UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Mobile Robot Navigation in Outdoor Environments: A Topological Approach

Alberto Manuel Martinho Vale

(Licenciado)

Dissertação para a obtenção do Grau de Doutor em Engenharia Electrotécnica e de Computadores

Orientadora: Doutora Maria Isabel Lobato de Faria Ribeiro

Júri

Presidente:	Reitor da Universidade Técnica de Lisboa
Vogais:	Doutor José Neira Parra
	Doutor Jorge Manuel Miranda Dias
	Doutora Maria Isabel Lobato de Faria Ribeiro
	Doutor Mário Alexandre Teles Figueiredo
	Doutor Pedro Manuel Urbano de Almeida Lima

Lisboa, Junho de 2005

UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Mobile Robot Navigation in Outdoor Environments: A Topological Approach

Alberto Manuel Martinho Vale
(Licenciado)

Dissertação para a obtenção do Grau de Doutor em Engenharia Electrotécnica e de Computadores

Orientadora: Doutora Maria Isabel Lobato de Faria Ribeiro

Júri

Presidente:	Reitor da Universidade Técnica de Lisboa
Vogais:	Doutor José Neira Parra
	Doutor Jorge Manuel Miranda Dias
	Doutora Maria Isabel Lobato de Faria Ribeiro
	Doutor Mário Alexandre Teles Figueiredo
	Doutor Pedro Manuel Urbano de Almeida Lima

Lisboa, Junho de 2005

aos meus pais

Agradecimentos

O doutoramento foi uma das decisões mais importantes na minha vida: uma aposta pela investigação e desenvolvimento científico, que envolve numerosos apoios e contribuições de muitas pessoas, a quem devo os meus agradecimentos.

Começo pela minha orientadora Professora Maria Isabel Ribeiro, a quem devo o especial incentivo à investigação e toda a sua disponibilidade e interesse pelo esclarecimento das mais pertinentes questões e discussões científicas salutares que promoveram um ambiente de desenvolvimento promissor no LRM (Laboratório de Robótica Móvel) no pólo de Lisboa do ISR (Instituto de Sistemas e Robótica).

Também no LRM, quero agradecer ao Professor João Sequeira as inúmeras conversas e debates aos mais variados níveis de controlo e robótica. Ao João Gomes Mota e José Castro, durante os meus primeiros anos no LRM e pelo quanto aprendi na administração dos servidores.

Ao IST (Instituto Superior Técnico) e ao ISR (Instituto de Sistemas e Robótica), agradeço todo o apoio material para a realização do trabalho de investigação. Este apoio é certamente crucial para o desenvolvimento de programas de doutoramento. E como tal, não poderia deixar de manifestar o meu apreço ao Professor João Sentieiro pela criação das condições e de um ambiente excelente de investigação no ISR.

A excelente equipa de trabalho envolvida no Projecto Rescue, no qual tive o privilégio de trabalhar, orientada pelo Professor Pedro Lima, a quem quero também agradecer a disponibilidade para diversas questões, ao Professor José Santos Victor, Alexandre Bernardino e José Gaspar pelas longas discussões sobre a Visão, João Frazão e Thomas Krause pelo trabalho em torno do ATRV-Jr e do Blimp. O projecto Rescue – Cooperative Navigation for Rescue Robots (POCTI/1999/SRI/33293) foi financiado pela Fundação para a Ciência e Tecnologia no âmbito do programa POCTI (fundos FEDER e do Estado Português) do QCAIII.

Agradeço ao Rodrigo Ventura pela disponibilidade em torno do Linux e pelas inúmeras discussões Linux vs. Microsoft! Ao Francisco Teixeira pelo excelente trabalho de equipa em torno do SLAM, que começou no *Summer School* de 2004 em Estocolmo.

Pelos diversos amigos no LRM desde o meu trabalho final de curso em colaboração com os meus colegas Nuno Antunes e Paulo Inácio, e no meu doutoramento, José Miguel Lucas, Francisco Melo, Zlatan e em especial com o professor Ognyan Manolov, durante a sua estadia no ISR.

Muito obrigado a todos aqueles que leram atenciosamente cada um dos capítulos da tese, em especial à Ana Relvas, Francisco Melo, Thomas Krause, José Miguel Lucas e Luis Díaz Dominguez, o que resultou em melhorias significativas.

Quero agradecer à FCT (Fundação para a Ciência e Tecnologia) pela bolsa de doutoramento que me foi atribuída.

Não podia terminar sem expressar os meus profundos agradecimentos aos meus pais pelo permanente apoio ao longo de toda esta vida. À minha Tia Madrinha, Avó, Tios e primos toda a minha amizade.

Alberto Vale

Acknowledgements

The Ph.D. was one of the most challenging decisions of my life: gambling on scientific research and development with lots of help and therefore it is almost impracticable to thank everyone. In many cases I do not even have the words to express all my gratitude.

First of all, I would like to thank Professor Maria Isabel Ribeiro, my supervisor, for her suggestions and constant support during this period of research at LRM (Mobile Robotics Lab) in the Lisbon pole of ISR (Institute for Systems and Robotics). We first met at the end of my graduation, which motivated myself for mobile robotics. Also at LRM, I would like to thank Professor João Sequeira, for the several suggestions about robotics and control, to Engineers João Gomes Mota and José Castro, during the first years at the lab, where I also learned a lot about network administration.

