiments show that the method is practical only in the case of deployments of a limited size. For instance, a layout with 37 locations and 10 constraints is solved in less than a second on a 3.2 GHz Intel Xeon PC with 4 GB of RAM, but already results in almost 8,000 minimal solutions, with the best ones (around 1,000 solutions) using just 4 landmarks. The extension of the same problem to 52 locations and 15 constraints increases the computation time to over 7 minutes, and almost 250,000 minimal solutions; 23 of them use 4 landmarks, while the largest minimal solutions rely on 11 landmarks. Therefore, this approach is impractical for larger deployments.

Alternatively, the problem can be rephrased as a constrained boolean optimisa-

tion, i.e.,

$$\min \sum_i a_i, \quad \text{subject to} \quad \forall i, \forall s, \quad \omega_{i,s} > 0$$

Even if the computational complexity of the problem is still exponential, one can solve the continuous relaxation of the same problem, which is polynomial. Of course, since in this case the variables may take any value between 0 and 1, the solution of the problem in general will be infeasible. Despite this, the relaxed optimal solution (that henceforth will be denoted with $\mathcal{L}_r$) provides a lower bound to the number of landmarks which are required to satisfy the constraints. In the following, we will use the result of this approach to evaluate the performance of the greedy placement algorithm.

4.3.3 Greedy Placement

The greedy algorithm for landmark placement leads to a good approximation of the optimal solution within a negligible computation time. It is based on the greedy heuristic for sub-modular functions described in [94]. In practice, we start with the coverage matrix (A_0) describing Γ, computed as described previously, where the columns are ordered with a decreasing number of elements equal to 1. With reference to Table 4.1, the first column will be l_2, then l_3 and so on. A landmark is placed in the position corresponding to the first column, i.e., the one satisfying the greatest number of clauses. The corresponding satisfied clauses (the matrix rows) are then removed from the matrix, together with the first column, and the matrix is reordered. With reference to Table 4.1, l_2 is added to $\mathcal{L}_g$ and the first three rows are removed. A new matrix A_1 is obtained, and the procedure starts over. The procedure ends when there are no more clauses to meet, i.e., when the matrix is empty. For the case of Table 4.1, the procedure may end with $\mathcal{L}_g = \{l_2, l_5\}$ or with $\mathcal{L}_g = \{l_2, l_3\}$, namely when at most two landmarks are placed. As shown in Section 4.4, despite its simplicity, the greedy solution $\mathcal{L}_g$ turns out to be very effective when compared to the (infeasible) lower bound $\mathcal{L}_r$. As a final remark, we notice that the lower the number of boolean variables shared between clauses, the more the greedy suboptimal solution approaches the best possible one.

4.4 Simulation Results

This section presents the simulation results in different scenarios. Throughout this section, the paramters used in simulation are reported in Table 4.2. For the sake of brevity, only the results with a constant target uncertainty are reported, i.e., $\xi(\boldsymbol{p_k})$ is constant for all $\boldsymbol{p_k}$.

Table 4.2: Numerical values adopted in the simulations.

Symbol	**Value**	**Unit**	**Symbol**	**Value**	**Unit**
$\xi(\boldsymbol{p}_k)$	0.8	m	t_s	0.01	s
r	0	m	R	4.0	m
α	$\pi/2$	rad	$\sigma_{c_y^2}$	30	cm^2
$\sigma_{c_x}^2$	25	cm^2	$\sigma_{c_{xy}}$	0	m^2
$\sigma_{c_\theta}^2$	0.001	rad^2	σ_v^2	0.09	m^2/s^2
$\sigma_{c_{x\theta}}$	0	m·rad	σ_ω^2	0.64	rad^2/s^2
$\sigma_{c_{y\theta}}$	0	m·rad	$\sigma_{v\omega}$	0	$\text{rad}\cdot\text{m}/\text{s}^2$

4.4.1 Realistic Environment

The realistic environment chosen for simulation purposes is the Department of Information Engineering and Computer Science (DISI) of the University of Trento. The *FriWalk* trajectories are generated using the path planner described in [102], which is conceived for robots moving in known structured environments. In this case, the set of trajectories is quite repetitive and regular, and robots moving in the corridor are likely to follow the same path, as clearly visible in Figure 4.4-(a). The regularity of the paths increases the number of shared boolean variables between the clauses, making this a very challenging situation for the greedy algorithm. If we assume to use a visual sensor with the values of R and α as reported in Table 4.2, the potential positions of landmarks determined as described in [91] amounts to $|\mathcal{L}_f| = 2085$. Such positions are represented with blue dots in Figure 4.4-(b). Considering 800 different paths, randomly generated by the path planner and depicted in Figure 4.4-(a), 1889 potential landmarks are observed at least once in at least one trajectory. The positions of these landmarks are highlighted with circled dots in Figure 4.4-(b). By solving the relaxed optimization problem, assuming $a_i \in [0, 1] \subset \Re$ (see Section 4.3.2), the overall optimal number of landmarks is $m_b = \sum_i a_i = 92.6$, which is also a lower bound for the optimal solution. To obtain a feasible deployment from this optimal infeasible solution, we first arrange the values of a_i in descending order. Then we place a landmark in the positions with the highest value (saturating a_i to 1), and then we continue to add landmarks in $\mathcal{L}_r$ until all the clauses are satisfied. In this way, the total number of landmarks is $M_b = |\mathcal{L}_r| = 133$, which is an upper bound of the optimal solution. The greedy algorithm instead leads to the selection of $|\mathcal{L}_g| = 115$ landmarks, i.e. which is included between M_b and m_b bounds confirming the validity of the introduced approximation. Such landmarks are represented with green circled dots in Figure 4.4-(b).

Notice that even if the number of trajectories and of potential landmark loca-

Table 4.3: Maximum, average and standard deviation of localisation uncertainty (4.5) for random and the greedy placement, respectively. All simulation results refer to the realistic scenario shown in Figure 4.4.

	Random deployment densities					
	5%	25%	45%	65%	85%	greedy (6%)
max [m]	35	5.3	1.6	0.8	0.4	0.79
mean [m]	2.8	0.3	0.1	0.07	0.06	0.14
std [m]	4.4	0.5	0.1	0.05	0.1	0.10

tions is quite large, the computation time of the greedy algorithm implemented in Matlab and running on a 3.50 GHz Intel Core i7 with 8 GB of RAM is about 15 minutes. Notice that the computation time comprises the simulations per trajectories of the vehicle motion and of the sensor readings, all coded in Matlab. To lower the computation time, we tested the greedy algorithm only on a subsets of randomly chosen 200 trajectories, and the result was that at most 6% of the overall paths did not satisfy the accuracy limit $\xi(\boldsymbol{p_k})$. In addition, we compared the greedy solution with the result of a naive approach in which different amounts of landmarks are randomly selected from $\mathcal{L}_f$. In particular, between 5% and 35% of possible landmark positions have been chosen repeatedly (i.e. 50 times) with the same probability. For each random placement the percentage of paths satisfying the maximum uncertainty limit $\xi(\boldsymbol{p_k})$ has been estimated. The results are summarized in Figure 4.5. The boxes define the 25-th and 75-th percentile, while the whiskers corresponds to the maximum and minimum value. The thick vertical line corresponds to the percentage of landmarks placed by the greedy algorithm for which all the paths meet the given uncertainty constraint, i.e. $\xi(\boldsymbol{p}_k) = 0.8$ m. It is worth noticing how the greedy solution outperforms the naive random choice. The localization uncertainty obtained with greedy and random placement over 800 trajectories are summarized in Table 4.3, where the maximum, the average and the standard deviation of (4.5) are reported. Observe that the greedy algorithm ensures a very good accuracy, even if only 5.5% of landmark is used (see the thick line in Figure 4.5).

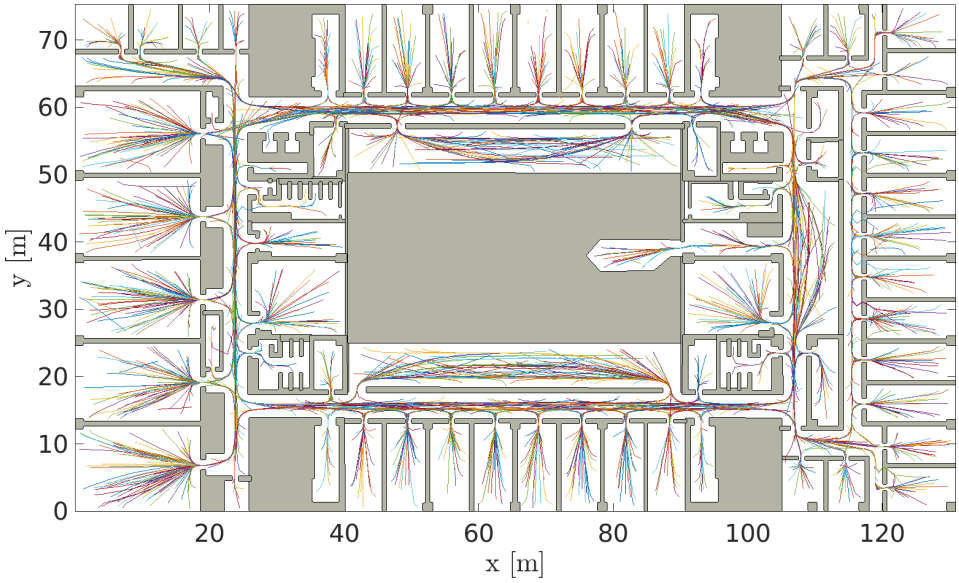
The results of landmark deployment for more realistic, i.e. human-like, trajectories is reported in Figure 4.6. 800 human-like trajectories in a corridor have been synthesized using the Headed Social Force Model (HSFM) [44]. This model emulates the motion of human beings moving in shared spaces and obeys to the kinematic model that falls in the generic representation of (4.1). For this case, we report in Figure 4.6 two different landmark deployments, both based on the results of [91], which gives $|\mathcal{L}_f| = 48$ possible landmark positions if landmark deployment is unconstrained, reported in Figure 4.6-(a) with blue dots, and instead $|\mathcal{L}_f| = 36$ where there is an area where landmarks cannot be placed, shown in Figure 4.6-(b), again in blue dots. Notice that in the latter case the landmark placement in [91] cannot be strictly satisfied in the landmark-forbidden region. For the unconstrained deploy-

ment, the upper and lower bounds to the optimal number of deployed landmarks are $M_b = 3$ and $m_b = 3$, respectively. The proposed greedy placement algorithm returns a solution with $|\mathcal{L}_g| = 3$ landmarks, i.e. green dots in Figure 4.6-(a). On the contrary, for the constrained scenario, the greedy solution places $|\mathcal{L}_g| = 6$ landmarks with bounds $M_b = 6$ and $m_b = 5$.

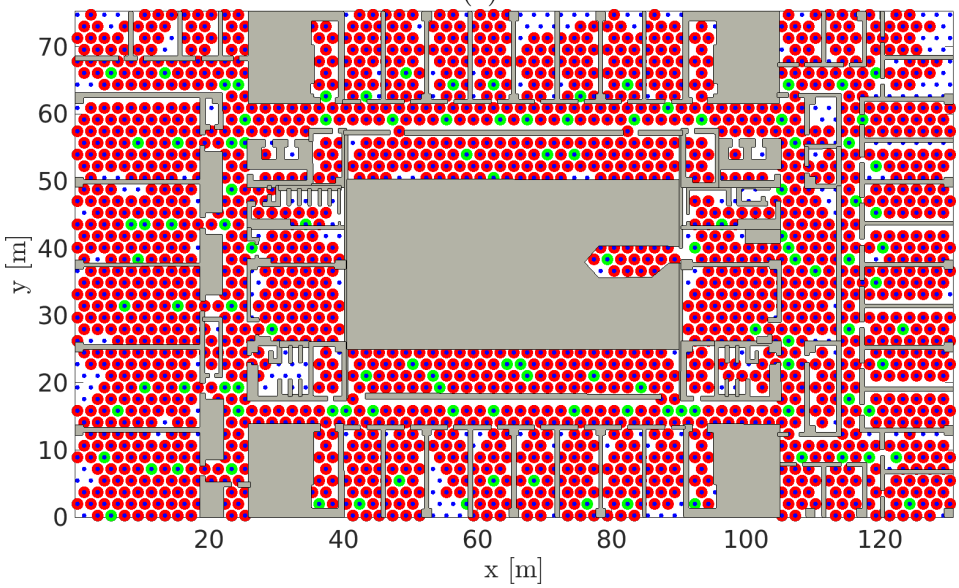
For the empirical validation of both the placements of Figure 4.6, we simulate 200 additional and independent paths considering multiple persons moving simultaneously in the corridor. This way, we emulates a typical human motion inside a populated environment, where its trajectory generation strategy depends on its personal behaviour and it is influenced by other human beings in the same ambient, hence no knowledge of the trajectory is available upfront. In both cases, the localization accuracy based on the greedy placement meets the given uncertainty constraint $\xi(\boldsymbol{p_k})$ with 99.5% probability (i.e., only one path violate once the threshold).

4.4.2 Real trajectories

As a further validation of the proposed solution in a context similar to the applicative scenario of the ACANTO project [2], 360 paths captured at the entrance of the ETH Zurich building (see Figure 4.7-(a)) have been used to test the performance of landmark greedy placement [100]. Again, the applicability of model (4.1) is substantiated by [44]. Hence, we can safely assume that each user drives a *FriWalk*. Figure 4.7-(b) shows 288 paths extracted from the video footage. With the SDA parameters defined in Table 4.2, we have $|\mathcal{L}_f| = 72$ possible landmark locations (blue dots in Figure 4.7-(c)). In this case, $M_b = 10$ and $m_b = 3.7$, respectively. The greedy algorithm selects $|\mathcal{L}_g| = 5$ landmarks. Using the remaining 72 paths of the available data set, we found the uncertainty constraint $\xi(\boldsymbol{p}_k)$ is met with 98.5% probability. Consider that the larger the SDA, the lower $|\mathcal{L}_f|$ and the more $|\mathcal{L}_g| \to |\mathcal{L}_f|$. For instance, Figure 4.7-(d) reports the placement results when the SDA range is two times larger than in the previous cases (i.e. $R = 8$ m). In this case, $|\mathcal{L}_f| = 18$, $M_b = m_b = 3$, and the solution of the greedy algorithm converges to $|\mathcal{L}_g| = 3$, as well. Moreover, all the remaining 72 paths meet the uncertainty constraint $\xi(\boldsymbol{p}_k)$. Similar results can be achieved if the growth rate of dead reckoning uncertainty increases. This behavior suggests that the solutions of greedy and naive random placements become closer and closer (as shown in Figure 4.5), depending on the ratio between the SDA dimension and the growth rate of dead reckoning uncertainty.



(a)



(b)

Figure 4.4: DISI scenario for vehicle trajectories generated with the chosen planner for robots. (a) 800 paths considered for the landmark placement. (b) potential landmarks locations (dots), landmarks detected from at least one trajectory (circled dots) and landmark deployment with the greedy algorithm (green circled dots).

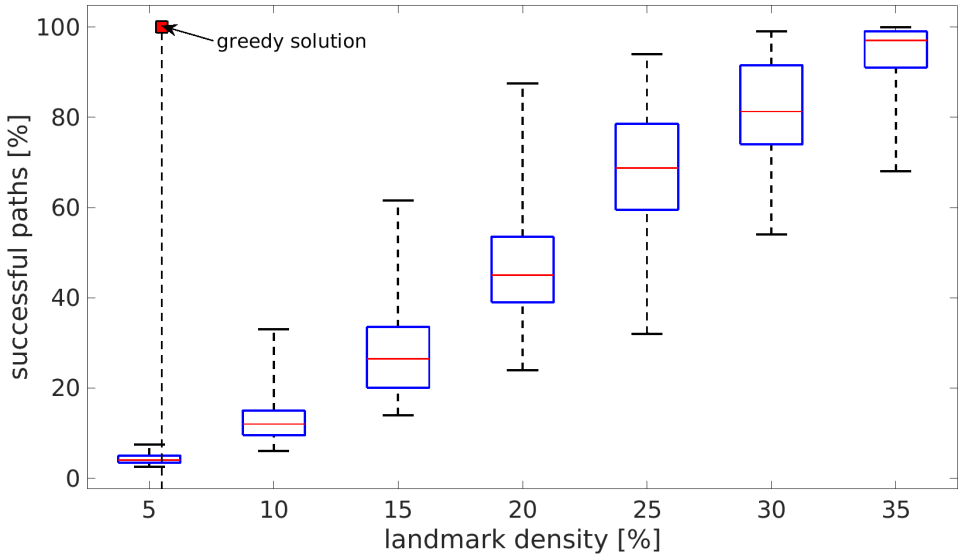


Figure 4.5: Percentage of path satisfying the maximum uncertainty limit $\xi(\boldsymbol{p}_k)$ (vertical axis) against the percentage of landmarks randomly placed with respect to $|\mathcal{L}_f|$ (horizontal axis) for the DISI scenario reported in Figure 4.4. The vertical thick line corresponds to the greedy solution, while the square on top recall that no path violates $\xi(\boldsymbol{p}_k)$.

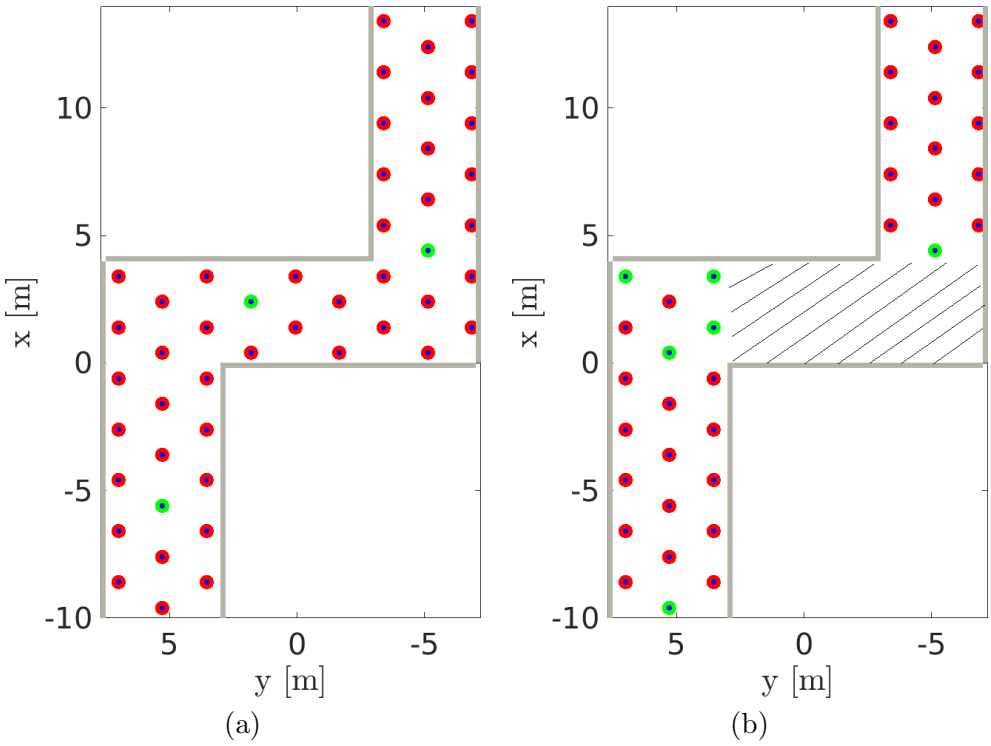


Figure 4.6: Corridor scenario for trajectories generated with the HSFM [44]. Landmarks locations (dots), landmarks locations detected from at least one trajectory (circled dots) and landmark deployment with the greedy algorithm (green circled dots) are reported. (a) Deployment without placement constraints and (b) deployment considering an area where the landmarks cannot be placed.

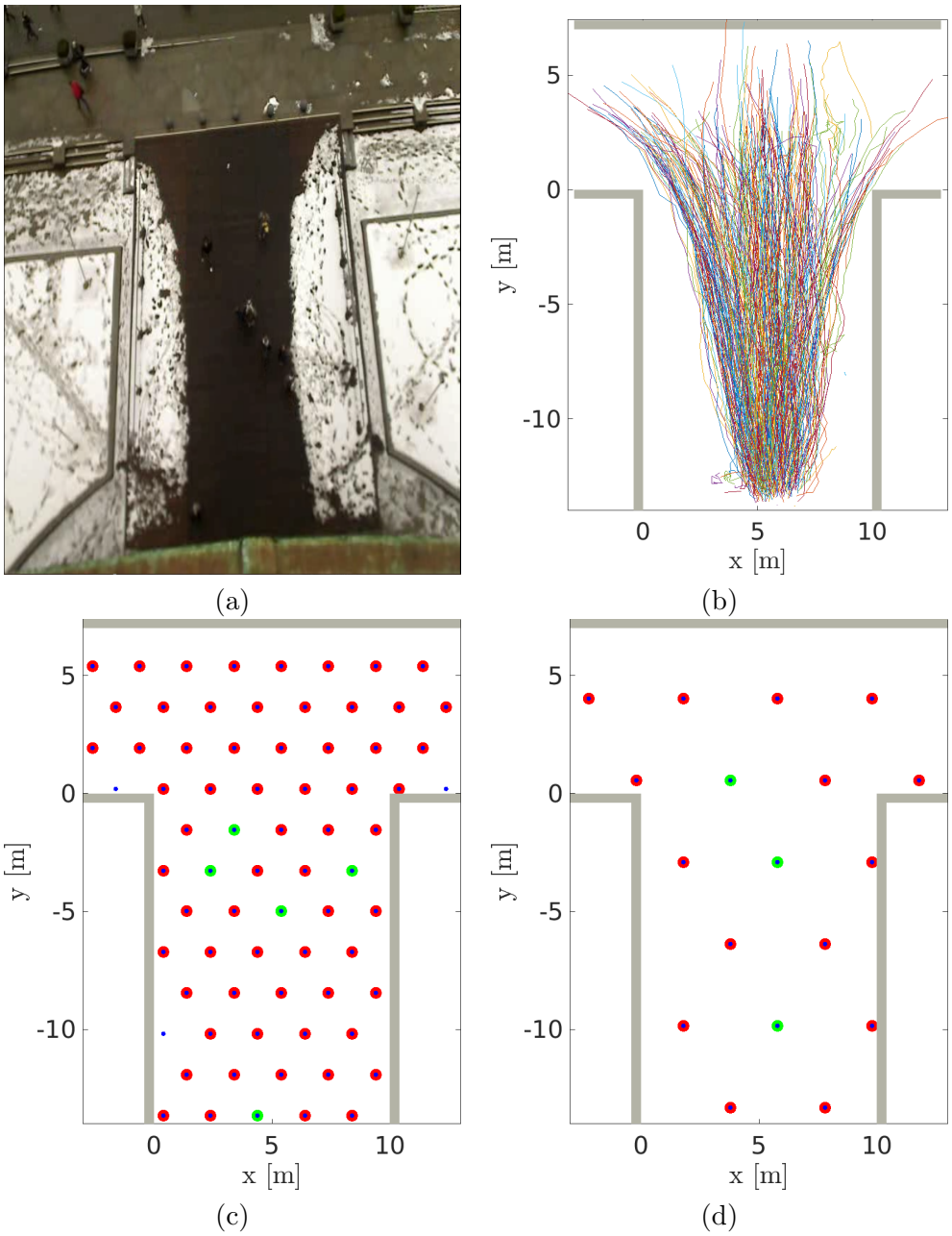


Figure 4.7: Simulation on actual data. (a) ETH Zurich building entrance. (b) measured paths [100]. (c) deployment of 5 landmarks for the greedy algorithm. (d) deployment of 3 landmarks for the greedy algorithm when the SDA range doubles.

4.5 Experimental validation

4.5.1 Experimental Setup

As mentioned in Chapter 1 the platform used to validate experimentally the proposed approach is the *FriWalk*. The robot can estimate its own speed, enabling odometric trajectory estimation, through two encoders AMT-102V mounted on rear wheels with a resolution of 0.08 mrad per tick. In addition, the relative pose of the camera with respect to the Aruco code detected in the camera field of view (namely the SDA in the case at hand) can be measured by using a front RGB camera (PLAYSTATION Eye) and software application based on OpenCv 3.1.0.

Encoder information is collected by a BeagleBone Black board via a Controller Area Network (CAN) bus. The BeagleBone Black board processes encoder data and sends odometry results to an Intel NUC mini PC (equipped with a microprocessor i7-5557 and 8 GB of DDR3 RAM) through Local Area Network (LAN) router. The PLAYSTATION Eye is connected directly to the Intel NUC mini PC through a USB link. The NUC mini PC is in turn also connected to the LAN router. The router provides Wi-Fi connectivity between the *FriWalk* and an external PC used for telemetry, e.g. to log the encoder measurement data and the relative position and orientation measures with respect to every detected Aruco code while the robot is moving. Accuracy and precision of the linear and angular velocity estimates v_k and ω_k based on odometry were evaluated by comparing the values returned by the BeagleBone Black board with those obtained by differentiating the position and orientation measured by an OptiTrack reference localization system. In all experiments, *FriWalk* and OptiTrack data were properly aligned in time. Moreover, the robot was driven repeatedly (i.e. about 50 times) and at a different speed (ranging from 0.3 m/s to 1.2 m/s) over an eight-shaped path. The OptiTrack localization system consists of 14 calibrated cameras and it able to measure the position of ad-hoc reflective markers attached to the *FriWalk* with standard uncertainty of about 1 mm, i.e. negligible compared with the positioning uncertainty based on odometry. The linear and angular velocity values computed by the BeagleBone Black platform using the encoder data were aligned in time and compared with those estimated from the sequence of positions measured by the OptiTrack. The histograms of the differences ϵ_k^v and ϵ_k^ω between the linear and angular velocity data, respectively, resulting from odometry and the OptiTrack-based localization system are shown in Fig. 4.8. Observe that the mean values of ϵ_k^v and ϵ_k^ω are negligible, while the elements of the covariance matrix E are reported in Tab. 4.4. Such values, although apparently small, have a significant impact on odometry-based positioning uncertainty since they tend to accumulate over time due to dead reckoning.