To Professor João Sentieiro I want to thank for creating such an excellent research environment. To IST (Instituto Superior Técnico) and to the ISR, I want to thank the resources given to pursue my research. My acknowledges for the Ph.D. grant from FCT (Portuguese Foundation for Science and Technology).

I enjoyed to work with the excellent research team of the Rescue Project oriented by Professor Pedro Lima, whom I would like to thank, also to Professors José Santos Victor, Alexandre Bernardino and José Gaspar for the suggestions about Vision, João Frazão and Thomas Krause about ATRV-Jr and Blimp. The Rescue Project – Cooperative Navigation for Rescue Robots (POCTI/1999/SRI/33293) is funding from the FCT, in the frame of POCTI (funds from FEDER and Portuguese Government) of QCAIII.

I would like to thank Rodrigo Ventura for the several tricks about Linux and for the discussions of Linux vs. Microsoft! Also to Francisco Teixeira for the excellent work about SLAM, started during the Summer School at Stockholm. I also met a lot of friends at ISR since my graduation, in special LRM, to José Miguel Lucas, Francisco Melo, Zlatan and Professor Ognyan Manolov. I must thank you all also the careful reading of the chapters of the thesis, in special to Ana Relvas, Francisco Melo, Thomas Krause, José Miguel Lucas and Luis Díaz Dominguez. Many improvements resulted from your suggestions.

Finally, I wish to express my deepest gratitude to my parents for the constant source of support all over these years.

Alberto Vale

Resumo

A tese tem como principal objectivo o desafio da navegação robótica em ambientes externos não estruturados, para o que são propostas metodologias baseadas em abordagens probabilísticas que focam os três grandes problemas: o tipo de representação, a localização e a navegação. A metodologia escolhida assenta numa base matemática sólida que procura resolver os três problemas simultaneamente de uma forma topológica, ou seja, a um nível elevado de abstracção.

A motivação da tese é inspirada na destreza de certas espécies animais com excelentes capacidades de navegação que permitem recolher o essencial à vida e manter a espécie.

Para este desafio foram desenvolvidos algoritmos, testados em simulação e em ambientes reais, com um robot móvel. As maiores contribuições científicas da tese baseiam-se na representação do cenário (uma nova abordagem de representação topológica, um conjunto de estados definidos por Gaussianas e ligados por orientações), na construção do mapa topológico (uma versão dinâmica do algoritmo de Estimação e Maximização), uma abordagem probabilística para a localização e navegação (versões optimizadas do algoritmo Progressivo-Regressivo) e finalmente, a extracção e selecção de propriedades (metodologias de extracção de diferentes tipos de propriedades do ambiente e a respectiva selecção de acordo com o ambiente a ser representado).

A tese termina com um capítulo de resultados experimentais em cenários reais. As experiências foram realizadas com um robot móvel equipado com diferentes tipos de sensores, comprovando assim a aplicabilidade da abordagem topológica, com um alto nível de abstracção.

As principais contribuições da tese consistem na definição e demonstração de como é possível implementar uma abordagem de alto nível de abstracção na navegação de robots móveis em ambiente não estruturados.

A tese tem como pano de fundo cenários de busca e salvamento, em especial o Projecto "The Rescue Project - Cooperative Navigation for Rescue Robots", onde o objectivo principal é desenvolver soluções para a criação de equipas de robots cooperativos que operem em ambientes externos não estruturados.

Palavras Chave

Navegação, Robots Móveis, Mapa Topológico, Robótica Probabilística, Extracção e Selecção de *Features*, Busca e Salvamento, Ambientes Externos

Abstract

The thesis addresses the problem of mobile robot navigation in outdoor environments and proposes methodologies based on a topological approach, concerning to three main issues: environment representation, localization and navigation. The selected approach, based on a mathematical support, has to solve the three main issues simultaneously.

The motivation of the thesis is based on some cutting edges of the nature, where are millions of species with fantastic navigation capabilities, that retrieve the essential for life.

For this purpose, complete algorithms were developed and tested in realistic scenarios with a real mobile robot. The main contributions of the thesis are the environment representation (a new topological representation, a set of notes defined by sum of Gaussians, connected by orientation), map building (a dynamic version of expectation and maximization algorithm), a probabilistic approach for localization and navigation (an optimized version of Forward-Backward algorithm) and feature extraction and selection (different types of feature extraction procedures with a selection criteria).

The thesis concludes in a chapter describing the experimental results acquired by a real mobile robot, showing that the developed algorithms achieve the main goals proposed by a topological approach and a high level of abstraction.

The main contribution provided in the thesis is the definition and demonstration of the applicability of mobile robot navigation in unstructured environments based on a high level of abstraction.

This work is concerned on a search and rescue like project, "The Rescue Project - Cooperative Navigation for Rescue Robots", where the main goal is to provide integrated solutions for the design of cooperative robots teams operating in outdoor environments.

Keywords

Navigation, Mobile Robots, Topological Map, Probabilistic Robotics, Feature Extraction and Selection, Search and Rescue, Outdoor Environments

Contents

List of Symbols

s_i	state i of the map
v_j	j^{th} feature or attribute, $j = 1, \ldots, M$
$s_i(v_j)$	value of the attribute v_j at state s_i
H_i	entropy of state s_i
S	set of N states of the map
$L(S)$	map's likelihood
$F(S)$	expectation of the map's likelihood
t	time instant
r_t	rawdata or unprocessed data acquired by the sensors at time instant t, a vector of dimension D
o_t	observed vector of features extracted from r_t
f	feature extraction function, $o_t = f(r_t) \equiv f_t$
$f_t^{v_j}$	extracts the feature v_j, $o_t(v_j) = f_t^{v_j}$
O_t	observations sequence up to the time instant t
q_t	robot location at time instant t
$\hat{q}_t$	robot location estimation at time instant t
Q_t	states sequence up to the time instant t
a_{ij}	state transition probability
$b_i(o_t)$	observation probability
B	observation probability in all possible states, $B = [b_1(o_t)\ b_2(o_t)\ \ldots\ b_N(o_t)]^T$
π	initial localization probability in all possible states, $\pi = [\pi_1\ \pi_2\ \ldots\ \pi_N]$
π_i	initial state distribution

w_{ij}	probability that the observation o_i belongs to the state s_j
δ_j	oscillation of w_{ij} for a state s_j
δ	oscillation of w_{ij} for all states
Δ_δ	maximum level of oscillation to consider when the mapping algorithm has converged
$\alpha_t(i)$	parameter relative to all the observations from the past up to the time instant t
$\beta_t(i)$	future observations until time instant t
V	translation velocity $[m \cdot s^{-1}]$
Ω	angular velocity $[rad \cdot s^{-1}]$
Ω_R, Ω_L	angular velocities of the right and left wheels $[rad \cdot s^{-1}]$
f_a	attractive behavior
f_r	repulsive behavior
f_w	weighted combination of attractive and repulsive behaviors
θ	robot's orientation
θ_d	orientation goal to reach a given state
θ_e	error between θ and θ_d
θ_o	orientation to the nearest obstacle
θ_{ij}	direction between state i and state j

Chapter 1

Introduction

The thesis addresses the problem of mobile robot navigation in unstructured outdoor scenarios and proposes methodologies based on topological navigation to accomplish its implementation. The topological navigation is a challenging approach given the high level of abstraction used to represent the environment, which supports the robot localization and navigation in a framework that also integrates the inherent uncertainty.

The motivation of the thesis starts on some cutting edges of the nature. There are millions of species with fantastic orientation capabilities, that retrieve the essential for life. In special, the ants, the honeybees and the migratory birds.

Ants have been living on the Earth for more than 100 million years and can be found almost anywhere on the planet [1, 54]. It is estimated that there are about 20,000 different species of ants as the one represented in Figure 1.1. For this reason, ants have been called Earth's most successful species. Western Harvester ants make a small mound above the ground, but then tunnel up to 30 cm straight down to hibernate during winter. Ant mounds consist of many chambers connected by tunnels. Different chambers are used for nurseries, food storage, and resting places for the worker ants. These complex environments are hundreds or even thousands times larger than the size of a simple ant, but the most enigmatic is that ants memorize and understand the maze. By looking carefully to ants during a short period of time it is possible to perceive how they navigate along the way and communicate with each other in an effective way. Ants communicate by touching each other with their antennae. Ants also use chemicals called pheromones to leave scent trails for other ants to follow.

Some scientists proved the efficiency of the ants navigation by blocking a "platoon" of ants with a rock. The colony had planned a path to reach their goal (Figure 1.2-a)), when an obstacle was placed, blocking the way (Figure 1.2-b)). To successfully reach the other side, they tried different possibilities, some of them with success. This is called a re-learning phase (Figure 1.2-c)). The succeeded ants reached the goal learning the right trails by pheromones, demonstrating a cooperative behavior. After a couple of seconds, all the ants were following the promising way, as illustrated in Figure 1.2-d).

Figure 1.1: Ants, one of the Earth's most successful species

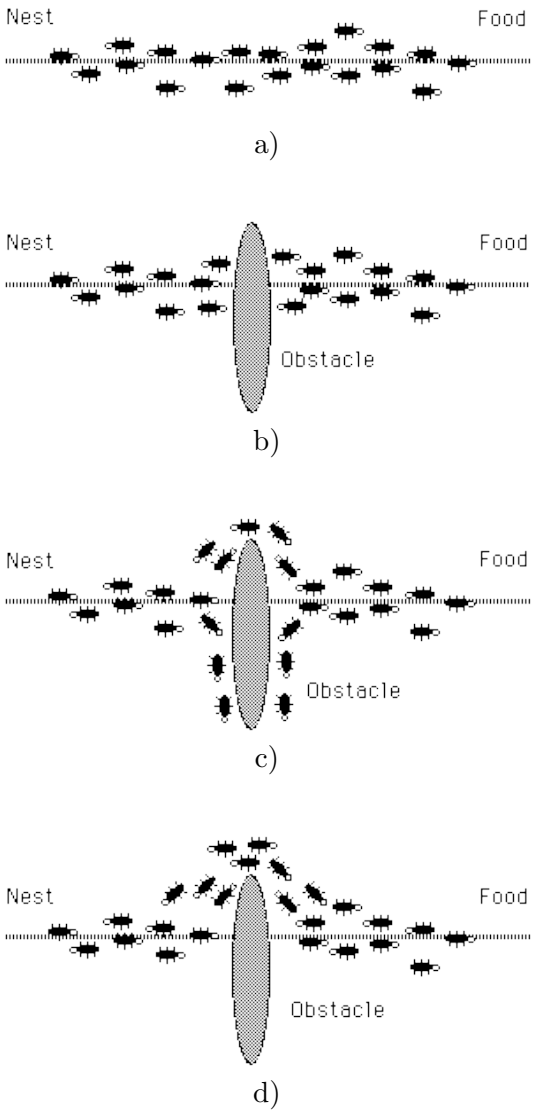


Figure 1.2: An illustration of the efficient navigation of ants: a) initial trail to the food, b) an obstacle was inserted to block the way, c) a learning phase d) the new (best) path adopted