The OptiTrack reference localization system was also used to evaluate accuracy and precision of distance and orientation measurements based on the PLAYSTATION Eye, whenever an Aruco code is detected. Again, the *FriWalk* was driven repeatedly over an eight-shaped path. The histograms of the differences between

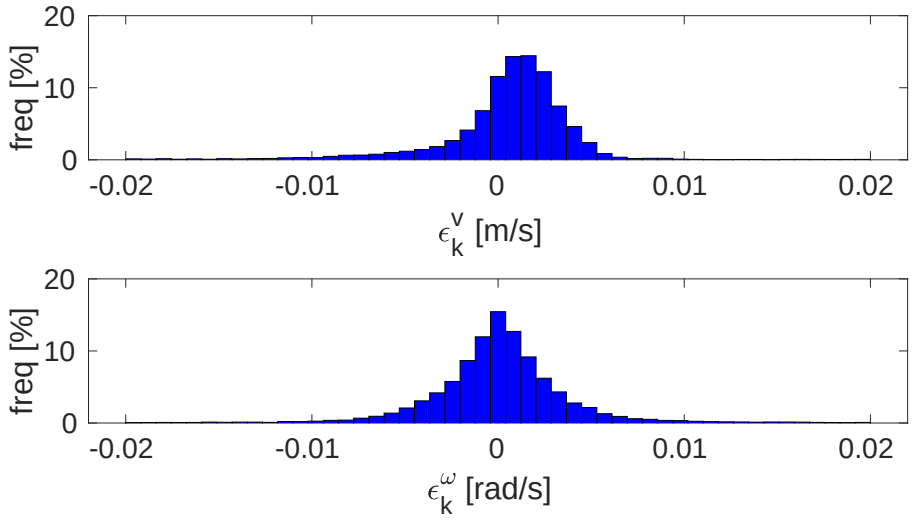


Figure 4.8: Experimental distribution of the linear velocity estimation error ϵ_k^v and of the angular velocity estimation error ϵ_k^ω due to encoder data.

the position and orientation values measured by the on-board vision system and those obtained with the OptiTrack are shown in Figure 4.9(a). Such differences are realizations of the components of the random vector η_k in (4.1), whose covariance matrix is (4.3). Observe that the mean values of the elements of $\boldsymbol{\eta}_k$ are -7.8 mm, -8.3 mm and 10 mrad, respectively, and can be easily compensated, thus obtaining a zero mean process, as assumed in Section 4.1.1. The corresponding standard uncertainty values (i.e., about 58 mm, 54 mm and 34 mrad) are considered adequate for the intended application. The positions of the landmarks detected in repeated trials (represented with about 20000 dots in Figure 4.9(b)) was also used to estimate the SDA of the PLAYSTATION Eye installed on the *FriWalk*. In particular, the SDA exhibits approximately a trapezoidal shape and the values of parameters r, R and α shown in Fig. 4.2 are summarized in Tab. 4.4. The same Table reports also the values of the elements of covariance matrices E and N, the sampling period t_s and the target uncertainty $\xi(\boldsymbol{p}_k)$ for landmark placement. For the sake of simplicity (but without loss of generality) in the rest of this Chapter $\xi(\boldsymbol{p}_k)$ is assumed to be constant, i.e. equal to 0.8 m regardless of actual *FriWalk* position. This value is just an example, but it is reasonable for the purposes of project ACANTO.

4.5.2 Experimental Results

The placement reported in Subsection 4.4.1 is based on the assumption that all points of every DISI room are fully accessible. However, in practice this is not true, due to obvious privacy or security issues. Therefore, to plan a fair and appropriate

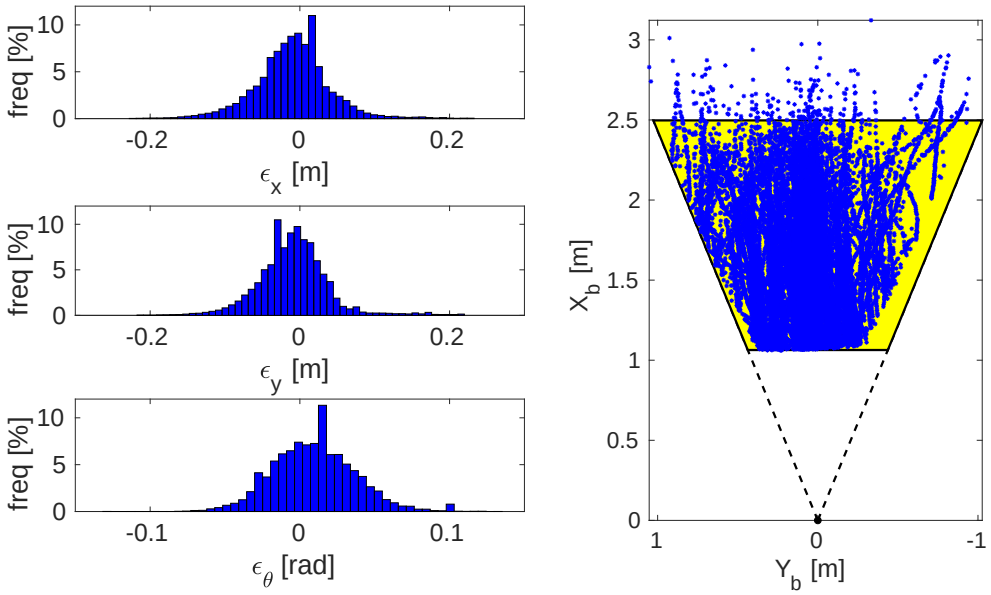
Table 4.4: Parameters of the *FriWalk*

Symbol	Value	Unit	Symbol	Value	Unit
$\xi(p_k)$	0.8	m	t_s	0.2	s
r	1.15	m	R	2.70	m
α	0.78	rad	$\sigma_{c_y^2}$	30	cm^2
$\sigma_{c_x}^2$	34	cm^2	$\sigma_{c_{xy}}$	-2	mm^2
$\sigma_{c_\theta}^2$	0.001	rad^2	σ_v^2	20	mm^2/s^2
$\sigma_{c_{x\theta}}$	-1	mm·mrad	σ_ω^2	20	mrad^2/s^2
$\sigma_{c_{y\theta}}$	-8	mm·mrad	$\sigma_{v\omega}$	2	$\text{mrad}\cdot\text{mm}/\text{s}^2$

experimental validation, the greedy placement algorithm was applied again considering a subset of all possible paths, i.e., limiting the analysis just to the rooms that are fully accessible. The results of this new landmark placement (assuming again that $\xi(\boldsymbol{p}_k) = 0.8$ m) are shown in Fig. 4.10, along with a snapshot of the actual setup in a corridor (i.e., white circle markers. The total number of landmarks deployed is 29 ($|\mathcal{L}_g| = 29$).

With the Aruco codes deployed as shown in Fig. 4.10, 10 users were asked to walk freely in the floor and the collected data cover a total distance of about 4 km. Note that the users were not informed that the Aruco codes deployed in the environment were necessary to the robot localization. The *FriWalk* position was estimated by the non-Bayesian algorithm (shortly referred in the following as NBE) described in Section 4.1.3. The corresponding estimated paths are shown in Fig. 4.10. We recall that, in this case, the estimated paths follow the discrete dynamic based on the input matrix (4.2) when only the encoders are available. Instead, whenever a landmark is detected, the position is restored to $h(\boldsymbol{p}_k^m)$ (defined in Section 4.1.2), with uncertainty N, where the superscript m stands for measured. Observe that, due to the sporadic nature of landmark detection, the estimated paths may exhibit sudden and visible changes if a landmark is detected after a quite long time, i.e., when the uncertainty due to dead reckoning becomes particularly large. In principle, such sudden large errors should be smaller if a Bayesian estimator fusing odometry and vision system data (e.g. the Extended Kalman Filter - EKF - described in the background chapter) were used.

To verify whether the uncertainty constraint is met, the positioning uncertainty has to be reconstructed from the differences e_x and e_y between the $x-$ and $y-$axis coordinates estimated by the *FriWalk* and those of some reference points, e.g., anytime one landmark code is detected. Unfortunately, no continuous position tracking is possible in DISI premises, since the Optitrack reference system cannot be used in



(a) (b)

Figure 4.9: Estimation of the camera parameters. (a) Error histograms of the camera reading pose. (b) Trapezoidal approximation of the SDA: the dots represent the relative measured positions of the Aruco codes with respect to the camera, while the shaded trapezoid is the estimated SDA.

such a large environment. In what follows, we present an experimental validation of the proposed placement solution based on the non-Bayesian estimator, named from this point on NBE. We verify contextually that the target uncertainty limit $\xi(\boldsymbol{p}_k)$ in Table 4.4 is respected and that the NBE is conservative. Fig. 4.11 reports 365 position estimation errors e_x and e_y associated with the NBE (empty circles) and the EKF (filled circles) immediately before landmark detection. The ellipses corresponding to the covariance matrices of either cloud of points are also plotted. In both cases the ellipses (as well as the possible circles circumscribing them) are safely included within the wanted uncertainty region, namely the dash-dotted circle of radius $\xi(p_k)$ shown in Fig. 4.11 and, as expected, the positioning uncertainty associated with the EKF is smaller than the NBE one, due to the Bayesian nature of the former approach. Indeed, just about 10% of position error values lie outside that circle, i.e., less than 33% that we would expect in a perfectly Gaussian case.

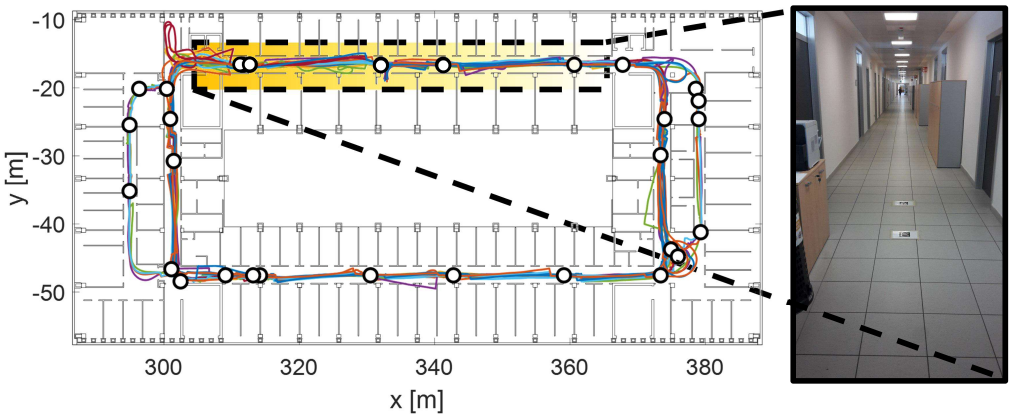


Figure 4.10: Paths followed by the *FriWalk* for experimental validation. The white circle markers represent the locations where the Aruco codes are actually placed, i.e., the elements of set $\mathcal{L}_g$. The picture on the right shows a snapshot of the actual landmark layout in a corridor.

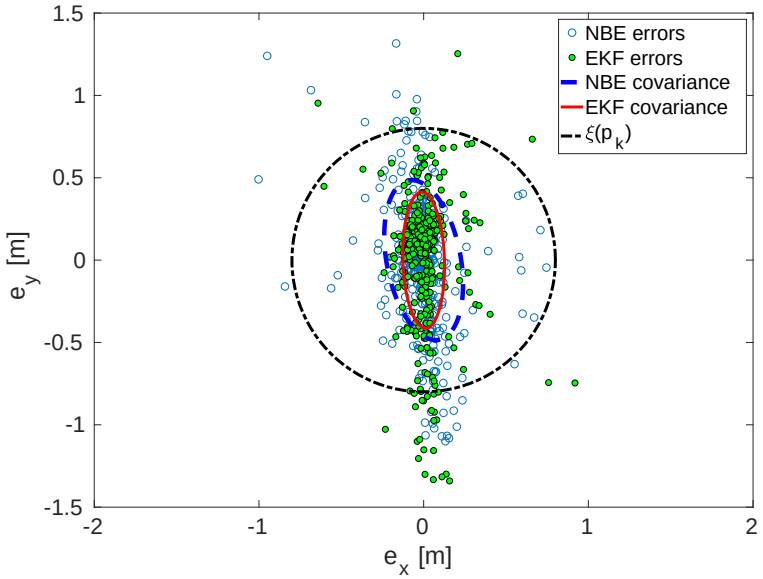


Figure 4.11: Scatter diagrams of the position estimation errors e_x and e_y associated with the NBE (empty circles) and the EKF (filled circles). In the same graph, the ellipses corresponding to covariance matrices $P_k^{x,y}$ given by (4.6) for the NBE (dashed line) and for the EKF (solid line) are also shown for the sake of comparison with the circle of radius $\xi(\boldsymbol{p_k})$ delimiting the target uncertainty region (dash-dotted line).

4.6 Extensions

In this section we show how the technique presented can be easily extended to more complex Baesyan localization systems that handle information from heterogeneous sensors. A frontal RGB-D camera (namely a Astra Orbbec[1]) is added at the localization system to detect environmental features, such as walls. The front Astra Orbbec detects and measures the distance from possible fixed obstacles, e.g. walls, when they are within its own SDA, which here is modeled to have the same shape of the camera SDA. The map of the indoor environment, including the walls, is assumed to be given.

In addition the analysis is extended to accommodate a non negligible landmark occlusion probability. Supposing the detection of a landmark to be a boolean stochastic variable uncorrelated from the others, it is possible to augment the deployment of artificial references to handle this situation.

In the following subsections first we present how the informations coming from the RGB-D sensor are integrated (Subsection 4.6.1); than in Subsection 4.6.2 a process to make the deployment robust to occlusion is presented and finally in Subsection 4.6.3 the concepts are applied in simulation and some preliminary result are drawn.

4.6.1 RGB-D camera model

When the Orbbec detects a wall, a segment parallel to the plane of motion delimited by points A and B can be measured, as shown in Figure 4.2. If (x_a, y_a) and (x_b, y_b) are the coordinates of A and B in the reference frame of the Orbbec, the angle and the distance between the Orbbec and point A are given by $\theta_a = \arctan(y_a/x_a)$, and $\rho_a = \sqrt{x_a^2 + y_a^2}$, respectively. As a consequence, the function modeling the measurements performed by the Orbbec is $\boldsymbol{z}_k^r = [\rho_k, \theta_{ab_k}] + \boldsymbol{\eta}_k^r$, where

$$\begin{aligned} \rho_k =& \sqrt{[x_p - x_k - x_r\cos(\theta_k) - y_r\sin(\theta_k)]^2 + \\ & \overline{[y_p - y_k - x_r\sin(\theta_k) - y_r\cos(\theta_k)]^2}, \\ \theta_{ab_k} =& \theta_k + \theta_r, \end{aligned} \tag{4.7}$$

(x_r, y_r) is the position of the Orbbec camera with respect to (x_k, y_k), θ_r is orientation of the Orbbec, (x_p, y_p) is one generic point belonging to the detected wall and available in the map $\mathcal{M}$ of the environment, and $\boldsymbol{\eta}_k^r$ denotes the measurement uncertainty vector. Again, this is a bivariate and normally-distributed random vector, with zero mean and covariance matrix $N^r = \text{diag}(\sigma_{\rho_r}^2, \sigma_{\theta_r}^2)$.

To clarify the role of the Orbbec camera, consider the qualitative straight path shown in Figure 4.13. Let us assume that the initial standard uncertainty of the *FriWalk* along axes X_w and Y_w is $\sigma_x = \sigma_y = 0.8$ m. If the position of the *FriWalk*

[1] https://orbbec3d.com/product-astra/

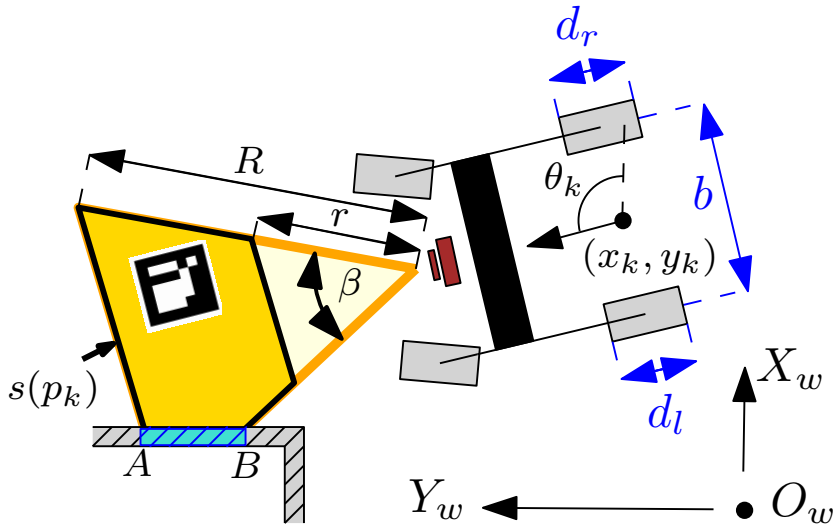


Figure 4.12: The *FriWalk* schematic representation and SDA.

were estimated only through dead-reckoning (i.e. using just encoder data), the initial positioning uncertainty would grow unboundedly. If instead the RGB-D camera detects the walls of the corridor, the distance and orientation measurements given by (4.7) can be used to reduce σ_y, e.g. by using an Extended Kalman Filter (EKF) based on model (4.1) [92]. Moreover, as the *FriWalk* keeps on moving along the corridor, the localization uncertainty can be further reduced along both X_w and Y_w, due to the larger amount of measurement data used to update the estimated state of (4.1).

Of course, the general approach above can be extended by including the measurements extracted when a landmark is detected that in this section will be defined as $\boldsymbol{z}_k^c$ to differentiate them from the Orbbec measurements $\boldsymbol{z}_k^r$. In this case, with reference to (4.1), a full vector of measurement data $\boldsymbol{z}_k = [\boldsymbol{z}_k^c, \boldsymbol{z}_k^r]^T$ is collected at time kt_s and the corresponding overall measurement uncertainty vector $\boldsymbol{\eta}_k = [\boldsymbol{\eta}_k^c, \boldsymbol{\eta}_k^r]^T$ has covariance matrix $N = \text{diag}(N^c, N^r)$, since the measurements performed by different sensors can be reasonably assumed to be uncorrelated. Of course, the presence of $\boldsymbol{z}_k^c$ and $\boldsymbol{z}_k^r$ in $\boldsymbol{z}_k$ as well as of $\boldsymbol{\eta}_k^c$ and $\boldsymbol{\eta}_k^r$ in $\boldsymbol{\eta}_k$ is inherently intermittent, as it depends on whether a landmark and/or a wall is detected at time kt_s.

4.6.2 Probability of Missed Aruco Code Detection

Let ς_f be the non-zero probability of not detecting a landmark. In such conditions, $u_p(P_k) \leq \xi(\boldsymbol{p}_k)$ can then be guaranteed only in a stochastic sense, that is with a probability greater than a defined threshold ς_d. Being n the number of landmarks that cover each clause, the constraint is then violated if the sensor misses a number of visual landmarks greater than or equal to n. Let m_i be the number of landmarks

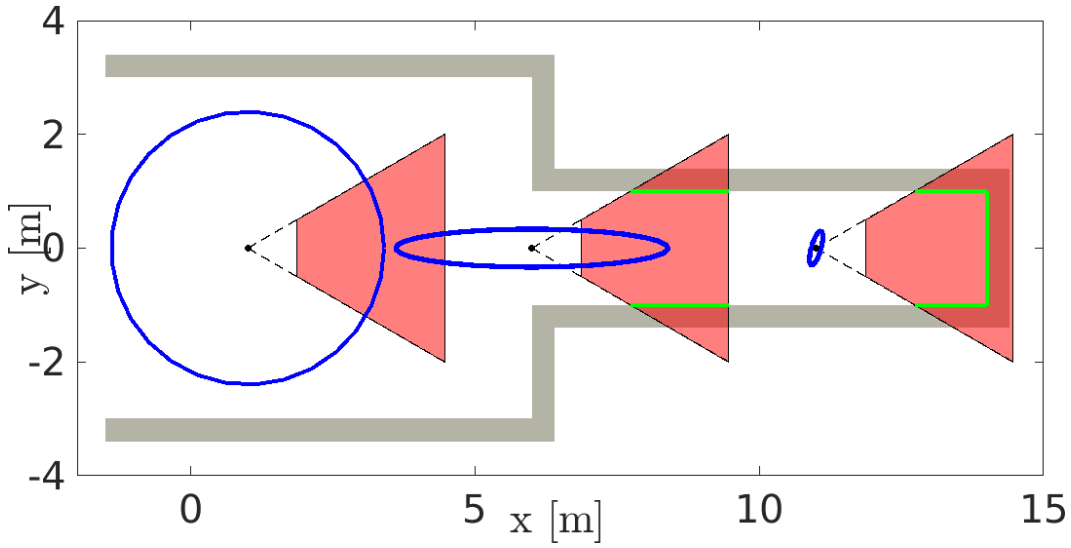


Figure 4.13: Example of uncertainty reduction due to the fusion of encoder and RGB-D measurement data. As the amount of information increases, the positioning uncertainty tends to decrease.

along the i–th trajectory T_i, the probability of missing n consecutive landmarks is given by the following recursive formula:

$$\zeta(m_i, n) = \begin{cases} 0 & \text{if } n > m_i \\ \varsigma_f^n + \sum_{j=1}^{n_i} (1-\varsigma_f)\varsigma_f^{j-1}\, \zeta(m_i - j, n) & \text{otherwise.} \end{cases}$$

Of course, such a probability increases with m_i and decreases if n grows.

Extending the approach presented in the subsection 4.3.3 to guarantee a minimum of n landmark per clause is simple. It is sufficient to link each row with a counter, say $n_{i,j}$, initialized to n. Every time a column is removed the counters corresponding to the rows covered by that column are decremented. If after the counters are decremented some of them evaluate to 0 the corresponding lines are removed from the matrix since that clause got covered n times. An important remark is that in order to find the appropriate value of n the problem should be iteratively solved. In other words, it is required to call the greedy solver iteratively increasing n until we found the smaller value of n such that

$$1 - \zeta(m_i, n) > \varsigma_d \tag{4.8}$$

is satisfied for all the trajectories T_i. This is necessary because m_i used to compute $\zeta(m_i, n)$ intrinsically depends on n: if we require more landmark to cover each clause (i.e., we increase n) the number of landmark deployed for each path (i.e., m_i) will increase.

Table 4.5: Parameter values used for simulations of the system with RGBD camera.

Symbol	Value	Unit
$\xi(p_k)$	0.8	m
r	0.8	m
α	$\pi/3$	rad
$\sigma^2_{c_x}$	34	cm^2
$\sigma^2_{c_\theta}$	0.001	rad^2
$\sigma_{c_{x\theta}}$	-1	mm·mrad
$\sigma_{c_{y\theta}}$	-8	mm·mrad
σ_{ρ_r}	50	mm
ς_f	0.4	

Symbol	Value	Unit
t_s	0.2	s
R	2.50	m
$\sigma_{c_y^2}$	30	cm^2
$\sigma_{c_{xy}}$	-2	mm^2
σ^2_v	20	mm^2/s^2
σ^2_ω	20	mrad^2/s^2
$\sigma_{v\omega}$	2	mrad·mm/s^2
σ_{θ_r}	0.17	rad
ν_d	0.95	

4.6.3 Simulation Results

This section presents the simulation results of two different scenarios based on the model of *FriWalk* with the parameters reported in Table 4.5.

Demo scenarios

The first scenario is conceived to highlight the features of the proposed greedy placement and consists of a 4×70 m corridor. Globally, 100 straight paths in both directions and uniformly spaced have been generated in the corridor. The possible landmark locations are 430 and result from the solution of [91] using the values of R, r and α reported in Table 4.5. Figure 4.14 shows the results of landmark placement with (red dots) and without (green dots) using the front RGB-D camera for $n = 1$ (a) and 5 (b). Notice how the use of contextual information (namely the detection of walls) drives the algorithm to place the landmarks in the center of the corridor, where the RGB-D camera cannot collect data. Another interesting result is shown in Figure 4.15, which represents the box-and-whiskers diagrams, computed over all the generated paths, of the probabilities $\zeta(m_i, n)$ of not detecting a landmark for different values of n with (a) and without (b) using the front RGB-D camera. The larger variability when the RGB-D camera is used (see Figure 4.15-a) (especially if $n = 1$) depends on the particular path of the agent. When the agent moves in the middle of the corridor the walls are not detected. Hence the uncertainty constraint $\xi(\boldsymbol{p}_k)$ is more likely to be violated. This is not the case for the paths detecting the walls. Of course, this phenomenon does not occur if the RGB-D camera is not used, as shown in Figure 4.15-b. Notice that, in both cases, to meet the desired stochastic

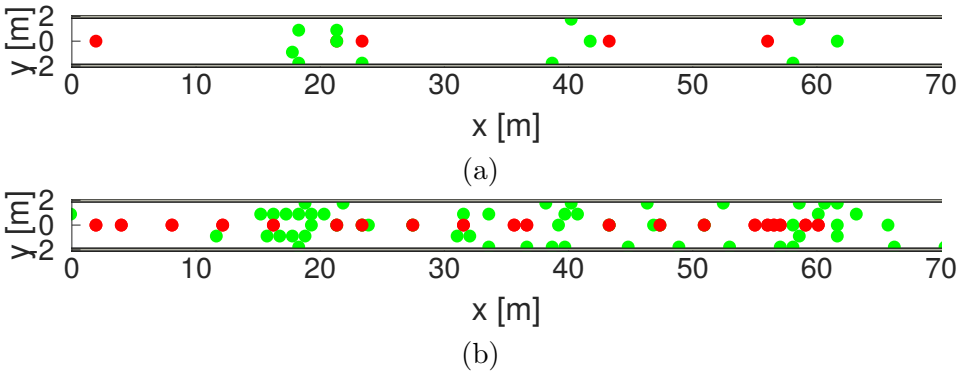


Figure 4.14: Landmark placement in a corridor with straight paths and with (red dots) or without (green dots) the RGB-D camera for (a) $n = 1$ and (b) $n = 5$.

Table 4.6: Number of landmarks placed in the corridor shown in Figure 4.14 with and without using the front RGB-D camera.

	Cover Counter n						
	1	2	3	4	5	6	7
no RGB-D	12	24	36	45	57	66	78
RGB-D	4	9	14	18	22	27	32

threshold of $\nu_d = 0.95$, than at least $n = 5$ landmarks are needed. However, from Table 4.6 it is evident that the number of landmarks to be placed is almost three times larger.