The last sentence points out one important issue in robotics, the navigation aspect, in particular, the consideration of the best path to reach a goal. The ants' capability is one of the motivations in the thesis, which addresses a navigation approach and obstacle avoidance behaviors prepared to dynamic environments, underlying the orientation as an important feature for the topological representation.

However, the ants' navigation is not the only interesting issue of animals' navigation. There are several interesting questions of animal biology around social insects. The well-known communication scheme between honeybees, as represented in Figure 1.3, has been studied in particular by Dr. Karl von Frisch, honored for his research with the Nobel prize, and his scholar Dr. Martin Landauer in [72]. They showed that during their famous dancing, honeybees are able to communicate at least three types of information to their sisters. Two of them are concerned with the navigation to a far nectar-source and the third is about the quality of the nectar. Recent research developments made it possible for the first time to communicate this information directly to the bees by simulating the dance using small and simple robots. This technical support allowed a real breakthrough in a science domain that was reserved for exact observation and description of this animal capabilities, to implement in real and useful robots.

Figure 1.3: The communication of social insects - honeybees dancing

It remains to understand the scheme/mechanism under which bees get and transform navigation information. The possible keys lie on the scheme with which bees collect visual landmarks and/or record information using their magnetic sense organ. The navigation abilities of bees retrieve the essential information to find their way home, by memorizing the angle of flight to the position of the Sun, and even by computing the Sun's movement in the sky, as depicted in Figure 1.4. Bees know-how to measure the flight-distance, involving the wind's force.

A guard bee inspects an incoming bee to see if she belongs to the family or not. Going into the wrong house may lead to rejection and occasionally, to death. Consequently, the

navigation algorithm used by bees has to be efficient, otherwise, it is rewarded with death.

Inspired on honeybees, a robot equipped with orientation sensors [3] records the most important directions between useful places. This issue is covered by the thesis, as the direction between places is recorded similarly to the bees' navigation.

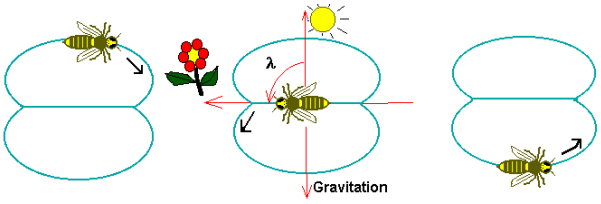


Figure 1.4: The navigation ability of honeybees: find the way home and to flowers, computing the position of the Sun and measuring the flight distance

Other species of animals also have special capabilities in different habitats, like the geese shown in Figure 1.5 and explained in [6]. In [78], it is described how some species of birds (ducks, geese, as well as migratory rails, doves and woodcock) with their feathered wings and tails, bones, lungs, air sacs and their metabolic abilities contribute to the amazing faculty of migratory journeys. The authors remark that even humans with their many ways of locomotion do not equal some birds in mobility. No human population moves each year as far as from the Arctic to the Antarctic with subsequent return.

Figure 1.5: Snow Geese, one of the most well known migratory birds, in flight

The long migratory journeys are followed by large flocks, large amounts of birds that know their current location, their destination, and the direction to travel to get from the current location to the destination, which could be represented by a vector. Most of the researchers describe this capability as a vector navigation. This vector, which is passed and updated along generations, is well defined given certain orientation cues (landmarks on the Earth's surface, the magnetic lines of flux that longitudinally encircle the Earth, both the Sun and the stars in the celestial sphere arching over the Earth, and perhaps prevailing wind direction and odors), resulting in patterns of migration (the routes of migrations). These migratory species are also able to deal with storms or other aerial obstacles or even with their own exhaustion, maintaining their orientation goal regardless of these hazardous events.

These species of birds develop a cooperative navigation mechanism, not only when they help each other on their exhaustion, but mainly when they pass and update their knowledge along generations.

Most of the target applications of topological navigation, as the project where the thesis is included, compounds cooperative navigation between robots. But more than the capability of developing a cooperative behavior, the migratory birds have a way to represent its trajectory and to anchor it to the world. The virtuous navigation of this species of birds raises two important issues that the robotic application in the thesis also shares: the type of map or representation of the environment (where the robot operates) and the way to acquire or to build that representation. Similarly to ants and honeybees, the migratory birds also record relevant features on the Earth and memorize them as landmarks to define a map. They use that map to optimize the trajectories, that are much longer when compared to the size of these animals. The thesis also develops algorithms to build a map of a large natural environment, based on clustering the available features, which results in a set of places defined with different profiles, i.e., different sets of characteristics. Similarly to the birds, if an environment representation is available, a mobile robot localizes itself on that map, and follows the best sequence of places to reach a target.

At this point it is important to stress the capability of different animal species to travel large distances, when compared to their dimensions. Motivated by this amazing capability, robotic researchers have looked to the different types of navigation of the species, including fishes, mammals (e.g., dolphins and whales, that travel between continents) and tried to produce similar robotic systems to work in real environments. Hopefully, this type of research gives some insight into why the animals do it this way and how they solve the three main question of robotics, addressed in the thesis: localization, navigation and mapping.

Furthermore, a robot must have a way to perceive the world like these animals. Otherwise, it could not get a feedback for the action taken and, in particular, will not be able to recognize and record the features previously referred. The world perception in robotics is provided by electronic sensors, that act similarly to some animals' organs. A question arising at this stage is: how is the mobile robot navigation dependent of the world perception? For instance, ants have compound eyes. How does this affect the visual homing task? Besides using local landmarks, some insects also use the position of the Sun (in fact, its polarization pattern) as a global landmark. The Sun does not move much during a standard trip around the nest for an ant. Obviously, using the Sun would not be as successful as using local landmarks for journeys of over a short period of time.

Animals also use dead-reckoning, just like the common odometry on robots, which could be useful in navigation for short periods of time. The odometry on robots is subject to drift errors. To deal with these errors and to minimize their effects, the robots use landmarks to back up the information.

Minuscule animals, in particular insects, even though being very small and with a

"peanut level of intelligence" can navigate with success in places thousands of times larger than their dimensions. Their mechanisms are sufficient to retrieve the essential for life. Given the mystifying nature of these animals' navigation capabilities, most of the robotic research started from biological inspiration, creating robots and copying the sensorial organs to perceive the world in a similar way. The thesis is inspired in nature, implementing algorithmic methodologies to build a topological representation, based on the perception acquired by sensors and to navigate over that representation.

When the knowledge about nature is referred, it is common to project and compare the animal behaviors into a human world, i.e., if the animals navigation capabilities also work in scenarios created by humans. For instance, ants draw trajectories where they avoid obstacles, honeybees are able to fly from flowers to their nest and even migratory birds cleverly navigate along continents. But does any of these species navigate in buildings? As their algorithms are robust enough, they can also navigate in extreme conditions, namely when unexpected and hazardous events occur. This means that their world representation (map), which can not be complex given their limited intelligence, is plenty of information to accomplish life. The map is a set of important places with simple features extracted using their sensorial capabilities, connected by events or actions. The places, further named as states in the thesis, are described by simple features, as the pheromones for the ants, position of the Sun or magnetic fields for the honeybees and migratory birds. On humans, these places could be defined using other features, based on our enormous database.

In humans it is much more complex to understand how the brain processes the observed information to navigate through a scenario. Cognitive science researchers have suggested that humans use cognitive maps that link together landmarks that are recognized given his/her database. These landmarks are obvious (to avoid ambiguities), common and understandable for any human, as rooms, buildings, stations, parking-places, gardens, hills, rivers, lands, oceans. They are often connected by routes. These types of maps, known as topological maps, are complex enough to support traveling long distances according to the appropriate transportation mean (e.g., foot, car, boat, train, plane) and simple enough to avoid the incumbency of recording every information over the physical location covered by the map. These cognitive maps, as the one illustrated in Figure 1.6, express an abstraction level of world representation, which appears to be stored in the hippocampus (part of the human brain responsible for emotions, navigation, spatial orientation and consolidation of new memories, illustrated in Figure 1.7), as explained in [82].

One of the most ambitious goals of robotic research is to develop an approach for mobile and completely autonomous navigation of robots in outdoor environments, similar to the scenarios where animals survive. The approach implemented in the thesis is prepared for a diversity of scenarios, the large spectrum of information acquired by the available sensors and the unexpected events that occur during the operation. Furthermore, the implementation must be robust to scenario changes and must take the best profit when

Figure 1.6: Illustration of a possible topological representation of a map of Lisbon's subway, with each station identified by particular buildings or other features around the area (mainly monuments)

cooperative operation is available. The thesis addresses this challenging mobile robot navigation application supported on a probabilistic topological approach, based on the previously motivations and supported by mathematical models to solve the three main issues of environment representation, localization and navigation.

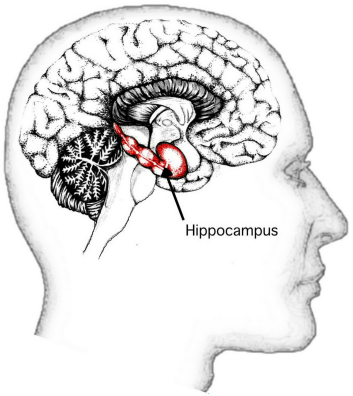


Figure 1.7: The hippocampus, part of the human brain responsible for emotions, navigation, spatial orientation and consolidation of new memories

1.1 Problem Definition

The main problems of mobile robot navigation in outdoor and unstructured environments consist on finding an appropriate type of representation and the methodologies to localize and navigate in the scenario based on that representation. Selected the best type of representation, it is necessary to find a methodology and a mathematical framework to initialize, build and update that representation over the time. It also requires a linkage between this type of representation characterized by a high level of abstraction and a motion control level.

As described in the first part of this chapter, small animals can move in complex environments and, consequently, perform their own lives, finding food, recognizing the environment and returning to their home. But the question still remains: how and what did they record with the sensorial capabilities available on their bodies? The answer bears the necessity to know the specific type of world representation used by these animals. The representation, at an unknown level of abstraction, can provide the essential information to perform navigation several times with success. Moreover, there are several unpredictable variables associated to the perception and to the representation that leads to uncertainty, but animals are able to overcome it.