Realistic Environment

The realistic environment chosen for simulation purposes is the Department of Information Engineering and Computer Science (DISI) of the University of Trento. In this case, 1000 random trajectories have been generated starting and ending in different rooms. The paths are quite regular, as can be expected in a highly constrained indoor environment. The dots in Figure 4.16(a) represent the 7363 possible landmark positions obtained using the geometric approach described in [91]. Assuming that the initial positions are known with minimal uncertainty and that the Aruco codes are always detected (i.e. $\varsigma_f = 0$), the result of the placement based on the greedy algorithm is shown in Figure 4.16(b) for $n = 1$. Again, red and green dots are obtained including or excluding the data from the RGB-D camera, respectively. If $\varsigma_f = 0.4$ and $\nu_d = 0.95$, at least $n = 5$ landmarks per clause are needed to satisfy (4.8) for all the trajectories, resulting in the deployment shown in Figure 4.16(c). Notice that the landmarks are placed mainly along the corridors, since the density of possible paths is higher than in the rooms. Moreover, when the RGB-D camera is used, the number of landmark clusters as well as their size are

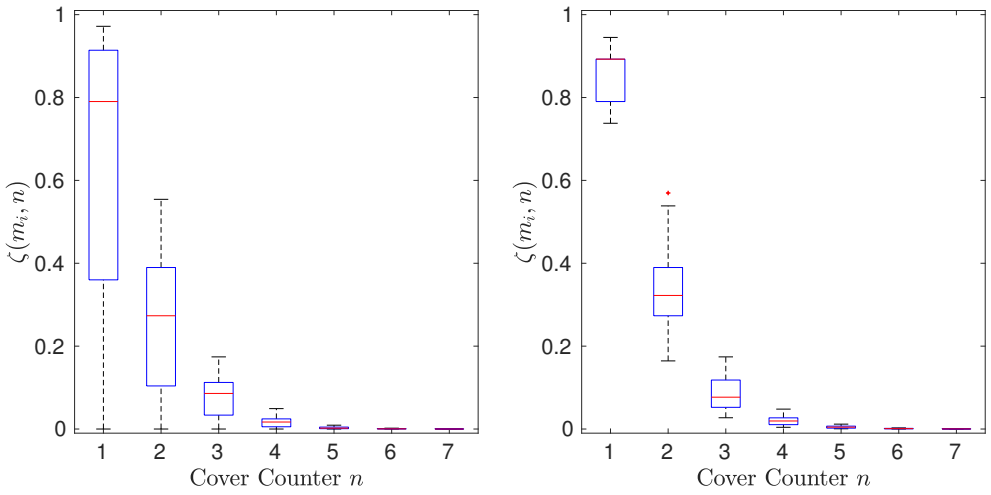


Figure 4.15: Box-and-whiskers plots of the probabilities $\zeta(m_i, n)$ of not detecting a landmark along the corridor shown in Figure 4.14 with (a) and without (b) using the front RGB-D camera.

Table 4.7: Number of landmarks placed with and without using the RGB-D camera in the realistic case shown in Fig. 4.16.

	Cover Counter n						
	1	2	3	4	5	6	7
no RGB-D	53	100	155	203	256	310	360
RGB-D	33	67	98	129	158	188	216

smaller. This is confirmed by the results in Table 4.7, which reports the amount of landmarks actually deployed. The results on the probability of failure $\zeta(m_i, n)$ are not reported because they are similar to those shown in Figure 4.15. As a final comment, it is worth emphasizing that, even if the number of trajectories and potential landmark locations is quite large, the computation time of the greedy placement algorithm implemented in Matlab and running on a 3.50 GHz Intel Core i7 with 8 GB of RAM is about 45 minutes.

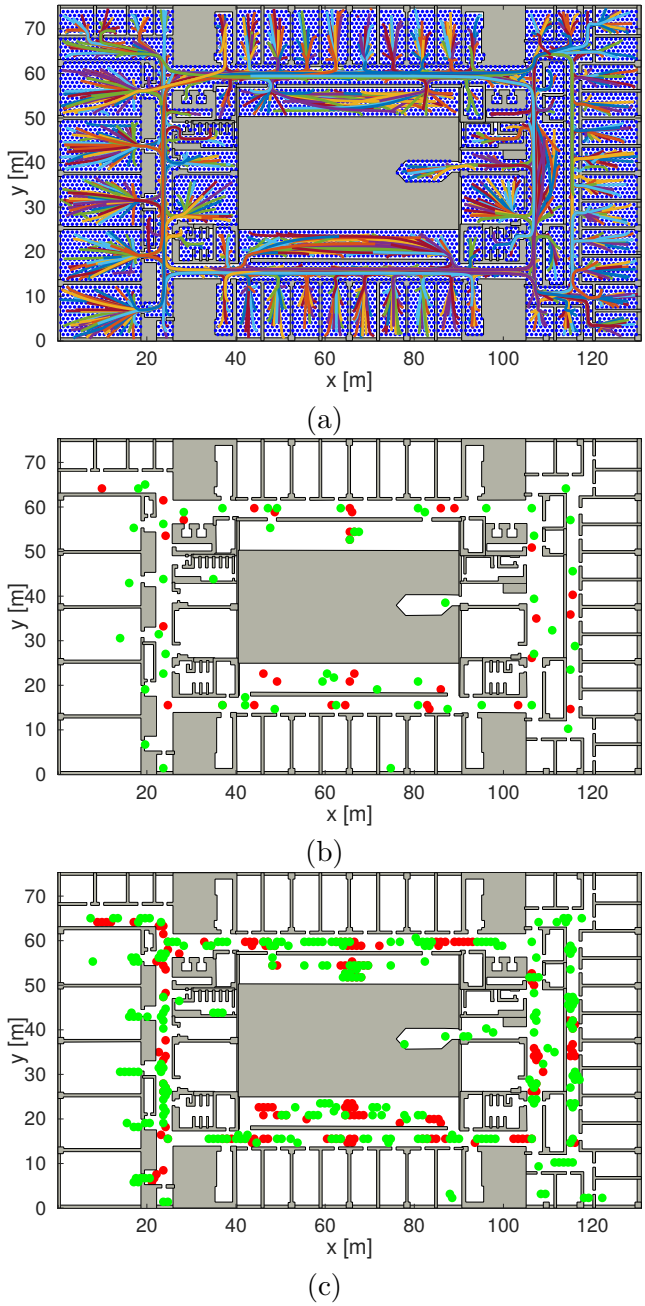


Figure 4.16: Simulation results in a realistic scenario. In (a) the *FriWalk* trajectories generated by a path planner are shown on the DISI map. In (b) and (c) the results of landmark placement with (red dots) and without (green dots) using the RGB-D camera are plotted for $n = 1$ with $\varsigma_f = 0$ and $n = 5$ with $\varsigma_f = 0.4$, respectively.

Chapter 5

Probabilistic authority-sharing

The knowledge of estimated pose probability distribution is important, but often in the literature only the mean value is used. Relying on a localization system with an accurate and quite homogeneous error distribution, often the control systems presented in the literature ignore the uncertainty of the pose estimation. Usually the controllers are designed and optimized considering a particular level of pose uncertainty and do not give any guarantee for situation where the uncertainty level is fluctuating. In this chapter we consider the problem of guiding a senior user along a path using a robotic walking assistant. This is a particular type of path following problem, for which most of the solutions available in the literature require an exact localization of the robot in the environment. This is not always the case especially in featureless area of the environment. Our key observation is that the intervention of the system (and a good level of accuracy) is only required in proximity of difficult decision points, while we can rely on the user in an environment where the only possibility is just to maintain a course (e.g., a corridor). The direct implication is that we can instrument the environment with a heavy infrastructure only in certain areas. This design strategy has to be complemented by an adequate control law that shifts the authority (i.e., the control of the actuators) between the robot and the user according to the accuracy of the information available to the robot. Such a control law is exactly the contribution of this Chapter. The proposed authority-sharing controller is extensively analyzed in simulations and validated experimentally.

5.1 Problem definition and background material

The *FriWalk* is modelled as a unicycle-like vehicle, as depicted in Figure 5.1. Let (x, y) be the coordinates of the mid point of the rear axle in a world reference frame $\langle W \rangle = \{O_w, X_w, Y_w, Z_w\}$ and θ the vehicle yaw, i.e. the orientation of the vehicle-attached reference frame $\langle M \rangle = \{O_m, X_m, Y_m, Z_m\}$ with respect to the world frame (see Figure 5.1). The differential kinematic model with respect to the state variables

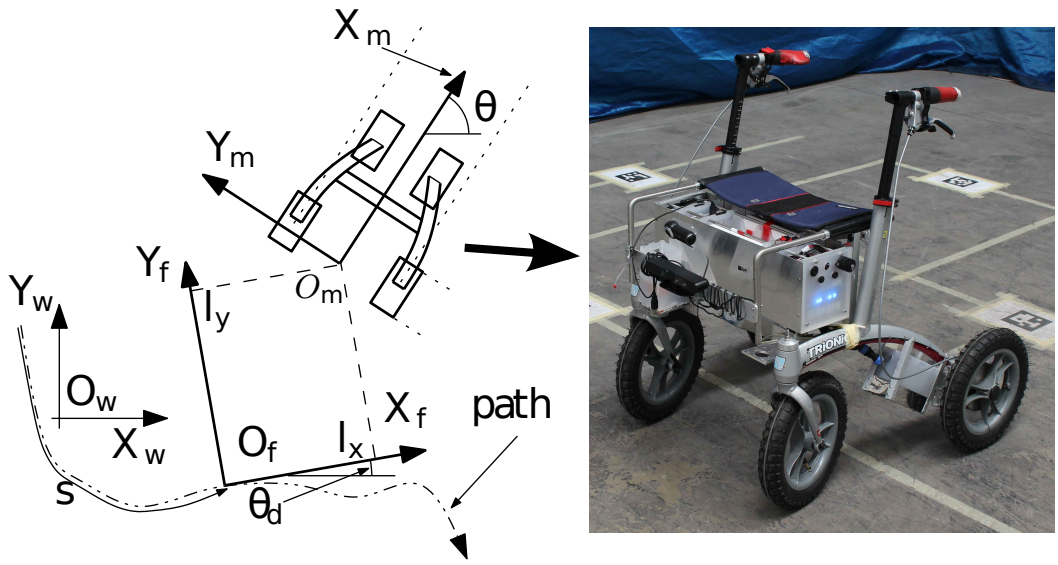


Figure 5.1: Reference frames and corresponding vehicle coordinates.

$\chi^\star = [x, y, \theta]^T$ is given by

$$\begin{cases} \dot{x} &= v\cos\theta, \\ \dot{y} &= v\sin\theta, \\ \dot{\theta} &= \omega, \end{cases} \tag{5.1}$$

where v and ω are the forward and the angular velocity of the vehicle, respectively. In passive robotics, the forward velocity is typically selected or imposed by the user and represents in this context the human-robot physical interaction. The angular velocity is considered as a control input, and it is applied using actuation systems such has front steering wheels [7], brakes [9, 46] or rear motors [10] (which can be also applied once the vehicle dynamic is considered). The path following problem is described by adopting a Frenet frame $\langle F\rangle = \{O_f, X_f, Y_f, Z_f\}$ moving along the path and defining the curvilinear abscissa s (see Figure 5.1). The orientation of the Frenet frame (i.e. the desired attitude of the vehicle) is denoted by θ_d, while the vehicle reference point O_m has coordinates (l_x, l_y) in the Frenet frame $\langle F\rangle$. The orientation error is thus defined as $\tilde{\theta} = \theta - \theta_d$. Using this new set of coordinates $[l_x, l_y, \tilde{\theta}]$, the rollator differential kinematics (5.1) is rewritten as [116]

$$\begin{cases} \dot{l}_x &= -\dot{s}(1 - c(s)l_y) + v\cos\tilde{\theta}, \\ \dot{l}_y &= -c(s)\,\dot{s}\,l_x + v\sin\tilde{\theta}, \\ \dot{\tilde{\theta}} &= \omega - c(s)\dot{s}, \end{cases} \tag{5.2}$$

where the path curvature is $c(s) = \frac{\mathrm{d}\,\theta_d}{\mathrm{d}\,s}$, and the velocity of the Frenet frame origin $\dot{s}$ is an auxiliary control input that can be freely chosen. The coordinates $\chi =$

$[l_x, l_y, \tilde{\theta}]^T$ are used to represent the path following problem as an asymptotic stability problem. The path is indeed approached and followed if

$$\lim_{t\to+\infty} |l_x(t)| = \lim_{t\to+\infty} |l_y(t)| = \lim_{t\to+\infty} |\tilde{\theta}(t)| = 0, \tag{5.3}$$

where t is the time. Notice that the path is assumed to be available upfront, e.g. using [32].

5.1.1 Path following

The solution to the path following problem is typically designed by supposing that the localisation algorithm is "accurate enough" to yield a negligible error on the estimate of the vehicle state χ. Of course, when intermittent observations are adopted, as in the localisation system running on the *FriWalk* and reported in [93], the effect of the feedback control can be highly wrong and, hence, the control should be given to the user. To implement this authority-sharing, how the localisation accuracy is derived and a description of the controller implemented is needed, which is the purpose of this section.

Vehicle localisation

Let us denote $\hat{a}$ the estimate of the quantity a and σ_a the corresponding standard deviation. With localisation algorithm we intend the execution of an estimator that provides "suitable estimates" of $\hat{x}, \hat{y}$ and $\hat{\theta}$ of the vehicle states of (5.1). For the rollator in Figure 5.1, the available sensors are encoders mounted on the rear wheels (odometry-based localisation) and a camera reading landmarks (QR codes placed on the floor, the ceiling or on the walls) whose positions in the map are known. The odometry data are always available but affected by dead-reckoning. The measures of vehicle position and attitude obtained by the landmarks are absolute but available only when a landmark is in the field of view of the camera. The two measures are fused using a Bayesian estimator, such as an Extended Kalman filter [14]. The estimator returns minimum variance estimates $\hat{x}, \hat{y}$ and $\hat{\theta}$ of the vehicle state and the corresponding estimation error covariance matrix

$$P = \mathrm{E}\left\{[x-\hat{x}, y-\hat{y}, \theta-\hat{\theta}]^T[x-\hat{x}, y-\hat{y}, \theta-\hat{\theta}]\right\}, \tag{5.4}$$

where $\mathrm{E}\{\cdot\}$ is the expected value operator.

Path following controller

It is a common practice in path following problems, to introduce a steering angle δ describing the manoeuvre that the vehicle should take to properly approach and follow the path, as sketched in Figure 5.2. For a vehicle moving forward (i.e. $\lim_{t\to\infty} v(t) > 0$), if $\delta(\cdot)$ is a continuous, strictly monotonic and odd function of

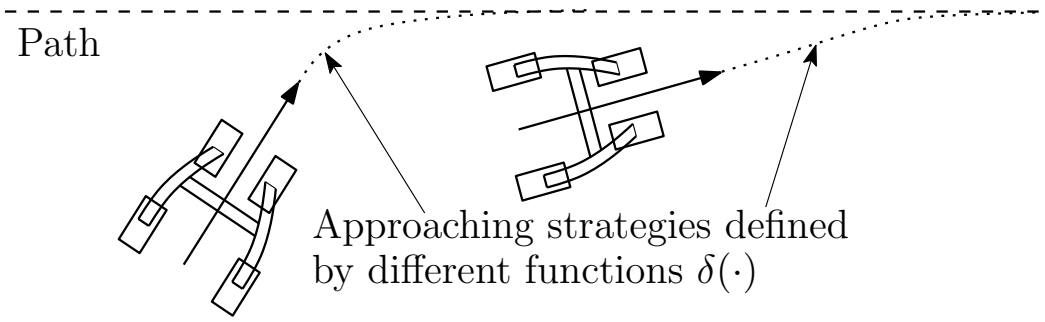


Figure 5.2: Examples of approaching manoeuvres.

l_y in (5.2) and satisfying $l_y\delta(l_y) \leq 0$ and $|\delta(l_y)| < \frac{\pi}{2}$, $\forall l_y$, and the velocity of the Frenet frame $\langle F \rangle$ is computed as $\dot{s} = v(\cos\tilde{\theta} + \kappa_x l_x)$, where $\kappa_x > 0$, it is sufficient to design a control input $\omega(\chi)$ that asymptotically drives to zero the attitude error $e_\theta = \tilde{\theta} - \delta(l_y)$ to solve the path following problem (5.3). A control law satisfying this requirement is

$$\begin{aligned} \omega(\chi) &= v\left[\gamma(\chi) - \kappa\left(\tilde{\theta} - \delta(l_y)\right)\right], \\ \gamma(\chi) &= c(s)(\cos(\tilde{\theta}) + \kappa_x l_x) + \\ &\quad + \left[-c(s)\left(\cos(\tilde{\theta}) + \kappa_x l_x\right) l_x + \sin(\tilde{\theta})\right] \delta'(l_y), \end{aligned} \tag{5.5}$$

where $\kappa > 0$ is a gain to be selected and $\delta'(l_y) \triangleq \dfrac{\mathrm{d}\delta}{\mathrm{d}l_y}(l_y)$. Indeed, using the Lyapunov function

$$V = \frac{1}{2}e_\theta^2, \tag{5.6}$$

its time derivative is

$$\begin{aligned} \dot{V} &= e_\theta \dot{e}_\theta = e_\theta\left(\omega - c(s)\dot{s} - \dot{l}_y\delta'(l_y)\right) = \\ &= e_\theta\left(\omega - v\gamma(\chi)\right) = -v\kappa e_\theta^2 < 0, \quad \forall e_\theta \neq 0, \end{aligned} \tag{5.7}$$

i.e. $e_\theta = 0$ is a uniformly globally asymptotically stable equilibrium.

5.2 Approach

The controller (5.5) ensures asymptotic tracking of the path in ideal conditions (i.e. the estimation error of χ is zero). Intuitively, if the estimation error is limited, controller (5.5) is expected to ensure that the path is followed with an error due to (5.7). However, if no landmark is detected, hence no absolute measure is available, the localisation is affected by dead-reckoning of odometry and hence the estimation error grows potentially unbounded. Hence, the path following error grows as well. In

this following section we will show the guard adopted to shift the control authority between the robot and the user and how this authority-sharing idea can be formally modelled using tools from hybrid systems [53].

5.2.1 Controller probabilistic analysis

To explain the rationale of the probabilistic analysis of the controller, let us specify what changes in (5.7) when the state χ is not known, the control input $\omega(\cdot)$ in (5.5) is computed using the available estimate $\hat{\chi}$, which is affected by the estimation error noise ε, i.e.

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_\theta \end{bmatrix} = \begin{bmatrix} l_x - \hat{l}_x \\ l_y - \hat{l}_y \\ \tilde{\theta} - \hat{\tilde{\theta}} \end{bmatrix} = \chi - \hat{\chi}. \tag{5.8}$$

Using the Taylor expansion for the nonlinear functions in (5.5) about the estimated quantities, and recalling (5.8), one gets

$$\begin{aligned} \gamma(\chi) = c(s)\Theta(\hat{\chi},\varepsilon) + \left(\delta'(\hat{l}_y) + \delta''(\hat{l}_y)\varepsilon_y\right) \\ \left[-c(s)\Theta(\hat{\chi},\varepsilon)(\hat{l}_x + \varepsilon_x) + \sin(\hat{\tilde{\theta}}) + \cos(\hat{\tilde{\theta}})\varepsilon_\theta\right] + \mathcal{O}(\varepsilon^2), \end{aligned}$$

where $\delta'(\hat{l}_y) = \dfrac{\mathrm{d}\delta}{\mathrm{d}l_y}(l_y)\Big|_{l_y=\hat{l}_y}$ and $\delta''(\hat{l}_y) = \dfrac{\mathrm{d}^2\delta}{\mathrm{d}l_y^2}(l_y)\Big|_{l_y=\hat{l}_y}$, where, with a light abuse of notation, we denote with $\mathcal{O}(\varepsilon^2)$ high order error terms, and where

$$\Theta(\hat{\chi},\varepsilon) = \cos(\hat{\tilde{\theta}}) - \sin(\hat{\tilde{\theta}})\varepsilon_\theta + \kappa_x\hat{l}_x + \kappa_x\varepsilon_x.$$

Hence

$$\gamma(\chi) = \gamma(\hat{\chi}) + H(\kappa_x, s, \hat{\chi})\varepsilon + \mathcal{O}(\varepsilon^2), \tag{5.9}$$

where $H(\kappa_x, s, \hat{\chi})$ is a row vector equals to

$$\begin{bmatrix} c(s)\kappa_x - c(s)\delta'(\hat{l}_y)\left(\cos(\hat{\tilde{\theta}}) + 2\kappa_x\hat{l}_x\right) \\ \left[-c(s)\left(\cos(\hat{\tilde{\theta}}) + \kappa_x + \hat{l}_x\right)\hat{l}_x + \sin(\hat{\tilde{\theta}})\right]\delta''(\hat{l}_y) \\ c(s)\sin(\hat{\tilde{\theta}})\left(\hat{l}_x\delta'(\hat{l}_y) - 1\right) + \cos(\hat{\tilde{\theta}})\delta'(\hat{l}_y) \end{bmatrix}^T.$$

We can therefore compute the first order approximation of (5.7) as

$$\dot{V} = e_\theta\dot{e}_\theta = e_\theta\left(\omega(\hat{\chi}) - v\gamma(\chi)\right) = e_\theta\left(v\gamma(\hat{\chi}) - \kappa v\hat{e}_\theta - v\gamma(\chi)\right),$$

and then, noticing that $\hat{e}_\theta = \hat{\tilde{\theta}} - \delta(\hat{l}_y)$ and that

$$e_\theta = \hat{\tilde{\theta}} + \varepsilon_\theta - \left[\delta(\hat{l}_y) + \delta'(\hat{l}_y)\varepsilon_y + \mathcal{O}(\varepsilon_y^2)\right] = \hat{e}_\theta + G(\hat{l}_y)\varepsilon + \mathcal{O}(\varepsilon_y^2),$$

where $G(\hat{l}_y) = [0, -\delta'(\hat{l}_y), 1]$, and then plugging (5.9), we finally have

$$\begin{aligned}\dot{V} &= e_\theta \left(v\gamma(\hat{\chi}) - \kappa v\hat{e}_\theta - v\gamma(\chi)\right) \\ &= \left(\hat{e}_\theta + G(\hat{l}_y)\varepsilon + \mathcal{O}(\varepsilon_y^2)\right) \\ &\quad \left(v\gamma(\hat{\chi}) - \kappa v\hat{e}_\theta - v\gamma(\hat{\chi}) - vH(\kappa_x, s, \hat{\chi})\varepsilon + \mathcal{O}(\varepsilon^2)\right) \\ &= -v\kappa\hat{e}_\theta^2 - v\hat{e}_\theta \left(\kappa G(\hat{l}_y) - H(\kappa_x, s, \hat{\chi})\right)\varepsilon + \mathcal{O}(\varepsilon^2) \\ &= -v\kappa\hat{e}_\theta^2 - v\hat{e}_\theta \Xi(\kappa, \kappa_x, s, \hat{\chi})\varepsilon + \mathcal{O}(\varepsilon^2) \\ &= -v\kappa\hat{e}_\theta^2 + f(\varepsilon),\end{aligned} \tag{5.10}$$

where $f(\cdot)$ is a nonlinear function of the estimation error. It is evident that the negative definiteness cannot be established. Moreover, if the noise affecting the measures is Gaussian, ε could be unbounded, which rules out standard techniques, such as proving the boundedness of the Lyapunov function [67]. More importantly, even if a bound can be determined, it is not given for granted that the human using the *FriWalk* could not do anything better. Instead, notice that the Lyapunov function derivative $\dot{V}$ in (5.10) is a random variable since it depends on ε.

Definition 1 (Controller reliability)**.** *Given $\Gamma \leq 0$, the reliability $p_\Gamma(\hat{\chi})$ of a control action $\omega(\hat{\chi})$ is given by probability*

$$p_\Gamma(\hat{\chi}) = Pr\left[\dot{V} < v\Gamma\right], \tag{5.11}$$

where $Pr\left[\dot{V} < v\Gamma\right]$ denotes the probability that the event $\dot{V} < v\Gamma$ takes place.

The constant $\Gamma \leq 0$ is a minimum convergence speed that the controller is required to guarantee. Roughly speaking, the reliability $p_\Gamma(\hat{\chi})$ is the probability that the controller ensures at least such convergence speed. Scaling Γ by v is not strictly necessary but it comes handy since $\dot{V}$ is linear with respect to v as well. In fact, if the controller were deterministic as in (5.7), we would get $\dot{V} < v\Gamma \iff -v\kappa e_\theta^2 < v\Gamma \iff -\kappa e_\theta^2 < \Gamma$.

Using (5.10), it is now possible to compute a first order approximation of the mean value $\overline{\dot{V}}$ and of the standard deviation $\sigma_{\dot{V}}$, which are required to compute the controller reliability as per Definition 1. Assuming as customary that $\mathrm{E}\{\varepsilon\} = 0$, we can readily have

$$\begin{aligned}\overline{\dot{V}} &= \mathrm{E}\left\{\dot{V}\right\} = -v\kappa\hat{e}_\theta^2 \\ \sigma_{\dot{V}}^2 &= \mathrm{E}\left\{\left(\dot{V} - \mathrm{E}\left\{\dot{V}\right\}\right)^2\right\} \\ &= \mathrm{E}\left\{v^2\hat{e}_\theta^2 \Xi(\kappa, \kappa_x, s, \hat{\chi})\varepsilon\varepsilon^T \Xi(\kappa, \kappa_x, s, \hat{\chi})^T\right\} \\ &= v^2\hat{e}_\theta^2 \Xi(\kappa, \kappa_x, s, \hat{\chi}) P_\varepsilon \Xi(\kappa, \kappa_x, s, \hat{\chi})^T.\end{aligned} \tag{5.12}$$

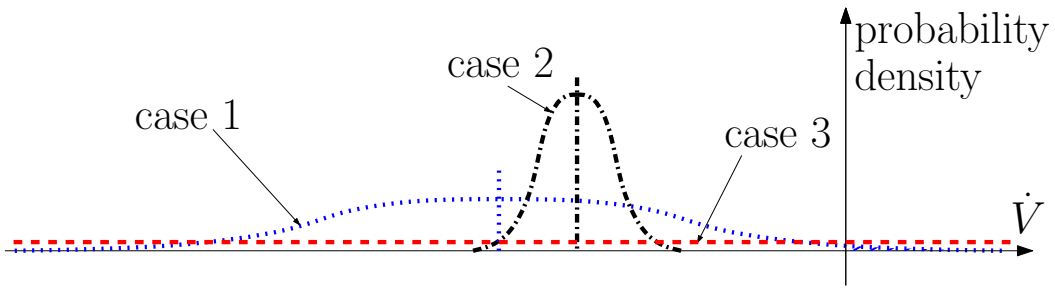


Figure 5.3: Examples of distributions of Lyapunov function derivatives.