The target applications addressed by the thesis (described in detail in Section 1.2), endow large and unstructured scenarios, which are similar to natural environments, where animals, in particular, ants, honeybees and migratory birds, live. Therefore the thesis has a biological inspiration, with the robot designed to observe, act and behave like an animal, or even a human, to face the common issues.

A mobile robot is developed and programmed to accomplish a mission, a problem on time and space: the robot has to move to the target(s) position(s) during a period of time (as fast and safe as possible). The first problem implies that the robot has to localize the target. It requires a world representation (commonly defined as a map), where the robot identifies the target and has to estimate its current position (the localization problem). The map could be incomplete or has to be updated, which addresses the map building problem. Finally, to accomplish the mission, the robot has to move to reach the target goal, selecting the best trajectory, which is the navigation problem.

In natural environments, large and unstructured scenarios, it is important that the robot is robust. Robustness is not limited to a hard case and powerful batteries. It also includes a way to perceive the world around and, based on a set of algorithms associated to the mechanical body, move the device and react to sensorial input, according to the perception of the world and its mission [85].

A robot designed to operate in outdoor environments is programmed to accomplish a mission, regardless of being equipped with legs, wings or wheels and appropriate sensors to perceive the world. Given the observations acquired by its sensors, it is important to know-how to represent this information to support the envisaged applications, as explained in Section 1.2. The first question lies on the choice of the best environment representation

that fits in the scenario of the target mission. After the selection of the representation type, it is necessary to build the environment map based on the acquired observations. The type of representation must be robust to deal, not only with the world changes, but also with the uncertainty included in the observations. Moreover, the map building algorithm must update the current world representation.

The navigation is accomplished based on the available map. There are different types of maps, as illustrated in Figure 1.8. The navigation and the map require the same language, which means that a metric navigation requires metric information and a topological navigation requires topological information available on the map. A metric navigation or a topological navigation can be accomplished in the same scenario, for the same mission, by using different levels of abstraction. For instance, when a robot moves from a room A to a room B, it may have different levels of navigation. The first and more abstract is "move from room A to room B". The next one, with a lower level of abstraction, is moving through the doors and corridors, followed by a metric navigation. The lowest level of navigation, the metric navigation, includes the path following that requires a metric referential, as illustrated in Figure 1.8. The same level also includes the obstacle avoidance and a motion control that provides commands to the actuators of the robot. It is important to underline that the different levels of navigation have connections between them. Otherwise, it would be impossible to control the mobile robot using only the topological level without the motion control.

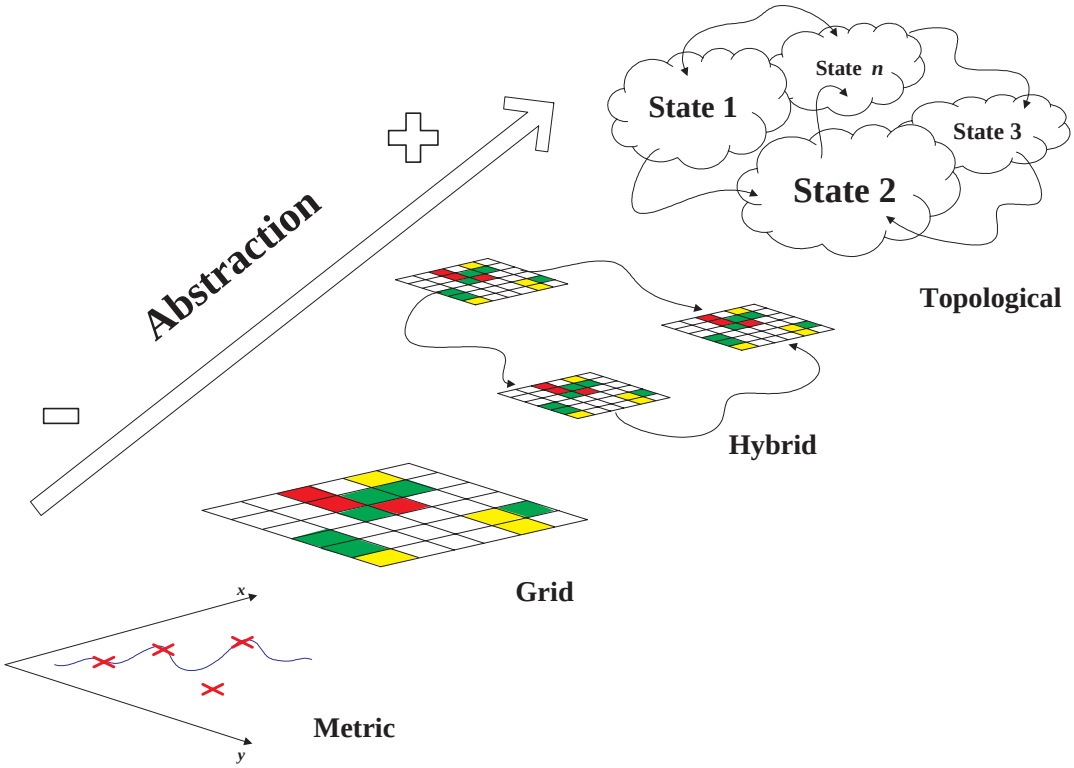


Figure 1.8: Environment representations at different levels of abstraction

The thesis focus on the navigation of a mobile robot with a high level of abstraction, defined as topological navigation. It is at exactly this level of abstraction that becomes such possible to define places as "rooms" or "corridors", or dividing the world in sub-regions, which are not necessarily closed or connected spaces, as a common map of a city (see Figure 1.6). For instance, a room or "corridor", "garage", "garden", "station", "school", are places characterized by a set of particular "properties" or "features". The sensors installed on the robot perceive the world by retrieving rawdata, which must be processed to extract the features that represent the world. Therefore, the world is similar to a panoply of mixed features and the places or "states" are well defined clusters of some features to avoid ambiguities. The amount of information available at these scenarios leads to a necessity of feature selection, to accomplish better world representation. In this paragraph, we mention some important and complex issues addressed in the thesis: feature extraction from the sensors' observations (rawdata), selection of the best ones and cluster and building a topological map.

The topological map is a set of clusters, places or, as defined in the thesis, a set of states. So two other questions turned out: how to represent states, given that the states are identified with features and how to represent the connection between states? The first question requires a map building algorithm as well as a mathematical support to develop the topological navigation approach and, consequently, to represent the states. The second question requires physical connections (metric information, motion commands) between states, linking to a low level of abstraction: a metric navigation, i.e., a navigation based on sending motion commands to the actuators. The mathematical support for navigation has to deal with the uncertainty included in the world, the perception of the world (observations and/or consequently on features) and also on the motion. This justifies a probabilistic approach, a fundamental key in the thesis.

After this perception and representation of the world at a high level of abstraction, it endows to the problem of navigation at the same level, trying to accomplish the mission, following the best path to reach the goal. New questions arise: how to localize over the map given the acquired observation (transformed into features) and how to navigate over the world given the map and the current location? These are the topological localization and navigation issues that the thesis addresses.

Resuming, the main questions addressed by the thesis are the mapping, the localization and the navigation. These questions are addressed by a topological approach, where the development is supported by mathematical algorithms to deal with the uncertainties always present on environment representation, perception and motion.

1.1.1 State of the Art

The mobile robot navigation is a broad topic, covering a large spectrum of different technologies and applications, which address the mapping, the localization and the navigation.

The first question, the mapping, aims at a scenario representation. There are several

approaches, some of them based on very ancient techniques [32, 37] or even on animal instinct, as mentioned in Section 1.

The first mobile robots, with small number of degrees of freedom and autonomy, were based on programmed machines engaged with human-machine interface for tele-operation as cited in [73, 116]. Consequently, the navigation was simply driving the machine to a target position, with a poor or even none world representation. The observations retrieved by on board sensors aimed only the assistance to human operator to perceive and recognize the scenario and building an abstract map in his/her mind. Consequently, the human operator drives the machine to the target goal, believing on his/her instincts and on the poor sensors retrieval.

Later, several researchers started to develop high levels of navigation autonomy, where the robot could self-localize. The pioneer works [20, 60, 70] discuss the localization problem, assuming that the map is known. The map could be defined as a set of beacons where the robot localizes using simple triangulation [15] or trilateration (very common on Global Positioning System - GPS [22], or Differential-GPS, DGPS [91]), or both techniques [20]. The installation of beacons require human intervention, which becomes a drawback or even impossible in certain missions. Consequently, the maps must result from natural landmarks available in the environment.

For map building in more complex world representations, it is necessary special types of sensors installed on the vehicle. Most common mobile robot are equipped with range sensors [87, 90] (ultrasound sensors and/or laser range finders) and intensity sensors [46, 57, 66, 121, 128]. Given the necessities and the growing up of the technology and the large spectrum of the sensors available, the environment representation became a challenging area of research. The most common approaches are divided in three groups: geometric (necessarily with metric information), topological (adjacency-graph based representation) and hybrid (sharing metric and topological information) .

The geometric representations commonly based on Hidden Markov Models (HMM) and Kalman Filtering (KF) [63, 92, 111] are the most popular approaches to uncertainty handling. The occupancy grids are a particular case of metric maps, deeply described in [35], mainly for indoor environments. Most of the topological research to recognize places and to record them as references is bundled on information retrieved by vision sensors, in particular edges [44, 62], the main components of the image (particular objects like trees, cars, people) [115], by colors [43] and by panoramic images [109]. There are several approaches that use vision and motion commands in a qualitative way [42] for mapping and navigation, based on bio-inspired techniques (as some species, e.g., ants and bees referred in the Section 1). The resulting maps are often improved using other types of information, like a set of geomagnetic signatures [10], when vision is not available or is poor (specially in underwater environments [46]). Given the uncertainty associated to the environments, as justified by [107], it is important that the topological maps also include a probabilistic support. Geometric maps and topological maps can be combined as hybrid

maps, described in [16, 29, 103], where most often the topological representation arises from the metric one [36, 106, 112] (from grid maps [120]), or from features of motion (e.g., velocities, accelerations) that can be obtained from odometry [19].

In spite of being distinct problems, the localization and map building must be handled as a combined issue. Moreover, the environment is not static [76] and since the localization returns uncertain estimations, it could be a clue to update the representation by the map building algorithm according to the feature dynamics [28].