By denoting with $\chi = \Phi(\chi^\star)$ the diffeomorphism between the two state spaces and with J_Φ its Jacobian, we immediately have that $P_\varepsilon = J_\Phi P J_\Phi^T$, where P has been defined in (5.4) as the localisation algorithm estimation uncertainty. Under the assumption of Gaussian distribution, using mean value and covariance from (5.12), the probability (5.11) can be explicitly computed.

The idea proposed in this Chapter is to allocate the control authority on the basis of the controller reliability (5.11). To intuitively describe this approach, we compare `case 1` and `case 2` in Figure 5.3. Suppose for simplicity that $\Gamma = 0$ in the definition of controller reliability (5.11). The mean value of $\dot{V}$ in `case 1` is smaller (i.e. larger convergence rate) than `case 2`, while its covariance is much larger than the covariance of `case 2`. This implies that the reliability of the controller is larger in `case 2`, since the probability to get $\dot{V} < 0$ is larger than `case 1`. Consider also `case 3`, where the covariance tends to infinity, i.e. absence of information. Since the controller reliability is in this case 0.5, any action the robot performs has 50% chance of reducing the attitude error e_θ.

Remark 1. *When $e_\theta \to 0$, the controller reliability decreases (see* (5.11)*), hence a higher probability of having the human in control of the vehicle is of course expected. By noting that $\dot{V}$ is a quadratic function of e_θ, those points correspond also to the region in which the first order linear approximation in* (5.12) *is less reliable. By combining this observation with the fact that in those points the control effort is also smaller, a certain degree of robustness of the proposed approach to linearisation errors can also be inferred.*

5.2.2 Hybrid authority-sharing

The control authority is shared with the user on the basis of the controller reliability, as shown in Figure 5.4. To properly implement a smooth transition, we define a hysteresis mechanism by formulating the control law as a hybrid system [53]. More in depth, let $q \in \{0, 1\}$ be a logic variable defining who retains the control authority. If $q = 0$ the controller reliability is small and then the user is in control of the vehicle, i.e. the vehicle actuators are not active (*user in control* state in

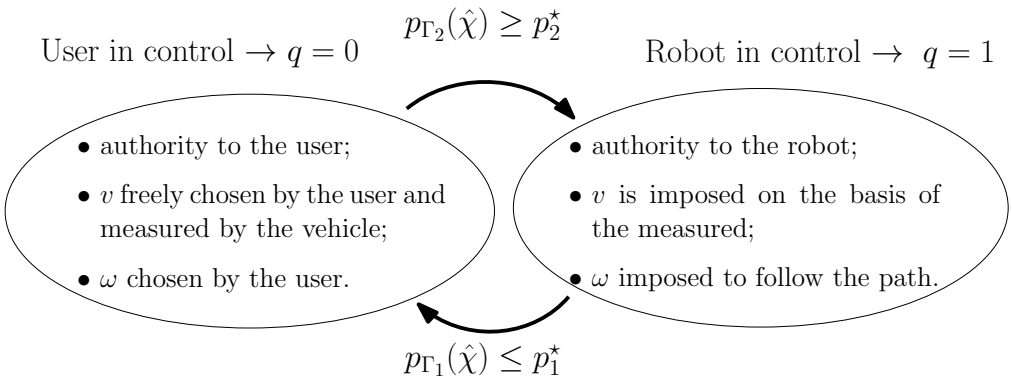


Figure 5.4: Control authority sharing of the hybrid controller (5.14).

Figure 5.4). While if $q = 1$ the controller reliability is large and hence the robot is in control (*robot in control* state in Figure 5.4 and the control action (5.5) is applied to steer the vehicle towards the path). The hysteresis is defined on the basis of two constants $\Gamma_2 > \Gamma_1 \geq 0$ representing convergence speed thresholds. Let $p_1^\star \in (0, 1)$ and $p_2^\star \in (0, 1)$, $p_1^\star \leq p_2^\star$, be the minimum tolerated reliabilities that, respectively, activate and disengage the controller. The overall controller is formalised as the following hybrid system having state $[e_\theta, q]^T$.

$$\begin{cases} \dot{q} & = 0, \quad [e_\theta, q]^T \in \mathcal{C}, \\ q^+ & = 1 - q, \quad [e_\theta, q]^T \in \mathcal{D}, \end{cases} \tag{5.13}$$

where $\mathcal{C} := \mathcal{C}_0 \cup \mathcal{C}_1$ and $\mathcal{D} := \mathcal{D}_0 \cup \mathcal{D}_1$ are the flow and the jump set respectively, where

$$\begin{aligned} \mathcal{C}_0 &= \{p_{\Gamma_2}(\hat{\chi}) \leq p_2^\star \ \wedge \ q = 0\}, \\ \mathcal{C}_1 &= \{p_{\Gamma_1}(\hat{\chi}) \geq p_1^\star \ \wedge \ q = 1\}, \\ \mathcal{D}_0 &= \{p_{\Gamma_2}(\hat{\chi}) \geq p_2^\star \ \wedge \ q = 0\}, \\ \mathcal{D}_1 &= \{p_{\Gamma_1}(\hat{\chi}) \leq p_1^\star \ \wedge \ q = 1\}. \end{aligned} \tag{5.14}$$

This way, the angular velocity of the vehicle is $\omega = qv\,(\gamma(\hat{\chi}) - \kappa\hat{e}_\theta) + (1 - q)\omega_{\text{user}}$, where ω_{user} is the angular velocity that the user imposes when he/she has the control authority.

5.3 Simulation Results

The proposed controller has been extensively tested in simulations with different choices of the controller tuning parameters. In the results here reported, those parameters are set as follows: $\kappa_x = 1$ and $\kappa = 0.5$ in (5.10), $p^\star = p_1^\star = p_2^\star = 0.9$ in (5.14), and $\Gamma_1 = -0.03$ and $\Gamma_2 = -0.24$ for the thresholds of Definition 1. Since

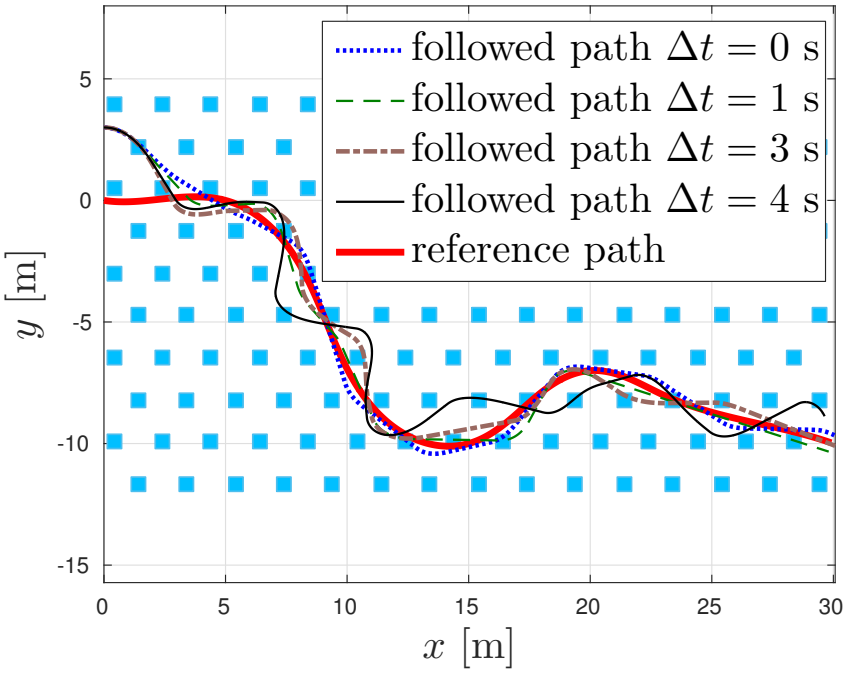


Figure 5.5: Paths followed for different reading periods Δt. The squares represent the landmark positions.

Γ_1 and Γ_2 are compared with $-\kappa\hat{e}_\theta^2$ in (5.14), the corresponding mean tolerated attitude errors are 15° and 40°, respectively. The implemented localisation algorithm computing (5.4) is an extended Kalman filter fusing the odometric data with the absolute position measure from the landmarks [93]. The landmarks are deployed following [91] to ensure that at least one marker is always in the field of view of the camera (depicted with squares in Figure 5.5). The landmark reading uncertainty is 10° for the vehicle orientation and 10 cm for the position. The uncertainty due to encoders is of 13 mm per wheel revolute.

Recall that the underlying assumption of the proposed solution is that the path following performance of the proposed solution depends on the ability of the user to follow the path when the uncertainty grows. In fact, if the user is cooperative, rely on her/him is quite rewarding, while if the user is completely uncooperative (i.e. he/she constantly moves away from the path on purpose), the path following error grows. Notably, the user behaviour cannot be known in advance (and also, it is a challenging problem to define a suitable "cooperativeness" measure). Nevertheless when the path following error grows, the controller reliability as per Definition 1 grows as well, thus limiting the deviation from the planned path. In the simulations, when the user has the control authority, he/she is modelled with a neutral behaviour, that is he/she pushes the vehicle forward (i.e. $\omega_{\text{user}} = 0$). Figure 5.5 shows the paths

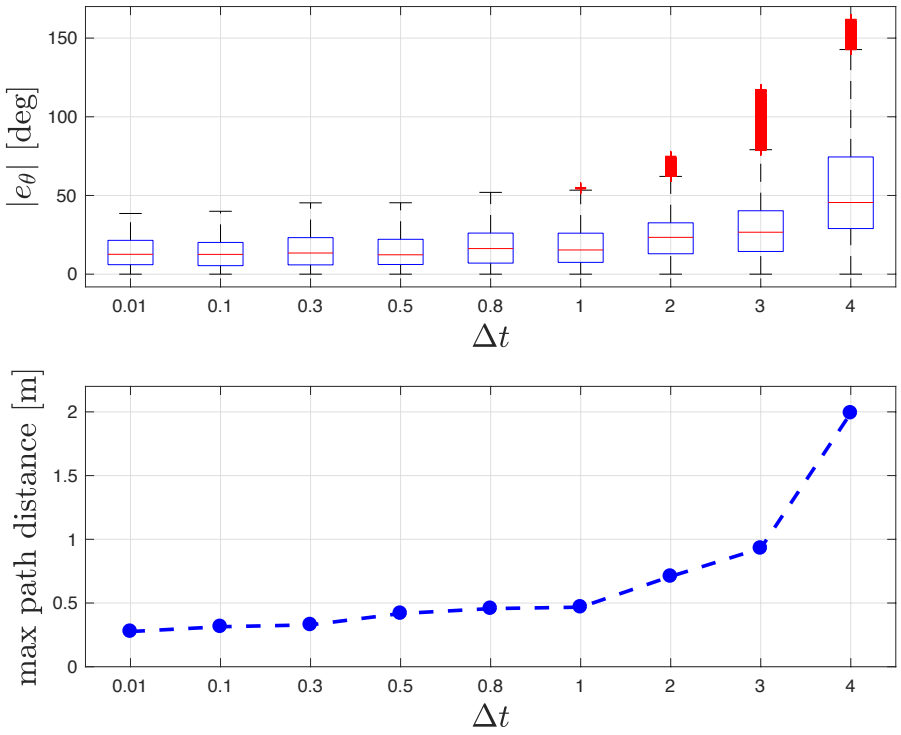


Figure 5.6: Box and whiskers plot of $|e_\theta|$ (top) and maximum path following error (bottom) for Montecarlo simulations with growing reading periods.

followed by the robot varying the landmark reading time interval Δt. Notice that if Δt is small (0, that is continuous reading, or 1 second) the vehicle is maintained close to the path, which is a trivial consequence of the small covariance (5.4) due to frequent landmarks readings: as a consequence, the robot remains in control most of the time. If Δt is larger (3 or 4 seconds), the controller reliability is, on average, smaller. In other words, the control authority is given to the robot only when the reliability exceeds the threshold $p^\star$, which happens for larger mean values of $\dot{V}$ (i.e. for larger attitude errors $|e_\theta|$, see (5.12)). Further simulations are presented in Figure 5.6, where the influence of landmark reading time Δt on the norm of the orientation error is shown in a probabilistic sense. For each Δt, 100 simulations are executed. Notice that, the larger Δt, the larger the attitude error e_θ, since the user is endowed with more control authority in the presence of large uncertainty. A similar behavior is obtained for the worst case distance to the path (see Figure 5.6, bottom plot).

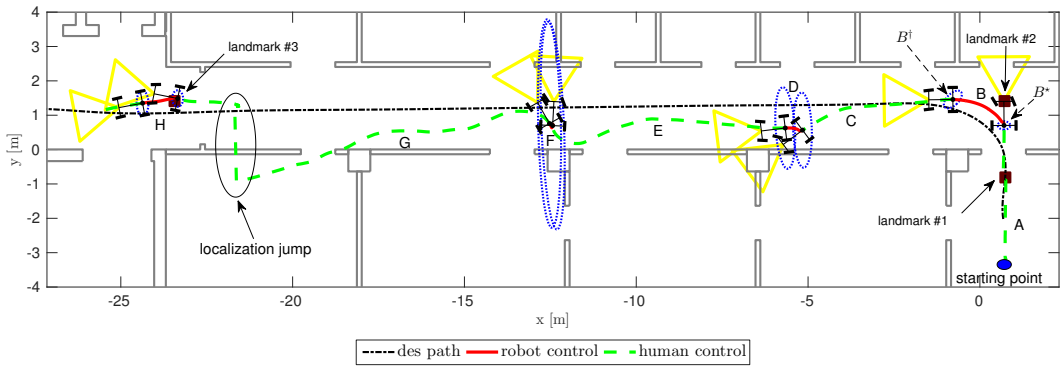


Figure 5.7: A sample trajectory of the experimental trial with localisation covariance depicted in selected points. The picture reports the desired path (dash-dotted line) and the estimated trajectory obtained by the localisation algorithm (dashed line). This trajectory is divided into sub-paths for reading easiness.

5.4 Experimental validation

The experimental results have been collected using the *FriWalk* (Figure 5.1; see also the accompanying video). The controller parameters adopted in the experiments are: $\kappa_x = 1$, $\kappa = 0.5$, $p_1^\star = 0.7$, $p_2^\star = 0.9$, $\Gamma_1 = -0.004$ and $\Gamma_2 = -0.137$. With respect to the simulation results in Section 5.3, the probability $p_1^\star$ has been reduced to give more authority to the controller, thus increasing the user's comfort. Similarly, both the mean tolerant attitude errors Γ_1 and Γ_2 have been reduced to 5° and 30°, respectively.

The experimental scenario is the Dept. of Information Engineering and Computer Science of the university of Trento, comprising corridors and rooms Figure 5.7. The starting point, of the *FriWalk* is inside one room, represented with a blue circle in Figure 5.7. Following the idea reported in Chapter 1, the landmarks are placed only in proximity of difficult decision points, i.e. landmark #1 is in the starting room in the vicinity of the exit door, landmark #2 has been collocated at the beginning of the corridor, while landmark #3 is deployed before two intersecting corridors. In the corridor, due to the particular desired path considered (dash-dotted black line of Figure 5.7), has no landmark since the only available choice is to maintain on the course. The depicted yellow solid triangle pointing forward represents the field of view of the camera attached to the vehicle and used to detect the landmarks, while the dotted blue ellipses represent the localisation error covariance P_{xy} (upper 2×2 matrix of (5.4)) in selected positions. To better analyse the experiments, the path is divided in the following parts:

Sub-path A: the user is in control of the robot ($q = 0$ in (5.13)) and pushes the *FriWalk* outside from the room since the localisation error is very high (i.e. kidnapped robot problem, dashed green line in Figure 5.7).

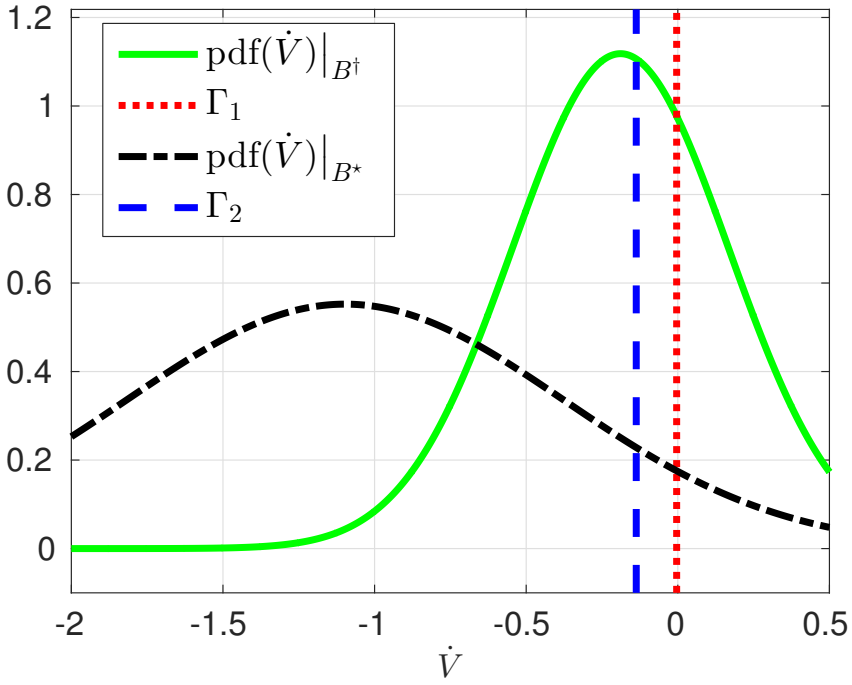


Figure 5.8: Distribution of $\dot{V}$ at the beginning of section $B \to B^\star$ (dash–dotted black line) and at the end of section $B \to B^\dagger$ (solid green line).

Sub-path B: when the vehicle detects a landmark in position $B^\star$, $p_{\Gamma_2}(\hat{\chi}) > p_2^\star$ and the controller (5.14) enters in the jump set $\mathcal{D}_1$ so that $q \to 1$. The robot is hence in control ($q = 1$ in Figure 5.4). The Gaussian probability density function (pdf) of $\dot{V}$ in point $B^\star$ is reported with dash-dotted black line in Figure 5.8. During the *robot in control* state ω is imposed by the control law and steers the walker toward the desired path (red solid line in Figure 5.7). At point $B^\dagger$, $p_{\Gamma_1}(\hat{\chi}) < p_1^\star$ and the authority is given back to the user since $q \to 0$ (the solid green Gaussian pdf in Figure 5.8).

Sub-path C: in this section the user is in control and the covariance P_{xy} grows (no landmark detected), hence the pdf flattens, so that it is more difficult for the controller to kick in. Nonetheless, at the end of sub-path **C**, the orientation error becomes so large (indeed, $\dot{V}$ is a quadratic function of $\hat{e}_\theta$ (5.10)) that $p_{\Gamma_2}(\hat{\chi}) \geq p_2^\star$ and the controller intervenes to align the user toward the path.

Sub-path D: due to the shape of the Gaussian, which is more flat than in sub-path **B**, it takes a smaller time to reach the condition $p_{\Gamma_1}(\hat{\chi}) < p_1^\star$. However, the user receives the input to realign towards the desired path.

Sub-path E: the user has the possibility to move freely since the covariance of the localisation error is very high. At the end of this sub-path the user tried to perform an U-turn, but the controller do not allow this manoeuvre as at the end of Sub-path

C.
Sub-path F: the same of sub-path **D**, but even shorter.
Sub-path G: from the beginning of this sub-path, no landmark is in view for 12.5 meters, so that the uncertainty grows unbounded. Notice that the walker wrongly localises through a wall, which is obviously not true: however, if the *robot in control* was active, the vehicle would be guided over the desired path and hence, aligning the green dashed line over the dash-dotted desired path, the *FriWalk* would be steered towards the wall on the other side of the corridor. Instead, after landmark #3 is detected and the uncertainty drops, it can be seen that the vehicle was correctly very close to the path, guided by the user.
Sub-path H: finally, the controller takes the control of the robot since $p_{\Gamma_2}(\hat{\chi}) \geq p_2^\star$.

Chapter 6

UWB

The combination of egomotion estimators and localization sensors is relatively common in robotics [121]. Typically it is achieved by fusing the localization sensor (in our case, UWB) at a low rate, and leveraging the fast egomotion estimates in between samples to keep stable the controller. However, selecting the appropriate sampling period is challenging if the robots motion is not uniform. The sampling rate must be dimensioned for the motion that generates the fastest uncertainty growth, yielding a frequency that over-provisions the number of measurements for most of the other motions; in addition, the system may still fail to meet the required guarantees if the worst-case is incorrectly estimated. In contrast, our approach *adapts* the sampling rate to the current uncertainty, *without requiring a priori estimation and regardless of its nature*. By triggering UWB samples only when needed we achieve the same accuracy at a fraction of the cost (<50% in many cases, Section 6.4–6.6). Moreover, as the sampling rate is not fixed (and bound) a priori, we can cope with unexpected changes due to environmental factors (e.g., the presence of uneven or slippery terrain) or malfunctioning of the odometer.

In this Chapter the problem of accurate and high-rate self-localization based on ultra-wideband infrastructure for a mobile robot is considered. We *adaptively* combine the speed information acquired by proprioceptive sensors with *intermittent* positioning samples acquired via ultra-wideband (UWB) radios. These are triggered only if and when needed to reduce the positioning uncertainty. Our formulation is agnostic w.r.t. the source of uncertainty and enables an intuitive specification of user navigation requirements along with stochastic guarantees on the system operation. Experimental results in simulation and with a real platform show that our approach *i)* meets these guarantees in practice *ii)* achieves the same accuracy of a fixed periodic sampling but with significantly higher scalability and lower energy consumption *iii)* is resilient to errors in UWB estimates, enabling the use of low-accuracy ranging schemes which further improve these two performance metrics.

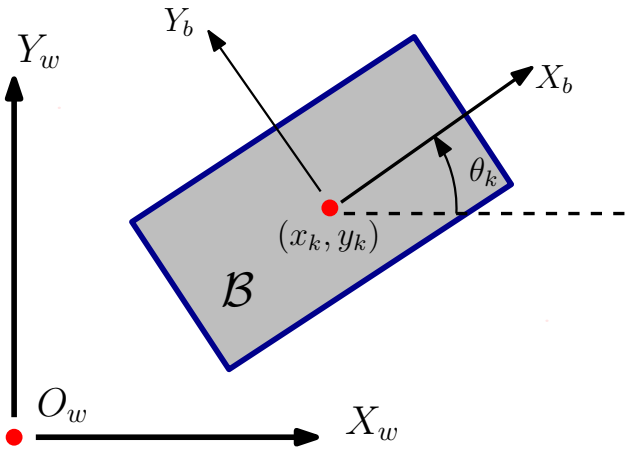


Figure 6.1: Platform model represented as a rigid body $\mathcal{B}$ moving on the $X_w \times Y_w$ plane with an attached reference frame $\langle B \rangle$.

6.1 Models Overview

The fixed, right-handed reference frame for platform localization is referred to as $\langle W \rangle = \{O_w, X_w, Y_w, Z_w\}$, as shown in Fig. 6.1. The robotic vehicle is regarded as a rigid body $\mathcal{B}$ moving on the plane $X_w \times Y_w$. Let t_s denote the sampling period, assumed common for all onboard sensors. The generalized coordinates of the robot at time kt_s are denoted by $\boldsymbol{p}_k = [x_k, y_k, \theta_k]^T$, with (x_k, y_k) being the coordinates of the origin of frame $\langle B \rangle = \{O_b, X_b, Y_b, Z_b\}$ attached to the robot, and with θ_k representing the angle between X_b and X_w. The kinematic model of a generic drift–less, input–affine robot can be written as:

$$\begin{cases} \boldsymbol{p}_{k+1} = \boldsymbol{p}_k + G_k(\boldsymbol{p}_k, \boldsymbol{p}_k + \boldsymbol{\epsilon}_k) \\ \boldsymbol{z}_k = h(\boldsymbol{p}_k) + \boldsymbol{\eta}_k \end{cases} \tag{6.1}$$

where $\boldsymbol{p}_k$ is the robot state, $\boldsymbol{u}_k$ is the input vector, assumed piecewise constant between $(k-1)t_s$ and kt_s, $\boldsymbol{\epsilon}_k$ is the additive zero-mean uncertainty term affecting the input quantities and $G_k(\cdot)$ is the input vector function. At any sampling period, the robot samples its internal sensors (e.g., encoders, IMU), but the estimate of the state is affected by the dead-reckoning effect modelled in the term $\boldsymbol{\epsilon}_k$. In order to prevent the drift of the estimate, the robot is also allowed to collect a measurement of a quantity $\boldsymbol{z}_k$ from an exteroceptive sensor (in our case the UWB localisation system) every t_s time units. The quantity $\boldsymbol{z}_k$ is related to the state through the non linear function $h(\cdot)$. We assume that: *i)* the measurement $\boldsymbol{z}_k$ is affected by a noise term $\boldsymbol{\eta}_k$, *ii)* $h(\boldsymbol{p}_k)$ ensures nonlinear observability of the robot state $\boldsymbol{p}_k$ in $\langle W \rangle$, and *iii)* each measurement of $\boldsymbol{z}_k$ has a cost $c(\boldsymbol{z}_k)$. For UWB localisation, the cost term accounts for the communication bandwidth used for every measurement

(which reduces the number of robots that can use the system at the same time), and for the energy consumed in transmitting and receiving packets (which reduces the robot's autonomy).

6.2 Problem Formulation

Our problem can be formalized in the following terms. Given a robot with kinematic model (4.1), produce an estimation $\hat{\boldsymbol{p}}_k$ for the state such that: *i)* the cost $c(\boldsymbol{z}_k)$ is minimized, and *ii)* the uncertainty is upper-bounded by a threshold Λ_k with a confidence ψ_k. The accuracy is modelled by a nonnegative performance function $H_k(\cdot)$ of the localisation error $e_k = \hat{\boldsymbol{p}}_k - \boldsymbol{p}_k$. The performance function $H_k(\cdot)$ is generally multi-dimensional: $H_k : \mathbb{R}^2 \times \mathbb{S} \to \mathbb{R}^n$, where $\mathbb{R}^2$ is the dimension of the state space and n is the number of constraints required. The confidence is defined as a probability, i.e., $\psi_k \in [0, 1]$. Hence, the accuracy requirement is analytically expressed as:

$$\Pr\left[H_k(\boldsymbol{e}_k) < \Lambda_k\right] \geq \psi_k, \forall k. \tag{6.2}$$

The performance function $H_k(\boldsymbol{e}_k)$ is time varying because depends on changing conditions of the application scenario. For instance, if the robot is moving along a corridor oriented along X_w, it is reasonable to upper-bound the error along Y_w (to avoid collisions with the walls) and on the orientation θ_k (e.g., to secure that the robot keeps moving in the chosen direction). A possible performance function could be:

$$H_k(\boldsymbol{e}_k) = \begin{bmatrix} |\hat{y}_k - y_k| \\ |\hat{\theta}_k - \theta_k| \end{bmatrix} \leq \begin{bmatrix} \lambda_{y_k} \\ \lambda_{\theta_k} \end{bmatrix} = \Lambda_k, \tag{6.3}$$

with confidence ψ_k. In other cases, we may want to limit the position error on the plane in order to safely enter into a room through the door; we could model this by:

$$H_k(\boldsymbol{e}_k) = \sqrt{(\hat{x}_k - x_k)^2 + (\hat{y}_k - y_k)^2} \leq \lambda_k = \Lambda_k. \tag{6.4}$$

The possible choices of Λ_k and ψ_k are evidently limited by the intensity of the noise η_k (see (6.1)).

6.3 Approach

Let $f_k(\boldsymbol{e})$ denote the probability density function (pdf) of the estimation error $\boldsymbol{e}_k = \hat{\boldsymbol{p}}_k - \boldsymbol{p}_k$ at timestamp kt_s. We can rewrite (6.2) as the Riemann integral:

$$\int_{\mathcal{E}_k} f_k(\boldsymbol{e}) d\boldsymbol{e} > \psi_k, \forall k, \tag{6.5}$$

over the state subspace:

$$\mathcal{E}_k = \left\{\boldsymbol{e} \in \mathbb{R}^2 \times \mathbb{S} : H_k(\boldsymbol{e}) < \Lambda_k\right\}. \tag{6.6}$$

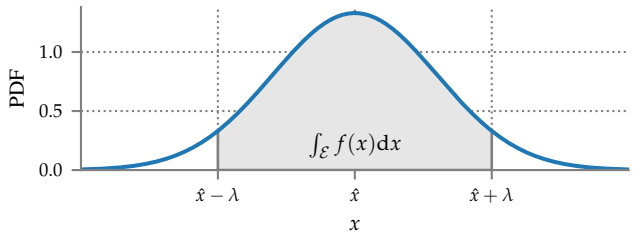


Figure 6.2: Graphical representation of the integral in (6.5) using a unidimensional pdf where $H(e) = |\hat{x} - x|$.

For illustration purposes, consider the unidimensional pdf $f(\boldsymbol{e})$ in Fig. 6.2 for generic time instant k. The function $H(e)$ is defined as $H(e) = |\hat{x} - x|$ and the resulting integration subspace is $\mathcal{E} = \{e \in \mathbb{R} : |\hat{x} - x| < \lambda\}$. The bound is respected if the pdf integral over $\mathcal{E}$ (grey area in Fig. 6.2) is $< \psi$.

Importantly, Eq. (6.5) can be used as a trigger event for the measurement of $z_k = h(\boldsymbol{p}_k)$. As long as $\int_{\mathcal{E}_k} f_k(\boldsymbol{e})d\boldsymbol{e} > \psi_k$, the degradation is not sufficient to violate the accuracy constraint, so we do not need any external measurement. However, in the absence of external measurement the dead-reckoning effect "inflates" the probability distribution of the error, with $f_k(\boldsymbol{e}) \to 0$ and, since $\mathcal{E}_k$ is a closed set, $\int_{\mathcal{E}_k} f_k(\boldsymbol{e})d\boldsymbol{e} \to 0$. Thereby, after a sufficient number of steps the threshold ψ_k will be approached. When this event occurs the $\boldsymbol{z}_k$ measurement is triggered to reset the error to acceptable levels. This policy evidently ensures that the accuracy constraint will not be violated with a very reduced numbers of expensive queries to the UWB localization system.

6.4 Evaluation setup

We now give a more specific description of the general approach and model outlined in Section 6.3 and Section 6.1 respectively tailored on the setup used simulation and experimental evaluation.

For the robot kinematics, we assume a unicycle model. The robot planar coordinates (x_k, y_k) correspond to the mid-point of the traction wheels, assumed to be the origin O_b of the body frame $\langle B \rangle$. The Y_b axis points through the left wheel, while the X_b axis is oriented in the forward direction, as shown in Fig. 6.3 for the robot used for the experiments. Assuming that relative encoders are attached to each wheel shaft, measuring respectively the left and right wheel angular displacements δ_k^l and δ_k^r in the time interval $[kt_s, (k+1)t_s]$, we can compute the linear motions of the left or right wheel as $L_k = \frac{\phi_l}{2}\delta_k^l$ and $R_k = \frac{\phi_r}{2}\delta_k^r$, being ϕ_l and ϕ_r the wheels radii. With reference to the generic model (6.1) and assuming for simplicity that in t_s the speed of the wheels is constant, we can safely assume that the inputs are $u_k = [L_k, R_k]^T$. As a consequence, we have that the nonlinear input vector is given

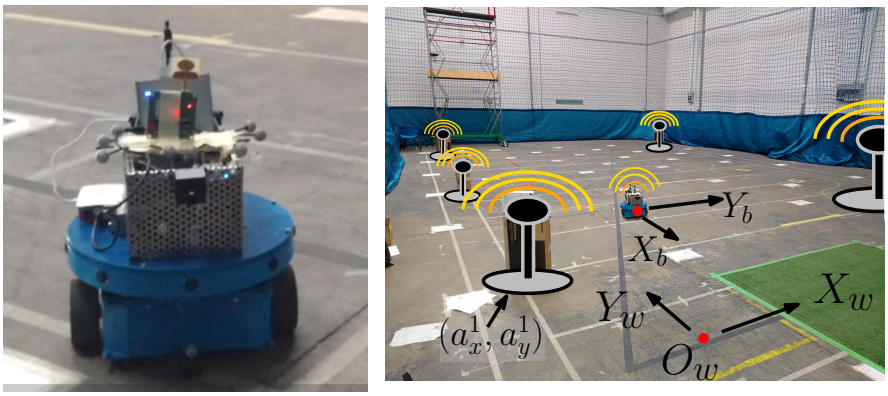


Figure 6.3: Unicycle-like vehicle adopted in the experiment and representation of the reference system. UWB anchors are depicted as well.

by

$$G_k = \begin{bmatrix} r_k \left(\sin\left(\theta_{k-1}\right) - \sin\left(\theta_{k-1} + \gamma_k\right)\right) \\ r_k \left(\cos\left(\theta_{k-1} + \gamma_k\right) - \cos\left(\theta_{k-1}\right)\right) \\ \gamma_k \end{bmatrix},$$

where $r_k = \frac{R_k + L_k}{2} + \frac{\epsilon_{R_k} + \epsilon_{L_k}}{2}$ and $\gamma_k = \frac{R_k - L_k}{b} + \frac{\epsilon_{R_k} - \epsilon_{L_k}}{b}$. As reported in [29], the additive white Gaussian noise covariance matrix E_k of $\boldsymbol{\epsilon}_k = [\epsilon_{L_k}, \epsilon_{R_k}]^T$ can be computed as:

$$E_k = \begin{bmatrix} k_L^2 \left|L_k\right| & 0 \\ 0 & k_R^2 \left|R_k\right| \end{bmatrix}, \tag{6.7}$$

with ϵ_{L_k} and ϵ_{R_k} being uncorrelated.

6.4.1 Measurement System

As a global localization system we use UWB sensors. Such a system is capable to carry out ranging measurements from a set of fixed anchors with known positions. Let us assume to have m anchors, each having known plane position (a_x^i, a_y^i), with $i = 1, \ldots, m$ (see Fig. 6.3 for the actual deployment of the UWB anchors). Therefore, the measure $\boldsymbol{z}_k$ is composed by the distance measure coming from m different anchors, i.e., $\boldsymbol{z}_k = [z_k^1, z_k^2, ..., z_k^m]$, where

$$z_k^i = \sqrt{\left(x_k - a_x^i\right)^2 + \left(y_k - a_y^i\right)^2}.$$

The zero mean, white Gaussian noise $\boldsymbol{\eta}_k$ affecting each measurement is described through the covariance matrix:

$$N_k = \operatorname{diag}\left\{\nu_k^1, \nu_k^2, \ldots, \nu_k^n\right\}, \tag{6.8}$$

where $\nu_k^i \in \mathbb{R}_{\geq 0}$ is the variance of the i-th ranging measure and the noise term of the anchors is assumed uncorrelated.

6.4.2 Stochastic Guarantees

One widely adopted solution for collecting both a localization estimate $\hat{\boldsymbol{p}}_k$ and the associated pdf of the error $f_k(\boldsymbol{e})$ is to resort to Bayesian filters, which are able to coherently fuse the prior on the odometry with the triggered measurement from the UWB sensing system. Due to the nonlinear nature of the model and of the measures, we select the Unscented Kalman Filter (UKF) [123]. This way, we can capture the posterior mean $\hat{\boldsymbol{e}}_k$ and covariance Σ_k up to the 3^{rd} order of the Taylor series expansion for the nonlinearities, modeled as a Gaussian. Without loss of generality, we adopt as performance function $H_k(\boldsymbol{e})$ the position error defined in (6.4), hence deriving for (6.6) the following

$$\mathcal{E}_k = \left\{ \boldsymbol{e} \in \mathbb{R}^2 \times \mathbb{S} : \sqrt{(\hat{x}_k - x_k)^2 + (\hat{y}_k - y_k)^2} < \lambda \right\}, \tag{6.9}$$

where λ is assumed here to be constant. $\mathcal{E}_k$ is a cylinder contained in the state space $\mathbb{R}^2 \times \mathbb{S}$, and it can be equivalently represented with $\mathcal{A}_k \times \mathbb{S}$ being $\mathcal{A}_k$ the base of the cylinder in the $X_w \times Y_w$ plane. Equation (6.5) can be written as:

$$\int_{\mathcal{E}_k} \frac{1}{\sqrt{2\pi d_{p_k}}} e^{-\frac{(\boldsymbol{p}-\hat{\boldsymbol{p}}_k)^T \Sigma_k^{-1} (\boldsymbol{p}-\hat{\boldsymbol{p}}_k)}{2}} d\boldsymbol{p}, \tag{6.10}$$

where $d_{p_k} = \det(\Sigma_k)$. Marginalizing along θ and defining as Σ_k^{xy} the top 2×2 matrix of Σ_k, Equation (6.10) can be simplified as:

$$\frac{1}{\sqrt{2\pi d_{xy_k}}} \int_{\mathcal{A}_k} e^{-\frac{(\boldsymbol{p}^{xy}-\hat{\boldsymbol{p}}_k^{xy})^T \Sigma_k^{xy-1} (\boldsymbol{p}^{xy}-\boldsymbol{p}_k^{xy})}{2}} d\boldsymbol{p}^{xy}, \tag{6.11}$$

where $\boldsymbol{p}^{xy}$ and $\hat{\boldsymbol{p}}_k^{xy}$ are the entries related to the cartesian positions of $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}_k$, respectively, while $d_{xy_k} = \det(\Sigma_k^{xy})$. Expressing the previous integral in cylindric coordinates where $\boldsymbol{v}(\psi) = [\cos(\psi), \sin(\psi)]^T$ is the unit vector indicating the integration direction, Equation (6.11) can be rewritten as

$$\frac{1}{\sqrt{2\pi d_{xy_k}}} \int_0^{2\pi} \int_0^{\lambda} \rho e^{-\frac{\rho \boldsymbol{v}(\psi)^T \Sigma_k^{xy-1} \rho \boldsymbol{v}(\psi)}{2}} d\rho \, d\psi. \tag{6.12}$$

The integral in ρ can be solved analytically leading to:

$$\frac{1}{\sqrt{2\pi d_{xy_k}}} \int_0^{2\pi} \frac{e^{\lambda^2 C(\psi)} - 1}{2C(\psi)} d\psi \tag{6.13}$$

where $C(\psi) = -\boldsymbol{v}(\psi)^T \Sigma_k^{xy-1} \boldsymbol{v}(\psi)/2$. Very efficient numeric solutions can be found for this integral within the desired level of accuracy.

Table 6.1: Anchor positions expressed in $\langle W \rangle$.

	Anchor i				
	1	2	3	4	5
$\mathbf{a_x^i}[\mathbf{m}]$	1.76	-7.2	-7.2	1.9	-1.9
$\mathbf{a_y^i}[\mathbf{m}]$	-5.3	-5.3	1.9	1.9	5.3

6.5 Simulation-based Evaluation

To validate our proposed approach, we developed a simulator assuming that the model noise of the odometer $\boldsymbol{\epsilon}_k$ and the UWB measurement noises $\boldsymbol{\eta}_k$ are generated by a white, zero-mean Gaussian stochastic processes and imposing parameters mirroring the actual experimental platform described in Section 6.6. The covariance matrix of $\boldsymbol{\epsilon}_k$ in (6.7) is computed with $k_L = k_R = 0.004$ m$^{1/2}$, while for the covariance matrix of the UWB noise η_k in (6.8) we have $\nu_k^i = 10^{-2}$ m^2, $\forall i = 1, \ldots, m$, with $m = 5$. The anchors are deployed in the environment as reported in Table 6.1.

We first depict the baseline of our approach imposing a straight line trajectory originating from position $(-4.5, -1.8)$ m and moving along increasing coordinates of the X_w axis (Fig. 6.4) with $\lambda = 0.2$ m for (6.4) and $\psi = 0.75$ for (6.5) (both constant in time). In Fig. 6.4 we may see that along the actual motion (dashed line) the pdf of the error marginalized w.r.t. $\mathbb{S}$ increases due to dead-reckoning, thus the estimated trajectory (solid line) deviates from the actual path. In the same figure, the base of the cylinders $\mathcal{A}_k$ are depicted as well (blue circles). When the robot reaches the fourth depicted position, the integral in (6.13) goes below the confidence ψ, thus a reading from the UWB system is triggered and the pdf narrows down (last depicted pdf in Fig. 6.4).

Fig. 6.5 shows the cumulative distribution function (CDF) of the localization error as a function of the user-defined confidence ψ and threshold λ (Fig. 6.5 (a)) as well as the impact of the dead-reckoning ϵ_k and UWB ranging uncertainty η_k (Fig. 6.5 (b)). Note that the CDFs reported are computed collecting the error values at their maximum right before triggering an UWB measurement, i.e., worst case scenario. Our technique easily adapts to changing performance metrics (ψ, λ) or increasing the odometry or ranging uncertainty (ϵ_k, η_k), tightly satisfying the user-defined bound. Increasing the uncertainty, however, has a remarkable effect in the frequency in which the UWB system is triggered as reported in Table 6.2. The UWB sampling frequency is dominated by the odometry uncertainty, while the UWB ranging uncertainty plays a significantly minor role. This allows us to use more energy- and time-efficient (although less accurate) ranging schemes like SS-TWR instead of DS-TWR as in [49], saving energy and increasing scalability w.r.t. the state of the art.

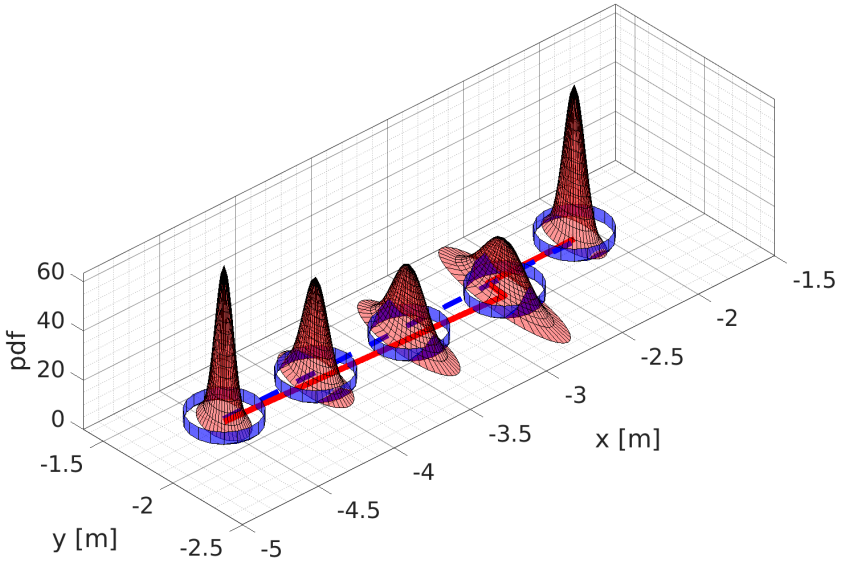
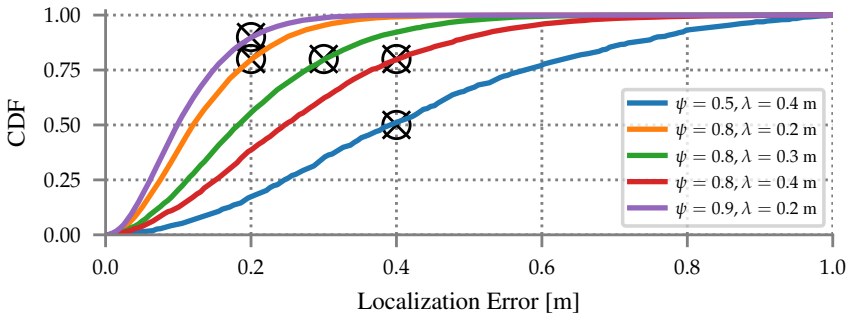
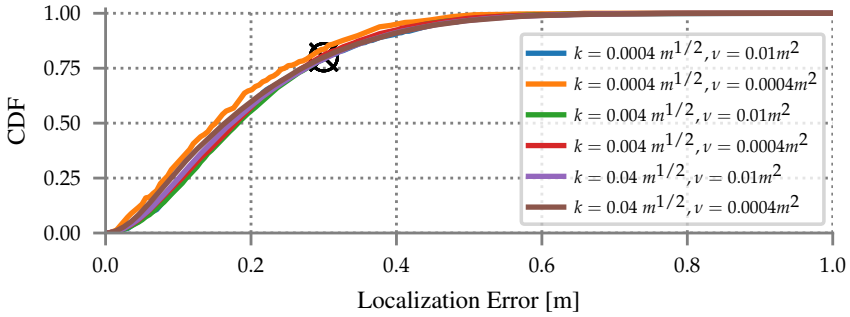


Figure 6.4: Time evolution of the marginal pdf of $f_k(e)$ for a robot moving along a straight line parallel to X_w. Both, the actual (dashed line) and estimated (solid line) trajectories are reported. At the fourth depicted position, the integral in (6.13) exceeds the given confidence ψ due to dead-reckoning, hence the UWB system is triggered and the pdf narrows down (fifth depicted pdf).

Besides the discussed intrinsic robustness of the proposed approach, we will now show how the algorithm effectively reduces the communication bandwidth utilization with respect to a periodic sampling using a trajectory with time-varying velocity. The rationale is to further highlight the adaptability of the solution at hand: when the trajectories are regular and the velocity is almost constant (see the odometer model (6.7) and its effect in Table 6.2), a low-frequency periodic sampling is sufficient; when instead the trajectories become more challenging, a high-frequency is needed. Therefore, a periodic sampling approach should be fine tuned on the worst case scenario, which leads to overcommitment of the shared UWB system. The adaptability of the proposed solution, instead, requests the use of the UWB system only when strictly needed. The example we propose stems from an industrial warehouse application and assumes the robot moving on a planned path imposing a bound of 0.5 m/s^2 in the forward and lateral accelerations, a maximum speed of 1.5 m/s and starting and ending with 0 velocity. The resulting velocity profile, visible in Fig. 6.6, is composed by a region in which the robot is moving straight at full speed in the middle of two turning sections. The obtained UWB measurements pseudo-frequencies, i.e. inverse of the local sampling periods, distribution is de-



(a)



(b)

Figure 6.5: Localization accuracy under different (ψ, λ, k, ν). Our solution adapts the CDFs (top) to the user-defined threshold λ and confidence ψ. Changing the encoder or ranging uncertainty (bottom) barely affects the resulting localization error.

picted in Fig. 6.7. The frequencies distribution are in the interval $[0.6, 1.5]$ Hz with a mean value of 1.08 Hz. Lower frequencies are relative to the slow-paced region of the velocity profile, whereas high frequencies are due to the fast-paced region (grey area in Fig. 6.6). Fig. 6.8 shows the worst-case localization error CDFs for this scenario obtained with our adaptive mechanism w.r.t. periodic UWB sampling at fixed rates. The periodic rates are respectively selected based on the minimum, mean, and maximum frequency reported in Fig. 6.7. When the robot moves at 1 m/s (left), all the CDFs comply with the user-defined bound ($\psi = 0.8, \lambda = 0.2$ m). However, increasing the velocity to 1.5 m/s (right), makes the periodic approaches with $f_s < 1.5$ Hz unable to cope with the fast-speed of the robot. Our technique, instead, adapts its sampling rate as needed and satisfies the requirements despite the actual working conditions.

Finally, we compare our proposed technique against the state of the art technique in [49], which is based on the eigenstructure of the localization error covariance. In particular, we take the greater axis of the ellipse which includes the ψ probability to scale the greater eigenvalue of the (x, y) position uncertainty matrix obtained

Table 6.2: Frequency of UWB triggering as a function of the uncertainty ϵ_k and η_k.

	$\mathbf{k_L = k_R [m^{1/2}]}$		
$\nu[\mathbf{m^2}]$	**0.0004**	**0.004**	**0.04**
0.0004	0.02 Hz	0.07 Hz	0.52 Hz
0.01	0.02 Hz	0.08 Hz	0.6 Hz

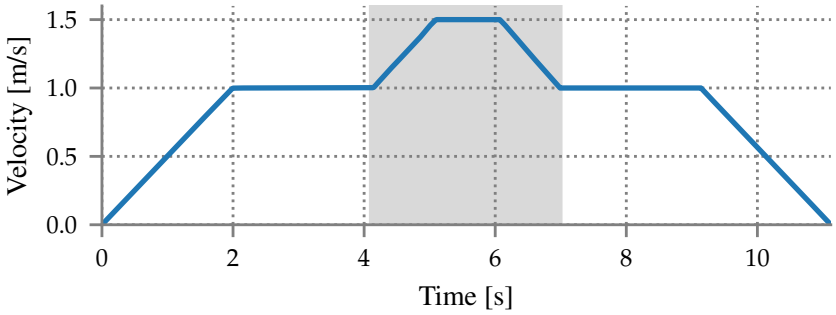


Figure 6.6: Velocity profile.

with Principal Component Analysis. The conservativeness of the methods relying on the eigenvalues is reported in Fig. 6.9, where the CDFs of the two methods are compared. It is evident that the proposed method is tighter for both tested configurations. In the first case with ($\lambda = 0.3$, $\psi = 0.8$), the proposed technique requires a mean UWB triggering frequency of 0.16 Hz instead of 0.25 Hz; in the second with ($\lambda = 0.3$, $\psi = 0.8$) the proposed adaptive system frequency was 0.38 Hz, while the eigenvalue technique yielded 0.50 Hz. Thus, the proposed technique can potentially reduce the required UWB sampling rate by a $24 - 36\%$ factor w.r.t. state of the art techniques based on the eigenvalue.

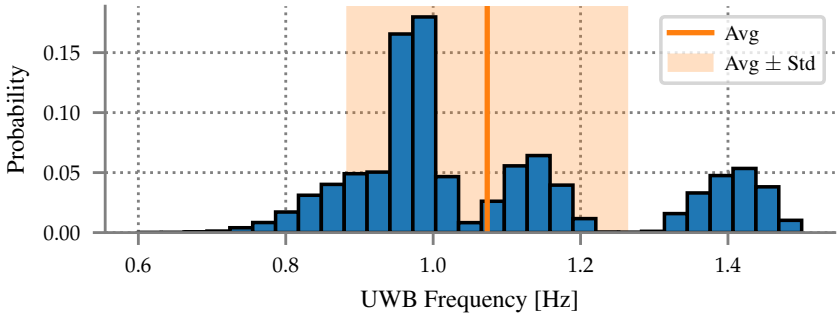


Figure 6.7: UWB sampling pseudo-frequencies distribution.

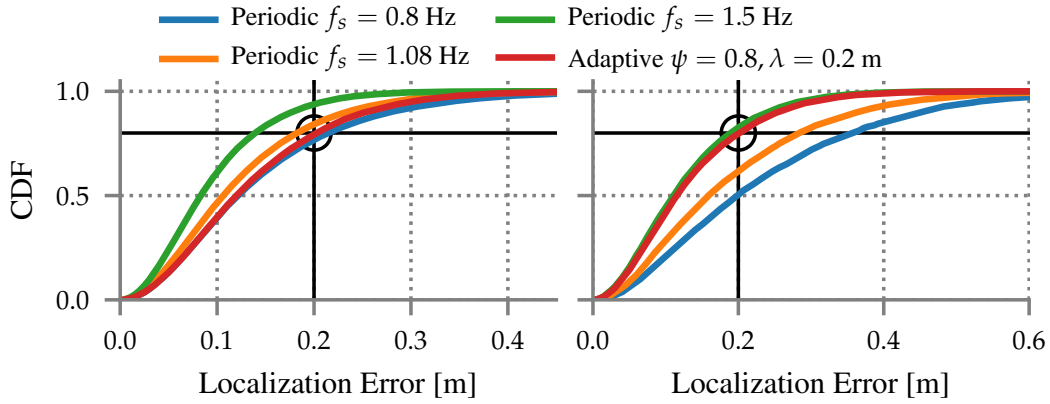


Figure 6.8: Comparison between periodic and adaptive UWB sampling with the robot moving at 1 m/s (left) and 1.5 m/s (right). Our adaptive mechanism satisfies the user requirements despite changes in the robot velocity.

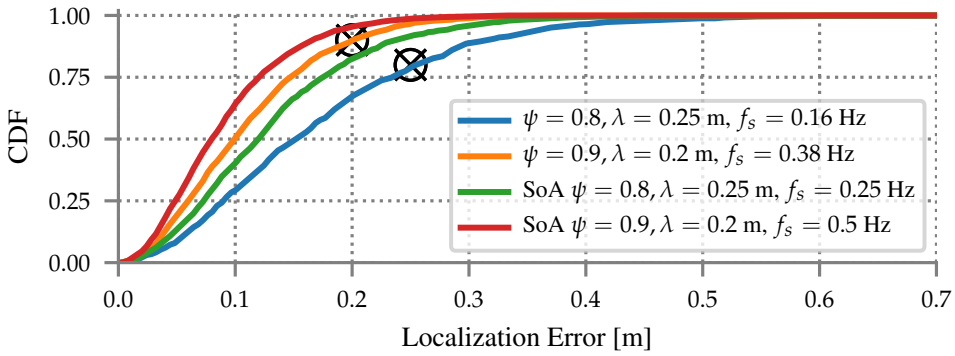


Figure 6.9: Comparison with the state of the art (SoA) for different (ψ, λ). SoA techniques based on the eigenvalue are conservative, over provisioning UWB measurements, increasing consumption, and decreasing scalability.

6.6 Experimental Results

To validate our proposed technique, we ran an experimental campaign in an environment equipped with 14 OptiTrack cameras [1] which are able to cover a 12×8 m^2 area providing mm-level localization at 125 Hz. The unicycle-like wheeled robot used for the experiments and the testing arena endowed with 5 UWB anchors is reported in Fig. 6.3. The robot had a UWB tag in its center position and measures its egomotion by wheel encoders sampled at a frequency of 50 Hz. The UWB tag performs SS-TWR in a round robin fashion with all $m = 5$ anchors at a ranging frequency of 200 Hz, providing a maximum positioning rate of 40 Hz. SS-TWR involves a two-way message exchange between the tag and an anchor. The firmware is implemented atop Contiki OS [33] for the DecaWave EVB1000 platform [36], equipped with an STM32F105 MCU, the DW1000 transceiver, and a PCB antenna. UWB and odometry data are sent via WiFi to a laptop, which also stores the ground truth data acquired from the OptiTrack system. We recorded positioning information from 6 different generic trajectories, recording about 13000 overall location samples. As a baseline, we first present the positioning results along the predefined trajectories when the UWB system is activated periodically and in isolation (i.e., without fusing UWB ranging information with the wheel encoders data). Fig. 6.10 shows the characterization of the ranging precision of the UWB (left) and the positioning results (right) when the measurements are collected at the maximum positioning frequency of 40 Hz. The standard deviation of the UWB ranging error is $\sigma = 11$ cm with a maximum error of 48 cm (Fig. 6.10 left). The ranging measurements are used to retrieve the position using a non-linear least squares solver, thus resulting in the positioning error, computed with (6.4), depicted in the right side of Fig. 6.10. The mean positioning error is $\mu_p = 9$ cm with a standard deviation of $\sigma_p = 5$ cm and a 90^{th} percentile error of 14 cm.

Fig. 6.11 shows the localization error when fusing odometry and UWB ranging data with the UKF, considering different periodic UWB sampling frequencies f_s. With $f_s = 40$ Hz the accuracy obtained is the same as the positioning error using only UWB (Fig. 6.10 right). Decreasing $f_s = 1$ Hz, the UKF yields a mean localization error $\mu = 12$ cm with a standard deviation of $\sigma = 8$ cm and a 90^{th} percentile error of 18 cm, slightly increasing the error w.r.t. only UWB, but reducing f_s by 97%. This, in turn, reduces the energy consumption, while increasing the scalability of the solution. Unfortunately, the *right* sampling frequency that satisfies a user-defined bound (ψ, λ) and fulfills the worst-case scenario for robot navigation can only be speculated (recall the analysis of the previous section, summarized in Fig. 6.8).

Sample trajectories of the experiments are reported in Figure 6.12, while the resulting PDFs are reported in Figure 6.13, proving the actual applicability of the approach for in an actual situation. Finally, Figure 6.14 reports the mean frequencies and their standard deviations as a function of the imposed performance indices.

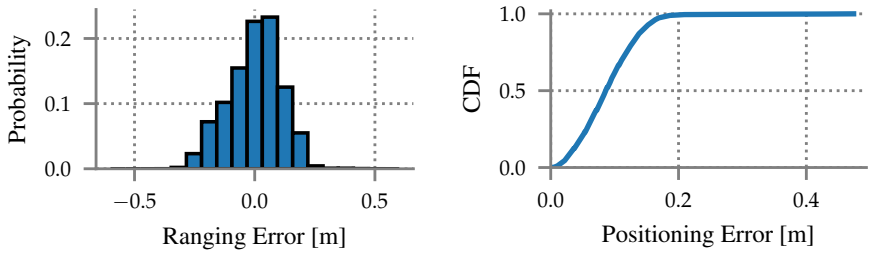


Figure 6.10: UWB ranging (left) and positioning (right) error.

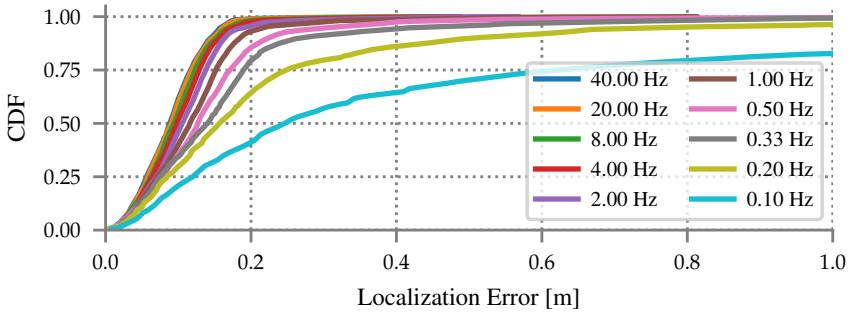
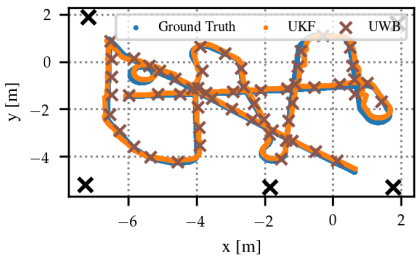
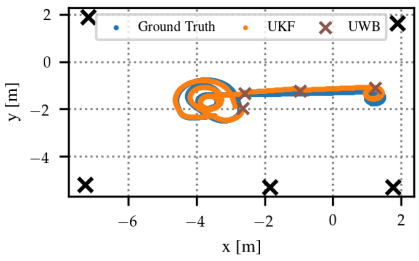


Figure 6.11: Localization error with periodic UWB sampling at different frequencies.

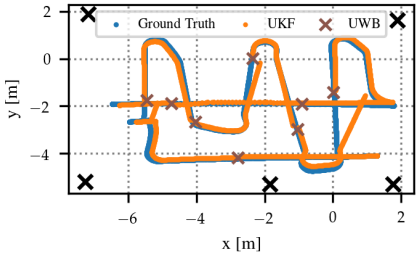
Two final remarks are in order. First, the curve at periodic 1 Hz sampling in Figure 6.11 has a maximum error $\lambda = 20$ cm with a probability of $\psi = 90\%$. With the same pair (λ, ψ), we obtain a frequency that is reduced to almost one half (see the second orange bar on the left side of Figure 6.14), further proving our claims in an experimental scenario. Second, we remark that the comparison results with [49] reported in Figure 6.9 can also be obtained in the experimental set-up, here not reported for space limits. Therefore, the experiments validates: *i)* the possibility to reduce the usage of the UWB system ensuring the same performances of periodic sampling thus improving the scalability potentialities; *ii)* the clear connection between the desired behavior chosen from calibration curves as in Fig. 6.11 and the actual algorithm.



(a) ($\psi = 0.9, \lambda = 20$ cm)



(b) ($\psi = 0.9, \lambda = 50$ cm)



(c) ($\psi = 0.9, \lambda = 100$ cm)

Figure 6.12: Localization tracking across three trajectories with $\psi = 0.9$ and different threshold λ. Each black cross represents a UWB anchor. The UKF output (orange) follows accurately the ground truth measurements (blue). As we increase λ, the number of UWB measurements (brown) needed decreases.

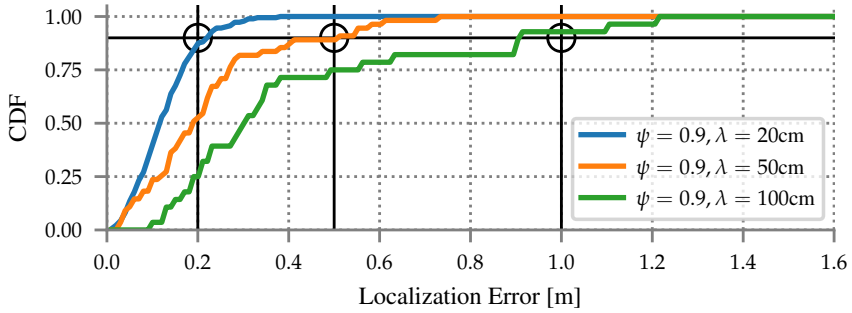


Figure 6.13: Localization error with dynamic UWB sampling and $\psi = 0.9$ for different thresholds λ. The only measurements considered are taken just before triggering UWB ranging (i.e., worst-case scenario).

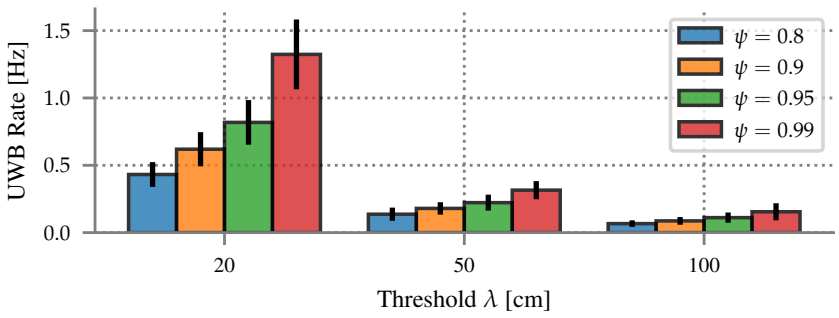


Figure 6.14: UWB localization rate with $m = 5$ ranging exchanges for different confidence intervals ψ and thresholds λ over all trajectories.

Chapter 7

RFID

In this Chapter, a theoretical analysis of the localization problem using Ultra High Frequency (UHF) RFID tags for mobile robots is performed. The feasibility of the proposed approach is demonstrated by analyzing the local nonlinear observability of the system at hand, despite the inherent ambiguity of the phase of backscattered RF signals, which can be measured by a system installed on the moving agent. The validity of the analysis and the practicality of this localization approach is further evaluated by using a position tracking estimator based on an Unscented Kalman Filter (UKF).

7.1 Problem Formulation and Models

The purpose of the method described in this Chapter is to track the position of a robot moving in an indoor environment where n RFID tags are placed at known locations. The robot is assumed to be equipped with encoders on the rear wheels providing dead reckoning. Each sensor is supposed to be affected by uncertainties generated by white stochastic processes with known features. Even though the presented approach can be applied to a generic platform moving indoor on an horizontal plane using both ego-motion data and phase measurement of RF signals backscattered from RFID tags, the reported analysis and the related simulation parameters refer to the case of a robot with an unicycle-like kinematic.

7.1.1 Platform Model

As already stated previously the analysis is done for a robot that can be modeled by a unicycle kinematic. The pose of the wheeled vehicle at time t is represented by triple $(x(t), y(t), \theta(t))$, where $(x(t), y(t))$ is the pair of planar coordinates of the mid-point of the rear wheels axle expressed in the fixed right-handed reference frame $\langle W \rangle = \{O_w, X_w, Y_w, Z_w\}$. The robot orientation at time t is instead given by $\theta(t)$ that is the angle between the forward-oriented longitudinal symmetry axis of the

robot and axis X_w. Thus, for localization purposes, the robot state can be defined as $\boldsymbol{p} = [x, y, \theta]^T \in \mathcal{S}$, where the explicit reference to time variable t is omitted for the sake of simplicity and $\mathcal{S}$ denotes the state space. The inputs to the system are v and ω, i.e. the forward and the angular velocities of the robot, respectively. The kinematic model is then given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{7.1}$$

and it can be more compactly rewritten as:

$$\dot{\boldsymbol{p}} = g(\boldsymbol{p})\boldsymbol{u} = g_u(\boldsymbol{p})v + g_\omega(\boldsymbol{p})\omega, \tag{7.2}$$

where $\boldsymbol{p} = [x, y, \theta]^T$ and $\boldsymbol{u} = [v, \omega]^T$. Observe that (7.2) is a *driftless system.*

7.1.2 Measurement Model

As briefly explained at the beginning of Section 7.1, the localization relies on two measurement devices, i.e., a front UHF-RFID reader and two incremental encoders installed on the rear wheels. The angular velocities ω_r and ω_l of the right and left rear wheels can be expressed as a function of the robot forward and angular velocities v and ω, i.e.

$$\omega_r = \frac{v}{r} + \frac{l\omega}{2r} \text{ and } \omega_l = \frac{v}{r} - \frac{l\omega}{2r}, \tag{7.3}$$

where r is the wheels radius and l is the rear inter-axle length. If $\Delta_r = \omega_r T_s$ and $\Delta_l = \omega_l T_s$ denote the angular displacements of the right and left wheels, respectively, measured by the encoders in a sampling period T_s, by inverting (7.3) the incremental encoder measurement model for system (7.1) can be defined as

$$\begin{bmatrix} vT_s \\ \omega T_s \end{bmatrix} = \boldsymbol{h}^e(\Delta_r, \Delta_l). \tag{7.4}$$

The RFID system is able to measure the phase of the signals backscattered by each RFID tag [90]. After the i-th tag is energised by the antenna, the electromagnetic wave backscattered by this tag is affected by a nearly constant phase shift δ_i^ϕ. Therefore, if λ is the wavelength of the RF signal and d_i is the distance between the reader antenna and the i-th tag, the phase delay of the signal received by the reader is

$$\phi_i = \frac{-4\pi d_i}{\lambda} + \delta_i^\phi. \tag{7.5}$$

Assuming that locally $\dot{\delta}_i^\phi \approx 0$, it follows from (7.5) that $\dot{d}_i = \frac{\lambda}{4\pi}\dot{\phi}_i$. By rewriting (7.5) in terms of the euclidean distance to the i–th tag, we have

$$d_i^m = d_i + \delta_i^d, \tag{7.6}$$

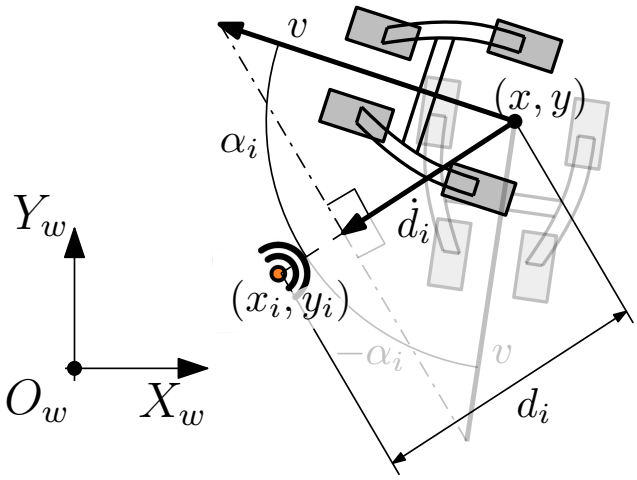


Figure 7.1: Schematic representation of the RFID measurement process assuming that the RFID antenna is placed in the middle of the rear wheels.

where δ_i^d is a function of δ_i^ϕ, and hence $\dot{d}_i^m = \dot{d}_i$. From Fig. 7.1, it follows that the actual distance d_i between the robot and the i–th tag expressed as a function of the elements of state vector $\boldsymbol{p}$ is

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}, \tag{7.7}$$

where (x_i, y_i) are the Cartesian coordinates of the i–th tag in $\langle W \rangle$. Therefore, it results that

$$\dot{d}_i = h_i(\boldsymbol{p}) = \dot{d}_i^m = \frac{\cos(\theta)(x - x_i) + \sin(\theta)(y - y_i)}{\sqrt{(x - x_i)^2 + (y - y_i)^2}} v. \tag{7.8}$$

As depicted in Fig. 7.1, $\dot{d}_i^m$ is the component of the robot forward velocity along the line connecting the point representing the robot position (namely the mid-point of the rear wheels axle) and the point where the i–th tag is located. As a consequence, if α_i denotes the angle between v and d_i (as depicted in Fig. 7.1) (7.8) inherently returns ambiguous results because $\dot{d}_i = v\cos(\alpha_i) = v\cos(-\alpha_i)$. This means that only an ambiguous information on the robot actual velocity v can be retrieved from (7.8) since angles θ and $\theta + 2\alpha_i$ return the same $\dot{d}_i^m$ value.

7.2 Observability Analysis

The design of a robot position estimator for the problem described in Section 7.1 requires a preliminary *observability* analysis to prove that the initial position of the robot $\boldsymbol{p}(t_0) \triangleq \boldsymbol{p}_0$ can be reconstructed by using both the sequence of input values (7.4) (namely the forward and angular velocity values returned by encoders)

and the distance variations between robot and tags given by (7.8). Note that when it is not possible to localize the system using all the available measures from a single fixed position (i.e. static observability) this analysis is mandatory. For the problem at hand, triangulation does not work when the static distance measures (7.6) are considered, even using an arbitrary number of RFID tags at known locations. This is due to the unknown distance term δ_i^d in (7.6). Moreover, even if δ_i^d were known, the vehicle orientation θ could not be observed in any case. Thus, since system and measurement models are nonlinear, a proper nonlinear observability analysis is needed. Given vector $\boldsymbol{h}(\boldsymbol{p}) = [h_1(\boldsymbol{p}), .., h_n(\boldsymbol{p})]^T$ consisting of n nonlinear measurement functions (7.8) associated with distinct tags, according to [3], the system is observable if $\boldsymbol{p}_0$ can be uniquely determined from $\boldsymbol{h}(\boldsymbol{p}_0)$ and its derivative at t_0. The time derivatives of the measurement functions along the system dynamics, i.e. the vector fields $g_u(\boldsymbol{p})$ and $g_\omega(\boldsymbol{p})$ in (7.2), are formally obtained through Lie-differentiation. In particular, the j-th Lie derivative $L_{g_\star}^j h_i(\boldsymbol{p})$ of measurement function $h_i(\boldsymbol{p})$ along vector field $g_\star$ is equal to

$$L_{g_\star}^j h_i(\boldsymbol{p}) = \nabla_{\boldsymbol{p}} L_{g_\star}^{j-1} h_i(\boldsymbol{p}) g_\star(\boldsymbol{p}), \tag{7.9}$$

where $\nabla_{\boldsymbol{p}}$ stands for the gradient over the robot state $\boldsymbol{p}$ and $L_{g_\star}^0 h_i(\boldsymbol{p}) = h_i(\boldsymbol{p})$. Let us formally describe the observation space as $\mathcal{O} = \text{span}[L_{g_\star}^0 h(\boldsymbol{p})^T, L_{g_\star}^1 h(\boldsymbol{p})^T, .., L_{g_\star}^r h(\boldsymbol{p})^T]^T$, where $L_{g_\star}^j h(\boldsymbol{p}) = [L_{g_\star}^j h_1(\boldsymbol{p})^T, L_{g_\star}^j h_2(\boldsymbol{p})^T, .., L_{g_\star}^j h_n(\boldsymbol{p})^T]^T$. If there is an injective function mapping the state $\boldsymbol{p}$ in a neighborhood of $\boldsymbol{p}_0$ into the measurement space, then the state $\boldsymbol{p}$ is *locally observable*. This injective function exists if the rank of the space spanned by the gradients of the elements of $\mathcal{O}$ (i.e. $\partial\mathcal{O}$) has the same dimension as the system state $\mathcal{S}$, i.e. if $\dim\{\partial\mathcal{O}\} = \dim\{\text{span}\{\nabla_{\boldsymbol{p}}\mathcal{O}\}\} = \dim\{\mathcal{S}\}$. Strictly speaking, this analysis aims at discovering if, when the robot moves (hence the gradient changes), there exists a unique evolution of the output functions from each possible initial state (hence the equivalence of the space dimensions). In general, the rank of $\partial\mathcal{O}$ is constant in $\mathcal{S}$ except for a non-compact set of singular points, where the rank is smaller [62].

Therefore, nonlinear observability just requires to have a locally injective function. In the next section, we will present the observability analysis assuming that the measurement data are not affected by uncertainty, since observability is indeed a structural system property. Since the problem with one tag only is clearly unobservable, the analysis for two tags is reported in the following.

Let us assume to have two RFID tags located (without loss of generality) in $(x_1, y_1) = (0, -l)$ and $(x_2, y_2) = (0, l)$ of $\langle W \rangle$. The vectorial measurement function in this case becomes

$$h(\boldsymbol{p}) = \begin{bmatrix} h_1(\boldsymbol{p}) \\ h_2(\boldsymbol{p}) \end{bmatrix} = \begin{bmatrix} \frac{\cos(\theta)(x)+\sin(\theta)(y+l)}{\sqrt{x^2+(y+l)^2}} v \\ \frac{\cos(\theta)(x)+\sin(\theta)(y-l)}{\sqrt{x^2+(y-l)^2}} v \end{bmatrix}. \tag{7.10}$$

By computing the Lie-derivative up to the first order with (7.9) for $j = 1$, the

Table 7.1: Conditions for matrix singularity for each matrix O_i constructed in a sequential way.

Singularity set	x	y	θ	ω
Υ_1	$\in \mathbb{R}$	l	$\in [0, 2\pi)$	$\in \mathbb{R}$
	$\in \mathbb{R}$	$x\tan(\theta) - l$	$\in [0, 2\pi)$	$\in \mathbb{R}$
	$\in \mathbb{R}$	$x\tan(\theta) + l$	$\in [0, 2\pi)$	$\in \mathbb{R}$
$\Upsilon_1 \cap \Upsilon_2$	$\in \mathbb{R}$	$x\tan(\theta) - l$	$\in [0, 2\pi)$	$\in \mathbb{R}$
	$\in \mathbb{R}$	$x\tan(\theta) + l$	$\in [0, 2\pi)$	$\in \mathbb{R}$
$\Upsilon_1 \cap \Upsilon_2 \cap \Upsilon_3$	$\in \mathbb{R}$	$x\tan(\theta) + l$	$\in [0, 2\pi)$	0
	$\in \mathbb{R}$	$x\tan(\theta) - l$	$\in [0, 2\pi)$	$\in \mathbb{R}$
	$\in \mathbb{R}$	l	0	$\in \mathbb{R}$
$\Upsilon_1 \cap \Upsilon_2 \cap \Upsilon_3 \cap \Upsilon_4$	$\in \mathbb{R}$	$x\tan(\theta) + l$	$\in [0, 2\pi)$	0
	$\in \mathbb{R}$	$x\tan(\theta) - l$	$\in [0, 2\pi)$	0
	$\in \mathbb{R}$	l	0	$\in \mathbb{R}$
	$\in \mathbb{R}$	$-l$	0	$\in \mathbb{R}$

following 4×3 matrix results

$$O = \begin{bmatrix} L^0_{g_\star} h_1(\boldsymbol{p}) \\ L^0_{g_\star} h_2(\boldsymbol{p}) \\ L^1_{g_\star} h_1(\boldsymbol{p}) \\ L^1_{g_\star} h_2(\boldsymbol{p}) \end{bmatrix}. \tag{7.11}$$

The rows of matrix (7.11) are the generators of space $\partial\mathcal{O}$. Notice that each row of $\mathcal{O}$ is a function of $\boldsymbol{p}$, v and ω due to (7.1) and (7.2). The system is then locally observable if $\text{rank}(O) = 3$. To test this condition, first the four 3×3 matrices obtained by selecting three rows of O at a time can be built. Then, the conditions that nullify each determinant can be determined. If there is a common condition for the three determinants, then the system is not locally observable as $\text{rank}(O) < 3$. Let matrix O_1 be composed by rows 1,2,3 of O, O_2 by rows 1,2,4, O_3 by 1,3,4 and O_4 by 2,3,4. We assume in this analysis that $v \neq 0$, i.e. the robot is moving with a certain forward velocity, since for $v = 0$ the determinants of all matrices are equal to zero. Let $\Upsilon_i \subset \mathcal{S} \times \mathbb{R}$ be the set of points where $\det O_i(s, \omega) = 0$, for $i = 1, 2, 3, 4$, i.e. the conditions for matrix singularity. The result of this analysis is subsumed in Tab. 7.1 following a sequential construction, i.e. considering only the common singularity conditions. From the analysis of the singularities for $\Upsilon_1 \cap \Upsilon_2 \cap \Upsilon_3 \cap \Upsilon_4$, it follows that $\text{rank}(O) < 3$ when $y = x\tan(\theta) + l$ or $y = x\tan(\theta) - l$ and $\omega = 0$. This condition corresponds to the case when the robot is moving along a straight line. In fact, the last two conditions, i.e. $y = \pm l$ and $\theta = 0$, are generally comprised in the previous two cases when $\omega = 0$, while for $\omega \neq 0$, they are isolated points.

To verify if this kind of paths are indeed not observable, the Lie derivatives of order higher than one should be computed. Unfortunately, this method makes the

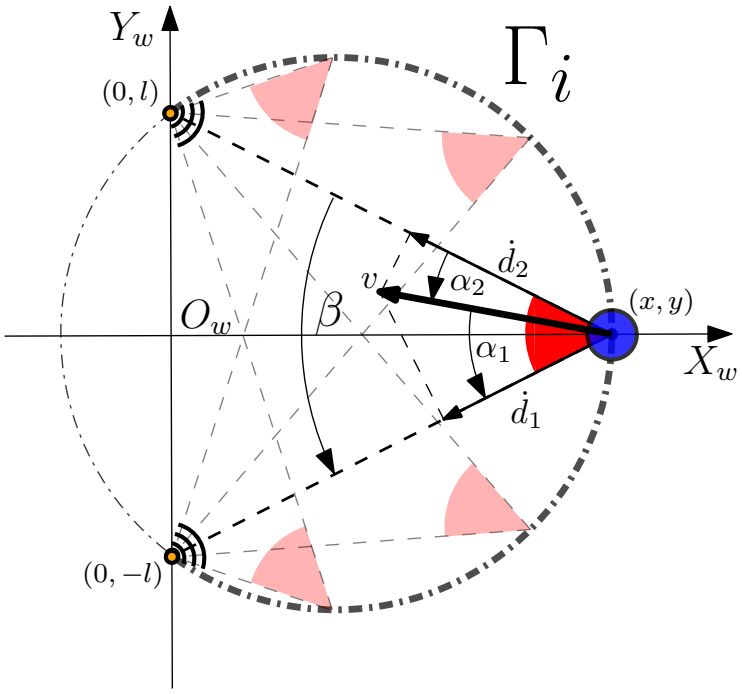


Figure 7.2: Ambiguous loci Γ_i derived for the i-th configuration in (7.12).

problem computationally intractable. So, a geometric approach has been adopted to bypass this issue. Indeed, if there exists a nonlinear invertible mapping that, using (7.10) and its derivative, is able to uniquely identify the vehicle position in the previous two rank conditions, then the system is observable. In particular, if angles α_i defined in Section 7.1.1 are expressed as $\alpha_i = \left|\arccos\left(\frac{\dot{d}_i}{v}\right)\right|$, for $h_1(\boldsymbol{p})$ and $h_2(\boldsymbol{p})$ in (7.8), 4 possible angular configurations result, i.e.

$$\begin{aligned} c_1 &= [\alpha_1^1, \alpha_2^1] = [\alpha_1, \alpha_2], \\ c_2 &= [\alpha_1^2, \alpha_2^2] = [-\alpha_1, -\alpha_2], \\ c_3 &= [\alpha_1^3, \alpha_2^3] = [-\alpha_1, \alpha_2], \\ c_4 &= [\alpha_1^4, \alpha_2^4] = [\alpha_1, -\alpha_2]. \end{aligned} \tag{7.12}$$

If angles are regarded as positive in counterclockwise directions and if β denotes the angle between the two straight lines passing through the points where robot and either RFID are located. (as depicted in Fig. 7.2), an index i must exist for which $\beta_i = \alpha_1^i - \alpha_2^i = \beta_i$. By geometric construction, the locus of the robot's positions associated with the same β_i belongs to the same arc of a circle, labeled as Γ_i in Fig. 7.2. In this way, four arcs can be identified. Each one of them corresponds to one of the 4 angular configurations in (7.12). The radius of such circles is $R_i =$

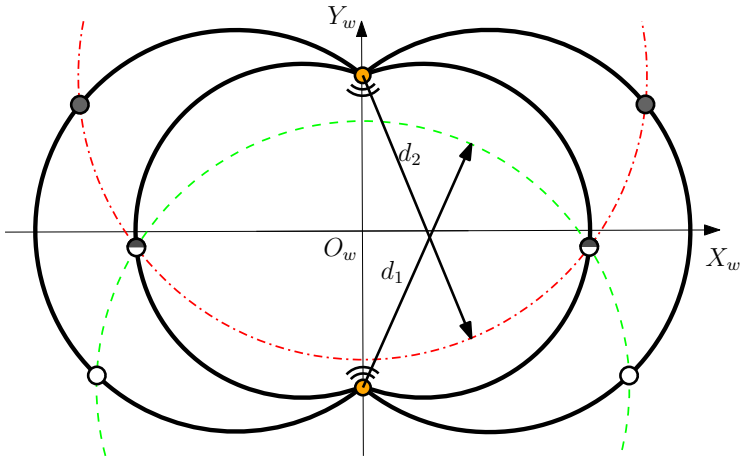


Figure 7.3: Reduction from 4 to 2 ambiguous locations by retrieving the actual values of d_1 and d_2.

$\frac{1}{2}\sqrt{\frac{l^2(\tan(\beta_i/2)^2+1)^2}{\tan(\beta_i/2)^2}}$, while their centers have coordinates $\left(\frac{l(\tan(\beta_i/2)^2-1)}{\tan(\beta_i/2)}, 0\right)$ in $\langle W\rangle$. Hence, the robot position belongs to the locus $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$, for $i = 1, 2, 3, 4$.

It can be shown through geometric and analytic manipulation that the relationship between $\dot{h}_i(\boldsymbol{p})$ (for the four ambiguous configurations in (7.12)) and the distances from the two RFID tags is described by the following expression, i.e.,

$$\begin{bmatrix} \dot{h}_1^i(s) \\ \dot{h}_2^i(s) \end{bmatrix} = \begin{bmatrix} \frac{\sin\alpha_1^i v}{d_1} - \sin\alpha_1^i v\omega \\ \frac{\sin\alpha_2^i v}{d_2} - \sin\alpha_2^i v\omega \end{bmatrix}. \tag{7.13}$$

Assuming to estimate $\dot{h}_i(\boldsymbol{p})$ as the robot moves (recall that in this analysis $v \neq 0$), if such values are denoted with $\hat{\dot{h}}_i(\boldsymbol{p})$ and measurement uncertainty is negligible, it can be shown using (7.13) that the distances between the robot and the RFID tags are given by

$$\begin{bmatrix} \hat{d}_1^i \\ \hat{d}_2^i \end{bmatrix} = \begin{bmatrix} \frac{\sin\alpha_1^i v}{\hat{\dot{h}}_1(s)+\sin\alpha_1^i v\omega} \\ \frac{\sin\alpha_2^i v}{\hat{\dot{h}}_2(s)+\sin\alpha_2^i v\omega} \end{bmatrix}. \tag{7.14}$$

To analyze local observability, the condition of interest in Tab. 7.1 is when $\omega = 0$. In this case, distances collapse to the actual solutions d_1 and d_2. Given two circles with their respective disjunct centres (corresponding to RFID tag positions) they will intersect in at most two points. Therefore, we can have at most two solutions in symmetric positions, belonging to two different Γ_i loci (i.e., highlighted with a half-white and half-grey circle markers in Fig. 7.3). Since the two positions are far apart and symmetric, the robot is locally observable with two tags. However, two singular cases exist. The first one occurs when the distance between the RFID tags

is negligible compared to the distance from the robot, namely when $\sqrt{x^2+y^2} \to \infty$. In this case, the derivatives in (7.13) tend to zero. The second singular case occurs when both $\alpha_1 = 0$ and $\alpha_2 = 0$, namely when the robot moves along the straight line joining the two RFID tags with $\omega = 0$. In both cases, no local observability is possible. Hence, any estimator will fail.

One additional outcome of this analysis (here not reported for space constraints), is that the robot is observable with at least 3 RFID tags that are not collinear, as it will be shown in Section 7.3.

7.3 Simulation Results

In this Section the results obtained with an observer from the Kalman Filter family are reported. In particular, due to the strong nonlinearity of both system and measurement models, a UKF is preferable [64]. In all the simulations, Gaussian zero-mean uncertainty contributions are considered. Assuming a sampling period $T_s = 10$ ms, the standard deviations of the forward and angular velocity fluctuations are 5 cm/s and 0.17 rad/s, respectively. The RFID-based distance measurements in (7.8) are sampled with the same period T_s. Measurement standard uncertainty is set to 5 cm/s. In all tests, robot forward velocity is set to $v = 1$ m/s, while the state of the system is initialized randomly (using independent Gaussian probability density functions - PDFs) around a given value $\boldsymbol{p}_0$ with standard deviations equal to 0.5 m along axes X_w and Y_w and 0.17 rad as far as the orientation angle θ is concerned.

7.3.1 Observability proof

In a first set of simulations, the sensing range of the RFID reader is supposed to be unlimited and four different setups are considered, as subsumed in Tab. 7.2. To confirm the validity of the observability analysis reported in Section 7.2, the joint posterior PDF has been computed for a single pair of measures (i.e., $h(\boldsymbol{p}) = [-0.3162, -0.9487]$ m/s) related to two RFID tags. By marginalising the joint PDF along θ, the resulting density for the (x, y) Cartesian coordinates is reported in Fig. 7.4. The red arcs on top of the PDF correspond to the ambiguity circles derived analytically in Section 7.2.

The first trajectory considered in simulations corresponds to the unobservable direction with 2 tags. The robot moves straight ($\omega = 0$ rad/s) along the line joining the first two tags. In particular, the robot moves along axis Y_w from $\boldsymbol{p}(t_0) = [0, 20, -\pi/2]^T$ to $\boldsymbol{p}(t_f) = [0, -20, -\pi/2]^T$, where $t_0 = 0$ and $t_f = 40$ seconds. The Root Mean Square Errors (RMSE) r_x, r_y and r_θ associated with state variables x, y and θ for 100 simulations in this scenario are reported in Fig. 7.5. Observe that at least three RFID tags are needed for UKF to converge, in accordance with the analysis reported in Section 7.2.

Table 7.2: Different set-up scenarios and RFID tag positions.

Set-up		RFID 4			
		RFID 3			
		RFID 2			
		RFID 1			
		Tag 1	Tag 2	Tag 3	Tag 4
[m]	x	0	0	5	-5
	y	5	-5	0	5

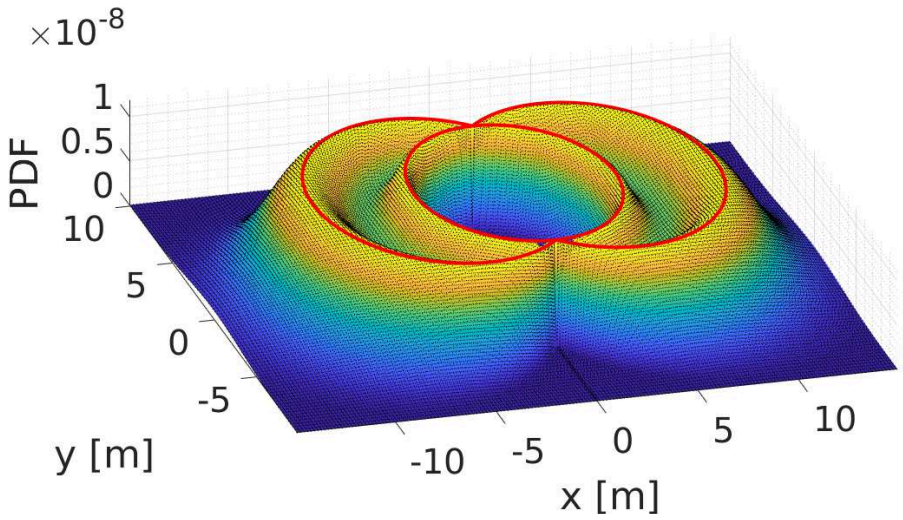


Figure 7.4: Joint PDF for the (x, y) Cartesian coordinates obtained from a single pair of measures with respect to two RFID tags.

In the second case study, the robot again moves along a straight line (i.e., axis X_w in this case) starting from $\boldsymbol{p}(t_0) = [-50, 2, 0]^T$ to $\boldsymbol{p}(t_f) = [50, 2, 0]^T$, where $t_0 = 0$ and $t_f = 100$ seconds. Again, the RMSE values computed over 100 simulations show that if one tag only is considered, the UKF does not converge, as stated in Section 7.2 (see Fig. 7.6). On the contrary, when at least two tags are used, the RMSE tends asymptotically to small values (i.e., the UKF converges), thus confirming that the system is observable. It is worth emphasizing that if $\omega \neq 0$ observability is always achieved even with 2 tags. This is confirmed by the simulation results shown in Fig. 7.7 for a circular trajectory. Again, the RMSE values tend to diverge when just 1 RFID tag is considered (due to dead reckoning), whereas they tend to converge when at least 2 tags are deployed.

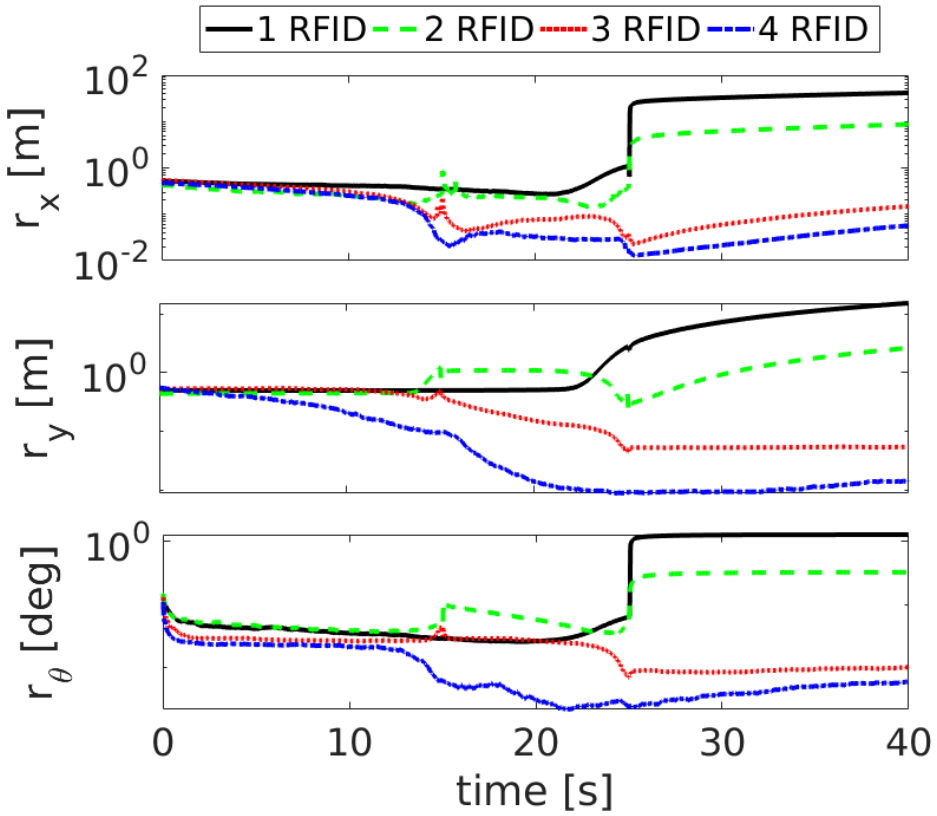


Figure 7.5: RMSE values of the state variables estimated by the UKF in the four cases of Tab. 7.2, when the robot moves straight along axis Y_w (which is unobservable in the scenario with two tags).

7.3.2 Realistic scenario

To complete the analysis, we have simulated a realistic scenario based on a 100×10 m corridor, where an RFID reader with a maximum range of 5 m is used. The RFID tags are placed over square-patterned regular grids of different granularity. In particular, the distance between pairs of adjacent tags ranges from a minimum of 1 m to a maximum of 10 m. The box-and-whiskers plot of the UKF-based localisation errors computed over 100 randomly generated trajectories is shown in Fig. 7.8 for various grid edge lengths. These results subsume the observability analysis and prove the practicality of this localization approach in realistic scenarios. Of course, localization uncertainty tends to grow as the grid of RFID tags become coarser.

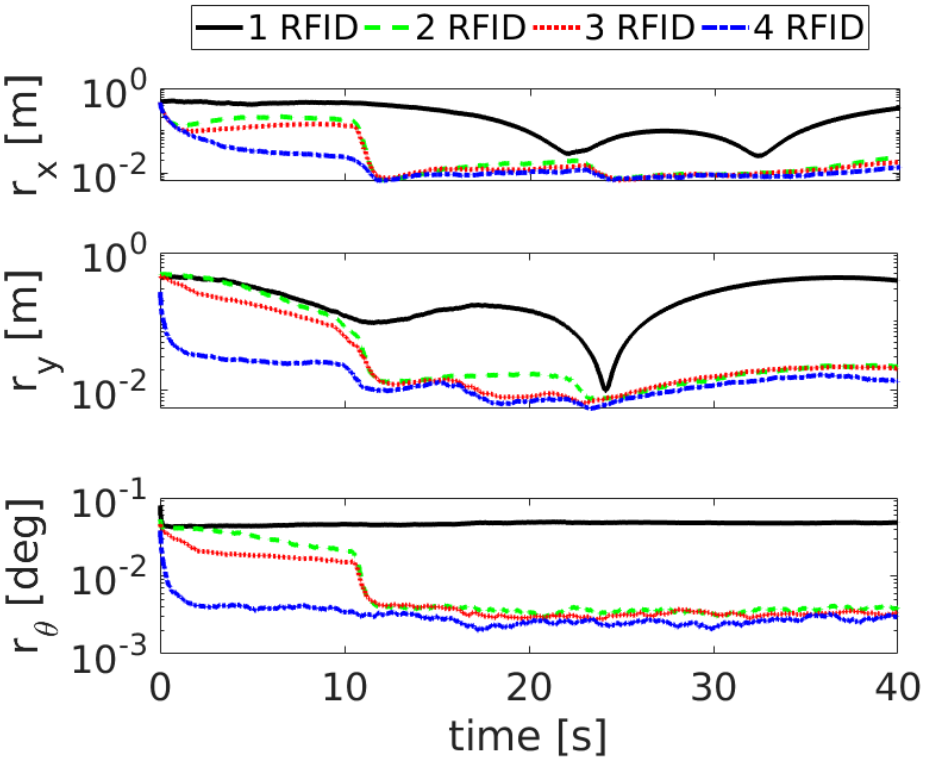


Figure 7.6: RMSE values of the state variables estimated by the UKF in the four cases of Tab. 7.2, when the robot moves straight along axis X_w.

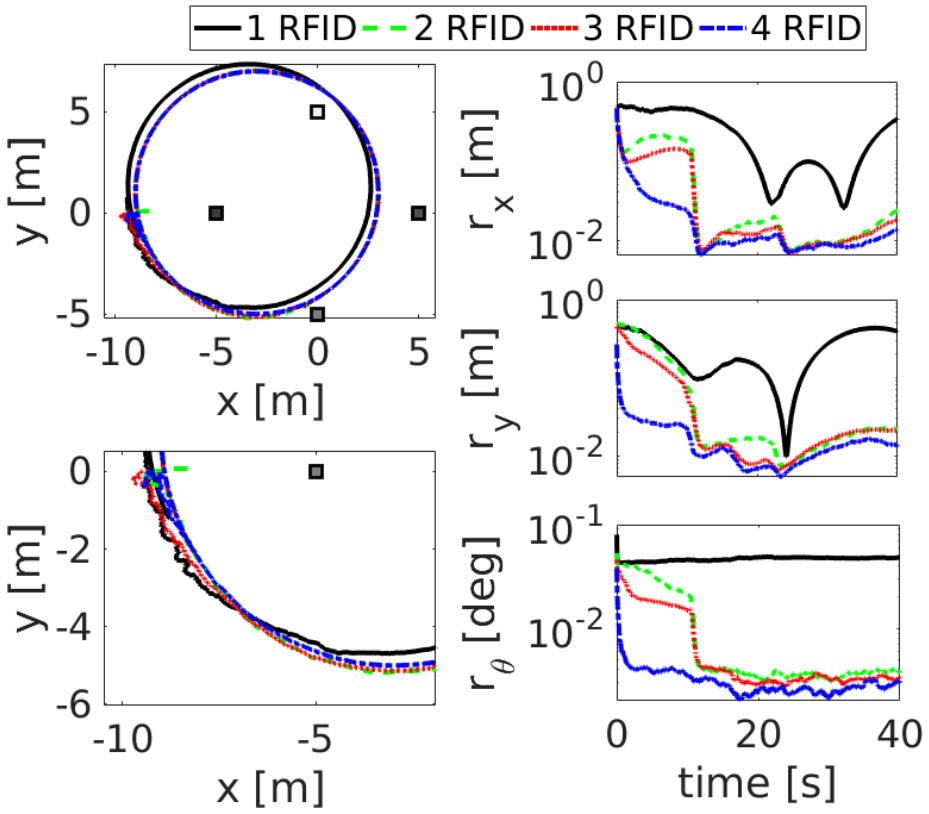


Figure 7.7: On the left: Circular trajectory centred in $(-3, 1)$ m and with a radius of 6 m and excerpt of the estimated trajectory at the beginning of the motion. On the right: RMSE of the state variables estimated by the UKF in the four cases of Tab. 7.2 when the robot moves along the depicted circular trajectory.

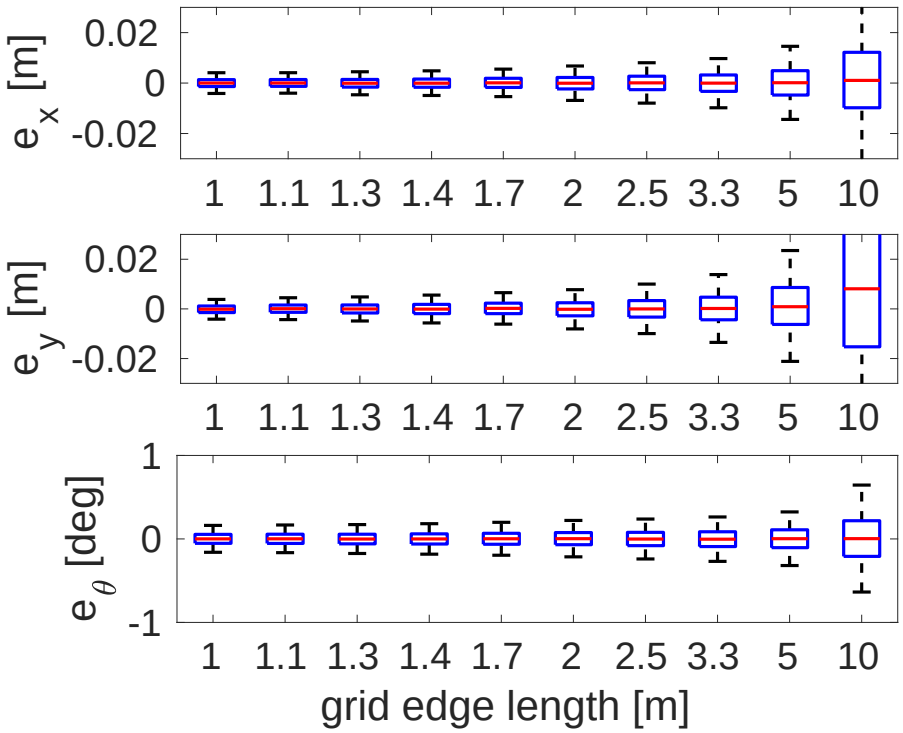


Figure 7.8: Box-and-whiskers plot of the localization error in the corridor scenario for different distances between pairs of adjacent RFID tags placed over square-patterned regular grids.

Chapter 8

Conclusion

In this thesis, we proposed a set of algorithms and approaches to support different kinds of localization methodologies. The general framework we identified is solving an optimization problem aiming at minimizing a generic cost function (e.g., infrastructural cost, energy consumption) while preserving the desired performance levels (e.g., the distance of the estimated robot pose from the real pose). This is a novel approach since commonly in the state of the art the precision of the localization system is evaluated a posteriori with limited possibility to adapt its performance depending on the task of the robot. This flexibility was achievable due to Baesyan filters. Assuming that these filters are based on a proper characterization of the system, they ensure a coherent estimation of the uncertaintyy.

Chapter 4 presents an approach that can extend the localization systems based on Baesyan filters of mobile robots by respecting stochastic guarantees on the accuracy. This is done by deploying a minimal set of landmarks in precise positions inside the environment, and by endowing the robot with an exteroceptive sensor able to estimate the global pose of the robot when it detects an artificial landmark. Indoor positioning techniques for mobile agents often rely on this approach where systems based on natural landmarks fail or do not provide enough accuracy. Usually, landmarks are placed randomly or following the designer intuition. However these approaches cannot ensure any guarantees on the localization performance. Most of the related papers try to solve this problem by deploying a predefined number of landmarks in order to maximize the covered area. In addition, most of these approaches assume an homogeneous detection range and/or focus on static condition, neglecting the prior on the pose distribution given from the Bayesian filters. In contrast, the technique presented in Chapter 4, taking into account the accuracy requirements and the knowledge of the agents trajectories inside the environment, gives the number and position of landmarks as an output. The landmarks positions are chosen by considering the prior information given by the Bayesian localization filter. The key advantage of this solution, with respect to the previous approaches, is the capability of selecting a very low number of landmarks, while ensuring the

required quality of service. What is more, although our analysis has been restricted to trapezioidal and triangular areas, any shape of the sensor detection range can be easily integrated. In Chapter 4, the problem of optimal landmark placement first is formalized in the framework of logic synthesis, and then it is solved through a greedy approach, which keeps into account the possible paths of the agents' within the environment considered. Even if the greedy algorithm generally does not converge to the globally optimal solution, multiple simulation results show that the number of landmarks deployed with the adopted heuristic approach is just slightly greater than the lower bound to the actual optimal solution. On the contrary, a much larger amount of randomly deployed landmarks is needed to achieve the same positioning uncertainty. In addition, since the greedy algorithm is computationally light, it can be effectively used even for large environments. The proposed approach was validated on the field using a reasonable body of experimental data in a real–life scenario.

Chapter 5 describes an application in which control and localization are tightly coupled. This coupling contributes to the reduction of localization requirements and improves the system robustness. The system, given the localization accuracy, can select the more stable control action. At the heart of our approach is the idea of a control strategy for shifting control authority between a human user and a controller for robotic navigation assistance. In the framework we have set up in Chapter 5, the problem of assisting a person can be seen as an instance of path-following problems, for which most of the available solutions currently require an accurate localization of the vehicle in the environment, which is not always feasible. Our idea to solve the problem is to use a precise localization only when needed (i.e., in the proximity of complex decision points) and leave the guidance responsibility to the user when the navigation task is relatively easy to do(i.e., when the user is simply required to move along a direction dictated by the environment). This strategy requires an effective way to shift the control authority to the user when the localization precision is low, and returns it to the controller when it increases (i.e., when more landmarks are in sight) or when the user is acting against the control goals (i.e., turning backward). This idea has been formalized with a hybrid control design. Chapter 5 sets up the theoretical framework for this controller and shows its efficacy through extensive simulations and experimental results. In this chapter we highlight several open problems that we consider interesting and could deserve further investigations. From the theoretical point of view, the most interesting problem that needs to be addressed is to offer "certifiable" performance guarantees based on the knowledge of the vehicle and the environment. Another important goal is to test the presented authority-sharing approach with a number of actual users and carry out a quantitative and qualitative study on their performance and impressions.

Chapter 6 proposes a localization approach for mobile robots that relies on fusing the relative information coming from the encoders with absolute measurements coming from an external UWB infrastructure, triggered adaptively only if and when

the estimated positioning accuracy fails to meet the user-specified requirements. The proposed mathematical framework directly estimates uncertainty via the a-posteriori probability density function and enables stochastic guarantees on the system performance. Simulation and real-world experiments confirm the soundness of our technique and its effectiveness in reducing the UWB sampling rate w.r.t. a fixed periodic solution, with direct benefits in terms of scalability and energy consumption. Moreover, our system allows for low-accuracy ranging schemes, which further improve these metrics. Several possible research directions lie before us for ulterior exploration and exploitation. For example the dependency of the technique from the motion patterns of the robots, the conditions under which it yields the optimal sampling rate and its scalability in scenarios where multiple robot are deployed have to be explored. From a system point of view, the scalability of the technique in scenarios with multiple robots and its dependency on their motion patterns have to be analyzed. In these scenarios, a non-negligible interference probability between the transmission of the robots has to be modeled, and proper scheduling of the transmission has to be defined.

Chapter 7 presents an observability analysis for the localization problem of a non-holonomic mobile robot relying on RFID infrastructure. In particular, we have laid the groundwork for the development of a system based on phase measurements as exteroceptive information since they are more sensitive to the motion of the system compared to the classically used RSSI signal. In contrast to state of the art systems based on phase measurements, we do not require a per tag characterization of the phase offset, and to the best of our knowledge, this is the first time that this analysis is performed for this specific application. The proposed analysis is particularly challenging due to the inherent ambiguity associated with the phase measurement of RF signals backscattered by UHF-RFID tags. As a result of this analysis, singular configurations and the minimum number of tags that need to be detected simultaneously for localization purposes are derived. The validity of the analysis and the practicality of the localization approach have been verified using an ad-hoc Unscented Kalman Filter (UKF). A possible future work could be the implementation and experimental validation of the position estimation algorithm in a real scenario, currently confirmed by some preliminary simulations.

References

[1] Optitrack, 2019. `https://optitrack.com/motion-capture-robotics/`, Sidst set 09/11/2016.

[2] ACANTO: A CyberphysicAl social NeTwOrk using robot friends. http://www.ict-acanto.eu/acanto, February 2015. EU Project.

[3] A. Isidori A. Krener. Linearization by output injection and nonlinear observers. *Syst. Control Lett.3*, 1:47–52, 1983.

[4] A. P. Aguiar and J. P. Hespanha. Trajectory-Tracking and Path-Following of Underactuated Autonomous Vehicles With Parametric Modeling Uncertainty. *IEEE Transactions on Automatic Control*, 52(8):1362–1379, Aug 2007.

[5] M. Andreetto, S. Divan, F. Ferrari, D. Fontanelli, L. Palopoli, and F. Zenatti. Simulating passivity for robotic walkers via authority-sharing. *IEEE Robotics and Automation Letters*, 3(2):1306–1313, April 2018.

[6] M. Andreetto, S. Divan, D. Fontanelli, and L. Palopoli. Passive robotic walker path following with bang-bang hybrid control paradigm. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1054–1060, Oct 2016.

[7] M. Andreetto, S. Divan, D. Fontanelli, and L. Palopoli. Harnessing Steering Singularities in Passive Path Following for Robotic Walkers. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 2426–2432, Singapore, May 2017. IEEE.

[8] M. Andreetto, S. Divan, D. Fontanelli, and L. Palopoli. Path following with authority sharing between humans and passive robotic walkers equipped with low-cost actuators. *IEEE Robotics and Automation Letters*, 2(4):2271–2278, Oct 2017.

[9] M. Andreetto, S. Divan, D. Fontanelli, and L. Palopoli. Path Following with Authority Sharing between Humans and Passive Robotic Walkers Equipped with Low-Cost Actuators. *IEEE Robotics and Automation Letters*, 2(4):2271–2278, Oct. 2017.

[10] M. Andreetto, S. Divan, D. Fontanelli, L. Palopoli, and F. Zenatti. Path Following for Robotic Rollators via Simulated Passivity. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and System (IROS)*, Vancouver, Canda, Oct. 2017. IEEE/RSJ. To appear.

[11] Marco Andreetto, Paolo Bevilacqua, Stefano Divan, Daniele Fontanelli, David Macii, Valerio Magnago, , and Luigi Palopoli. Characterization of sensors for smart walker indoor localization. In *2018 Convegno Nazionale Sensori(CNS)*, 2018.

[12] Marco Andreetto, Stefano Divan, Francesco Ferrari, Daniele Fontanelli, Luigi Palopoli, and Domenico Prattichizzo. Combining Haptic and Bang-Bang Braking Actions for Passive Robotic Walker Path Following. *Transaction on Haptics*, 2018. Submitted.

[13] A. Antonucci, V. Magnago, L. Palopoli, and D. Fontanelli. Performance assessment of a people tracker for social robots. In *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, May 2019.

[14] Y. Bar-Shalom, X.R. Li, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. Wiley-Interscience, 2001.

[15] Maximilian Beinhofer, Jörg Müller, and Wolfram Burgard. Near-optimal landmark selection for mobile robot navigation. In *IEEE Intl. Conf. on Robotics and Automation*, pages 4744–4749. IEEE, 2011.

[16] Maximilian Beinhofer, Jörg Müller, and Wolfram Burgard. Effective landmark placement for accurate and reliable mobile robot navigation. *ROBOT. AUTON. SYST.*, 61(10):1060–1069, Oct. 2013.

[17] Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning motion patterns of people for compliant robot motion. *Int. J. Robot. Res.*, 24(1):31–48, 2005.

[18] P. Bevilacqua, M. Frego, E. Bertolazzi, D. Fontanelli, L. Palopoli, and F. Biral. Path Planning maximising Human Comfort for Assistive Robots. In *IEEE Conference on Control Applications (CCA)*, pages 1421–1427, Buenos Aires, Argentina, Sept. 2016. IEEE.

[19] Mauro Boccadoro, Francesco Martinelli, and Stefano Pagnottelli. Constrained and quantized kalman filtering for an rfid robot localization problem. *Auton. Robots*, 29:235–251, 11 2010.

[20] D. Briese, H. Kunze, and G. Rose. High precision uwb-based 3d localization for medical environment. In *2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pages 1–5, Oct 2015.

[21] A. Buffi, D. Fontanelli, D. Macii, V. Magnago, A. Motroni, P. Nepa, and B. Tellini. Uhf-rfid localization: The problem of antenna phase center in phase-based methods. In *2019 13th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5, March 2019.

[22] A. Buffi, A. Motroni, P. Nepa, B. Tellini, and R. Cioni. A SAR-based measurement method for passive-tag positioning with a flying UHF-RFID reader. *IEEE Transactions on Instrumentation and Measurement*, pages 1–9, 2018.

[23] A. Buffi and P. Nepa. An rfid-based technique for train localization with passive tags. In *2017 IEEE International Conference on RFID (RFID)*, pages 155–160, May 2017.

[24] A. Buffi, P. Nepa, and F. Lombardini. A phase-based technique for localization of UHF-RFID tags moving on a conveyor belt: Performance analysis and test-case measurements. *IEEE Sensors Journal*, 15(1):387–396, Jan. 2015.

[25] Alice Buffi, Paolo Nepa, and Fabrizio Lombardini. A phase-based technique for localization of uhf-rfid tags moving on a conveyor belt: Performance analysis and test-case measurements. *Sensors Journal, IEEE*, 15:387–396, 01 2015.

[26] L. Cavanini, G. Cimini, F. Ferracuti, A. Freddi, G. Ippoliti, A. Monteriù, and F. Verdini. A qr-code localization system for mobile robots: Application to smart wheelchairs. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–6, Sep. 2017.

[27] T. Chang and A. Mehta. Optimal scheduling for resource-constrained multirobot cooperative localization. *IEEE Robotics and Automation Letters*, 3(3):1552–1559, July 2018.

[28] Ping Chen, Yu Bin Xu, Liang Chen, and Zhi An Deng. Survey of WLAN fingerprinting positioning system. *Applied Mechanics and Materials*, 380:2499–2505, Aug. 2013.

[29] K. S. Chong and L. Kleeman. Accurate odometry and error modelling for a mobile robot. In *Proceedings of International Conference on Robotics and Automation*, pages 2783–2788, April 1997.

[30] V Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18(1):39–41, feb 1975.

[31] P. Coelho and U. Nunes. Path-following control of mobile robots in presence of uncertainties. *IEEE Transactions on Robotics*, 21(2):252–261, April 2005.

[32] Alessio Colombo, Daniele Fontanelli, Axel Legay, Luigi Palopoli, and Sean Sedwards. Efficient customisable dynamic motion planning for assistive robots

in complex human environments. *Journal of Ambient Intelligence and Smart Environments*, 7(5):617–633, 2015.

[33] P. Corbalán, T. Istomin, and G. P. Picco. Poster: Enabling Contiki on Ultra-wideband Radios. In *Proc. of EWSN*, 2018.

[34] J. A. Corrales, F. A. Candelas, and F. Torres. Hybrid tracking of human operators using imu/uwb data fusion by a kalman filter. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 193–200, March 2008.

[35] I. J. Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, Apr 1991.

[36] DecaWave. DecaWave ScenSor EVB1000 Evaluation Board, 2013.

[37] DecaWave. DW1000 Data Sheet, 2016.

[38] E. DiGiampaolo and F. Martinelli. Mobile Robot Localization Using the Phase of Passive UHF RFID Signals. *IEEE Transactions on Industrial Electronics*, 61(1):365–376, Jan 2014.

[39] Emidio DiGiampaolo and Francesco Martinelli. A passive uhf-rfid system for the localization of an indoor autonomous vehicle. *IEEE Transactions on Industrial Electronics - IEEE TRANS IND ELECTRON*, 59:3961–3970, 10 2012.

[40] Emidio DiGiampaolo and Francesco Martinelli. Mobile robot localization using the phase of passive uhf rfid signals. *Industrial Electronics, IEEE Transactions on*, 61:365–376, 01 2014.

[41] H. Durrant-Whyte, D. Pagac, B. Rogers, M. Stevens, and G. Nelmes. Field and service applications - an autonomous straddle carrier for movement of shipping containers - from research to operational autonomous systems. *IEEE Robotics Automation Magazine*, 14(3):14–23, Sep. 2007.

[42] B. Dzodzo, Long Han, Xu Chen, Huihuan Qian, and Yangsheng Xu. Realtime 2D code based localization for indoor robot navigation. In *Proc. IEEE Int. Conference on Robotics and Biomimetics (ROBIO)*, pages 486–492, Shenzhen, China, Dec. 2013.

[43] A. F. C. Errington, B. L. F. Daku, and A. F. Prugger. Initial position estimation using rfid tags: A least-squares approach. *IEEE Transactions on Instrumentation and Measurement*, 59(11):2863–2869, Nov 2010.

[44] Francesco Farina, Daniele Fontanelli, Andrea Garulli, Antonio Giannitrapani, and Domenico Prattichizzo. Walking Ahead: The Headed Social Force Model. *PLOS ONE*, 12(1):1–23, 01 2017.

[45] Federal Communications Commission, Washington, DC, 2002. First Report and Order 02–48,.

[46] D. Fontanelli, A. Giannitrapani, L. Palopoli, and D. Prattichizzo. A Passive Guidance System for a Robotic Walking Assistant using Brakes. In *Proc. IEEE Int. Conf. on Decision and Control (CDC)*, pages 829–834, Osaka, Japan, 15-18 Dec. 2015. IEEE.

[47] D. Fontanelli, A. Giannitrapani, L. Palopoli, and D. Prattichizzo. A passive guidance system for a robotic walking assistant using brakes. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 829–834, Dec 2015.

[48] D. Fontanelli, D. Macii, P. Nazemzadeh, and L. Palopoli. Collaborative Localization of Robotic Wheeled Walkers using Interlaced Extended Kalman Filters. In *Proc. IEEE Int. Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, Taipei, Taiwan, May 2016. IEEE. Available online.

[49] Emil Fresk, Kristoffer Ödmark, and George Nikolakopoulos. Ultra wideband enabled inertial odometry for generic localization. *IFAC-PapersOnLine*, 50(1):11465–11472, 2017.

[50] Yulu Fu, Changlong Wang, Ran Liu, Gaoli Liang, Hua Zhang, and Shafiq Ur Rehman. Moving Object Localization Based on UHF RFID Phase and Laser Clustering. *Sensors*, 18(3), 2018.

[51] M. J. Gallant and J. A. Marshall. Two-dimensional axis mapping using LiDAR. *IEEE Trans. on Robotics*, 32(1):150–160, Feb. 2016.

[52] Nancy M Gell, Robert B Wallace, Andrea Z Lacroix, Tracy M Mroz, and Kushang V Patel. Mobility device use in older adults and incidence of falls and worry about falling: Findings from the 2011–2012 national health and aging trends study. *Journal of the American Geriatrics Society*, 63(5):853–859, 2015.

[53] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[54] J. Gonzalez, J. L. Blanco, C. Galindo, A. Ortiz-de-Galisteo, J. A. Fernandez-Madrigal, F. A. Moreno, and J. L. Martinez. Combination of uwb and gps for indoor-outdoor vehicle localization. In *2007 IEEE International Symposium on Intelligent Signal Processing*, pages 1–6, Oct 2007.

[55] A. Goswami, M. A. Peshkin, and J. E. Colgate. Passive robotics: an exploration of mechanical computation. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 279–284 vol.1, May 1990.

[56] Birgit Graf. An adaptive guidance system for robotic walking aids. *CIT*, 17:109–120, 2009.

[57] Soonshin Han, HyungSoo Lim, and JangMyung Lee. An efficient localization scheme for a differential-driving mobile robot based on rfid system. *Industrial Electronics, IEEE Transactions on*, 54:3362 – 3369, 01 2008.

[58] S. He and S. . G. Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys Tutorials*, 18(1):466–490, Firstquarter 2016.

[59] Y. Hirata, A. Hara, and K. Kosuge. Motion control of passive intelligent walker using servo brakes. *IEEE Transactions on Robotics*, 23(5):981–990, Oct 2007.

[60] Y. Hsieh, K. Young, and C. Ko. Effective maneuver for passive robot walking helper based on user intention. *IEEE Transactions on Industrial Electronics*, 62(10):6404–6416, Oct 2015.

[61] IEEE 802.15.4-2015, Standard for Low-Rate Wireless Networks.

[62] Alberto Isidori. *Nonlinear control systems*. Springer Science & Business Media, 2013.

[63] S. Jiang, C. Lin, K. Huang, and K. Song. Shared control design of a walking-assistant robot. *IEEE Transactions on Control Systems Technology*, 25(6):2143–2150, Nov 2017.

[64] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics, 1997.

[65] A. A. Khaliq, F. Pecora, and A. Saffiotti. Inexpensive, reliable and localization-free navigation using an RFID floor. In *European Conf. on Mobile Robots (ECMR)*, Lincoln, United Kingdom, Sep. 2015.

[66] Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pages 324–331, Arlington, Virginia, United States, 2005. AUAI Press.

[67] Miroslav Krstic, Ioannis Kanellakopoulos, and Peter V Kokotovic. *Nonlinear and adaptive control design*. Wiley, 1995.

[68] V. Kulyukin, A. Kutiyanawala, E. LoPresti, J. Matthews, and R. Simpson. iwalker: Toward a rollator-mounted wayfinding system for the elderly. In *2008 IEEE International Conference on RFID*, pages 303–311, April 2008.

[69] P. Kumar, L. Reddy, and S. Varma. Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks. In *2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN)*, pages 1–4, Dec 2009.

[70] M. Lahijanian, M. Svorenova, A. A. Morye, B. Yeomans, D. Rao, I. Posner, P. Newman, H. Kress-Gazit, and M. Kwiatkowska. Resource-performance tradeoff analysis for mobile robots. *IEEE Robotics and Automation Letters*, 3(3):1840–1847, July 2018.

[71] Jean-Paul Laumond. *Robot motion planning and control. Lectures Notes in Control and Information Sciences 229*, volume 3. 1998.

[72] D. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, March 1986.

[73] Geunho Lee, Takanori Ohnuma, and Nak Young Chong. Design and control of jaist active robotic walker. *Intelligent Service Robotics*, 3:125–135, 2010.

[74] U. Lee, J. Jung, S. Shin, Y. Jeong, K. Park, D. H. Shim, and I. s. Kweon. EureCar turbo: A self-driving car that can handle adverse weather conditions. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2301–2306, Oct 2016.

[75] Cheng-Kai Lu, Yi-Che Huang, and Cheng-Jung Lee. Adaptive guidance system design for the assistive robotic walker. *Neurocomputing*, 170:152–160, 2015.

[76] D. Macii, A. Colombo, P. Pivato, and D. Fontanelli. A Data Fusion Technique for Wireless Ranging Performance Improvement. *IEEE Trans. on Instrumentation and Measurement*, 62(1):27–37, January 2013.

[77] V. Magnago, P. Bevilacqua, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii. Optimal landmark placement for indoor positioning using context information and multi-sensor data. In *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, May 2018.

[78] V. Magnago, P. Corbalán, G.P. Picco, L. Palopoli, and D. Fontanelli. Robot localization via odometry-assisted ultra-wideband ranging with stochastic guarantees. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.

[79] V. Magnago, L. Palopoli, A. Buffi, B. Tellini, A. Motroni, P. Nepa, D. Macii, and D. Fontanelli. Ranging-free uhf-rfid robot positioning through phase measurements of passive tags. *IEEE Transactions on Instrumentation and Measurement*, pages 1–1, 2019.

[80] V. Magnago, L. Palopoli, D. Fontanelli, D. Macii, A. Motroni, P. Nepa, A. Buffi, and B. Tellini. Robot localisation based on phase measures of backscattered uhf-rfid signals. In *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, May 2019.

[81] V. Magnago, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii. A nearly optimal landmark deployment for indoor localisation with limited sensing. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2017.

[82] V. Magnago, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii. Effective landmark placement for robot indoor localization with position uncertainty constraints. *IEEE Transactions on Instrumentation and Measurement*, 68(11):4443–4455, Nov 2019.

[83] Valerio Magnago, Marco Andreetto, Stefano Divan, Daniele Fontanelli, and Luigi Palopoli. Ruling the control authority of a service robot based on information precision. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7204–7210. IEEE, 2018.

[84] Maria M. Martins, Cristina P. Santos, Anselmo Frizera-Neto, and Ramón Ceres. Assistive mobility devices focusing on smart walkers: Classification and review. *Robotics and Autonomous Systems*, 60(4):548–562, 2012.

[85] Maria M. Martins, Cristina P. Santos, Anselmo Frizera-Neto, and Ramón Ceres. Real time control of the asbgo walker through a physical human robot interface. 2014.

[86] D. Meyer-Delius, M. Beinhofer, A. Kleiner, and W. Burgard. Using artificial landmarks to reduce the ambiguity in the environment of a mobile robot. In *2011 IEEE International Conference on Robotics and Automation*, pages 5173–5178, May 2011.

[87] S.M. Moe, Z. Winy, A. Robert, and H. Scholtz. Impulse radio: How it works. *IEEE Commun Lett*, 2:20–90, 01 2006.

[88] Andreas Molisch, Yves Nakache, Philip Orlik, Jin Zhang, Yunnan Wu, Sinan Gezici, and S Kung. An efficient low-cost time-hopping impulse radio for high data rate transmission. 12 2003.

[89] Salama A. Mostafa, Mohd Sharifuddin Ahmad, and Aida Mustapha. Adjustable autonomy: A systematic literature review. *Artif. Intell. Rev.*, 51(2):149–186, February 2019.

[90] A. Motroni, P. Nepa, V. Magnago, A. Buffi, B. Tellini, D. Fontanelli, and D. Macii. SAR-based Indoor Localization of UHF-RFID Tags via Mobile Robot. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Nantes, France, Sept. 2018. IEEE.

[91] P. Nazemzadeh, D. Fontanelli, and D. Macii. Optimal placement of landmarks for indoor localization using sensors with a limited range. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Oct 2016.

[92] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli. Indoor Localization of Mobile Robots through QR Code Detection and Dead Reckoning Data Fusion. *IEEE/ASME Transactions on Mechatronics*, 22(6):2588–2599, Dec. 2017.

[93] Payam Nazemzadeh, Federico Moro, Daniele Fontanelli, David Macii, and Luigi Palopoli. Indoor Positioning of a Robotic Walking Assistant for Large Public Environments. *IEEE Trans. on Instrumentation and Measurement*, 64(11):2965–2976, Nov 2015.

[94] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

[95] Harry Duong Nguyen, Gim Soh, Shaohui Foong, and Kristin Wood. Localization of a miniature spherical rolling robot using imu, odometry and uwb. 08 2018.

[96] P. V. Nikitin, R. Martinez, S. Ramamurthy, H. Leland, G. Spiess, and K. V. S. Rao. Phase based spatial identification of uhf rfid tags. In *2010 IEEE International Conference on RFID (IEEE RFID 2010)*, pages 102–109, April 2010.

[97] Joseph O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, Inc., USA, 1987.

[98] Luigi Palopoli et al. Navigation Assistance and Guidance of Older Adults across Complex Public Spaces: the DALi Approach. *Intelligent Service Robotics*, 8(2):77–92, 2015.

[99] Sunhong Park and Shuji Hashimoto. Autonomous mobile robot navigation using passive rfid in indoor environment. *Industrial Electronics, IEEE Transactions on*, 56:2366 – 2373, 08 2009.

[100] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *IEEE 12th Intl. Conf. on Computer Vision*, pages 261–268, Sep. 2009.

[101] F. Potortì, A. Crivello, P. Barsocchi, and F. Palumbo. Evaluation of indoor localisation systems: Comments on the iso/iec 18305 standard. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7, Sep. 2018.

[102] S. Quinlan and O. Khatib. Elastic bands: connecting path planning and control. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 802–807 vol.2, May 1993.

[103] ISO/IEC Guide 98-3:2008. *Uncertainty of measurement – Part 3: Guide to the expression of uncertainty in measurement (GUM:1995)*. Jan. 2008.

[104] M. Rosenblatt. A central limit theorem and a strong mixing condition. *Proceedings of the National Academy of Sciences of the United States of America*, 42(1):43–47, 1956.

[105] Christopher Rouff and Mike Hinchey. *Experience from the DARPA Urban Challenge*. Springer Publishing Company, Incorporated, 2011.

[106] Paul Runkle, John W. McCorkle, Timothy R. Miller, and Matthew L. Welborn. Ds-cdma: the modulation technology of choice for uwb communications. *IEEE Conference on Ultra Wideband Systems and Technologies, 2003*, pages 364–368, 2003.

[107] T. Rupp and P. Levi. Optimized landmark arrangement for absolute localization-a practical approach. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 1, pages 448–453 vol.1, Oct 2000.

[108] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson. Landmark selection for vision-based navigation. *IEEE Transactions on Robotics*, 22(2):334–349, April 2006.

[109] Joaquin Salas and Jose Luis Gordillo. Placing artificial visual landmarks in a mobile robot workspace. In *Proceedings of the 6th Ibero-American Conference on AI: Progress in Artificial Intelligence*, IBERAMIA '98, pages 274–282, London, UK, UK, 1998. Springer-Verlag.

[110] T. Sasaki, D. Brscic, and H. Hashimoto. Human-Observation-Based Extraction of Path Patterns for Mobile Robot Navigation. *IEEE Transactions on Industrial Electronics*, 57(4):1401–1410, April 2010.

[111] Robert A. Scholtz. Multiple access with time-hopping impulse modulation. *Proceedings of MILCOM '93 - IEEE Military Communications Conference*, 2:447–450 vol.2, 1993.

[112] J. Schroeder, S. Galler, and K. Kyamakya. A low-cost experimental ultra-wideband positioning system. In *2005 IEEE International Conference on Ultra-Wideband*, pages 632–637, Sep. 2005.

[113] E.M. Sentovich et al. Sis: A system for sequential circuit synthesis. Technical Report UCB/ERL M92/41, EECS Department, University of California, Berkeley, 1992.

[114] B. Silva, Z. Pang, J. Åkerberg, J. Neander, and G. Hancke. Experimental study of uwb-based high precision localization for industrial applications. In *2014 IEEE International Conference on Ultra-WideBand (ICUWB)*, pages 280–285, Sep. 2014.

[115] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, Sept 2004.

[116] D Soetanto, L Lapierre, and A Pascoal. Adaptive, non-singular path-following control of dynamic wheeled robots. In *IEEE Conf. on Decision and Control*, volume 2, pages 1765–1770. IEEE, 2003.

[117] Xiang Song, Xu Li, Wencheng Tang, Weigong Zhang, and Bin Li. A Hybrid Positioning Strategy for Vehicles in a Tunnel Based on RFID and In-Vehicle Sensors. *Sensors*, 14(12):23095–23118, 2014.

[118] Hauke Strasdat, Cyrill Stachniss, and Wolfram Burgard. Which landmark is useful? Learning selection policies for navigation in unknown environments. In *IEEE Intl. Conf. on Robotics and Automation*. IEEE, 2009.

[119] Sajjad Taghvaei, Yasuhisa Hirata, and Kazuhiro Kosuge. Control of a passive walker using a depth sensor for user state estimation. *2011 IEEE International Conference on Robotics and Biomimetics*, pages 1639–1645, 2011.

[120] Sebastian Thrun. Finding landmarks for mobile robot navigation. In *IEEE Intl. Conf. on Robotics and Automation*, volume 2. IEEE, 1998.

[121] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics.* Intelligent robotics and autonomous agents. MIT Press, 2005.

[122] Michael P Vitus and Claire J Tomlin. Sensor placement for improved robotic navigation. *Robotics: Science and Systems VI*, page 217, 2011.

[123] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158, Oct. 2000.

[124] Yina Wang and Shuoyu Wang. A new directional-intent recognition method for walking training using an omnidirectional robot. *J. Intell. Robotics Syst.*, 87(2):231–246, August 2017.

[125] M.Z. Win and R.A. Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. *Communications, IEEE Transactions on*, 48:679 – 689, 05 2000.

[126] Peter Wurman, Raffaello D'Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29:9–20, 03 2008.

[127] G. A. Zachiotis, G. Andrikopoulos, R. Gornez, K. Nakamura, and G. Nikolakopoulos. A survey on the application trends of home service robotics. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1999–2006, Dec 2018.

[128] Junyi Zhou and Jing Shi. RFID localization algorithms and applications—a review. *Journal of Intelligent Manufacturing*, 20(6):695, Aug 2008.

[129] Ö. Çetin, H. Nazh, R. Gürcan, H. Öztürk, H. Güneren, Y. Yelkovan, M. Çayir, H. Çelebi, and H. P. Partal. An experimental study of high precision toa based uwb positioning systems. In *2012 IEEE International Conference on Ultra-Wideband*, pages 357–361, Sep. 2012.

[130] Višeslav Čelan, Ivo Stancic, and Josip Music. Ultra wideband assisted localization of semi-autonomous floor scrubber. *Journal of Communications Software and Systems*, 13:109, 06 2017.