One active and relevant research area is the Simultaneous Localization and Mapping - SLAM [23, 31, 41, 80, 95, 114], also known as Concurrent Map Localization - CML [12, 38, 113]. Most of the SLAM approaches are oriented to indoors, well structured and static environments (like domestic ones [131]) and give only metric information regarding the position of the mobile robot and of the landmarks. Furthermore, some applications on dynamic scenarios are emerging [12] also for outdoor environments [48, 80]. There is some research on cooperative or collaborative SLAM - CLAM [38], or probabilistic CLAM [100], which covers some issues on topological approach [30]. The SLAM is a strong approach to implement in particular areas, but extremely dependent of matching between beacons or features. It also requires a high computational load, since the algorithm propagates a matrix that models the uncertainty and correlation between the robot and the landmarks. Since the SLAM usually requires a large number of landmarks, the size of the covariance matrix is extremely large (the size is equal to the square of the number of landmarks plus the pose), and therefore it is difficult to implement it in outdoor environments, since it is necessary to use this matrix several times, namely its inverse. Even with an optimized version of the algorithm, defined as FastSLAM [80, 86], it leads to a large computational burden.

The SLAM approaches are fundamentally oriented to metric navigation, with the map defined by the position of several landmarks on a referential. Consequently, the associated robot's localization also retrieves metric information. It is important to extend the SLAM for a more abstract level of navigation, where the three main issues, localization, navigation and mapping, remain simultaneously addressed and supported on a topological representation. Even for the topological approach, with a high level of abstraction, it is important to establish a connection with the low level of navigation, providing motion controls for the robot. Moreover, a topological approach could be implemented, concerning different layers of abstraction, where the lowest level corresponds to the motion control.

1.1.2 The Thesis Approach

The thesis addresses the problem of mobile robot navigation in outdoor environments and proposes methodologies based on a topological approach, including the three main issues of environment representation, localization and navigation. The selected approach, based on a statistical framework, solves the three main problems in an integrated scheme. The main open and relevant questions regarding a solution to the described navigation problem

are:

- Which type of world representation should be adopted;
- How to initialize, build and update this representation over the time;
- Using this same representation, how to localize and navigate a mobile robot to accomplish a mission in a given scenario;
- Based on a type of representation with a high level of abstraction, how to link it with a motion control level;
- May the same representation be shared by more than one robot and how to perform this task.

There are several ways to represent the scenario where the robot operates, as referred in Section 1.1.1. Most of them rely on metric information, commonly used in SLAM approaches. Since outdoor environments include large scenarios, with enormous physical areas, the amount of information for representing the environment increases immeasurably. To accomplish the mobile robot navigation in real time with limited number of sensors, actuators and computational power to process the information, the environment representation must be discretized in locations or places, defined as states. This is defined as a topological map, which provides an abstracted description of the environment. This representation does not require metric information and is based on physical (natural or artificial) features that characterize the locations or places. The structure of a topological map relies on a set of nodes that represent the states, characterized by a set of relevant features modeled by mathematical functions. In the thesis we used a sum of n-dimensional Gaussian probability density functions to represent each state of the topological map, where the dimension n corresponds to the number of different features. Given the high level of abstraction, each state defines a place not characterized by its position, but with a feature's profile observed by the robot when localized in that place. A topological map does not provide a high accuracy representation to support the robot's motion. Instead, a metric approach can locally, in each state, control the robot. The selected approach also addresses a modular architecture, where topological representation is combined hierarchically with a metric representation for motion control purposes.

To build and update a topological map, i.e., to find the parameters that define the sum of n-dimensional Gaussian pdfs, a Dynamic Expectation and Maximization algorithm is used. This represents a revised version of the Expectation and Maximization (EM) algorithm [93, 111, 113], to estimate the number of states and the parameters that define the sum of n-dimensional Gaussian pdfs. Given the natural conditions of outdoor environments and the robot dynamics, all the past information is condensed just into the previous estimation, which represents a Markov assumption. Consequently, Markov Models are the framework that supports the study of the state evolution along time.

The connection between the states is given by the orientation angles between them. For instance, to cross from state A to state B the robot has to follow North-East direction. The angles may assume discretized values (like a compass with the 8 main directions, north, south, east, ...) or numerical values (commonly between -180º and 180º). To identify the direction between states, a learning method (detailed in Chapter 4) is implemented, similar to the transition probabilities estimation of Markov Models. For instance, if it is possible to estimate the transition probability between state A and state B, it is also possible to estimate the directions where this transition occurs.

After retrieving the map, the states and their connections, it is necessary to estimate the robot's position in that map. Since the map follows a topological representation, the localization procedure is also developed using a topological approach. Similarly to the metric maps used in SLAM, the topological navigation has to consider the uncertainty, always present on environment representation, perception and motion. Therefore, the localization develops a probabilistic and topological approach. The robot estimated location is the map's state that is most likely to have produced the same features extracted from the sensors' rawdata, acquired during a given time interval. To estimate the robot's state, an optimized version of the Forward-Backward algorithm is used [98].

The navigation is developed based on the topological map, knowing the robot's current position and the target. Solving the navigation problem in this topological framework corresponds to finding the best path to reach a goal in the map. Consequently, the paths must be expressed in a language containing states, since the current location is a state and the map is a set of states. The path is a sequence of well identified states, computed by a revisited Forward-Backward algorithm, from the current state to the goal. Moreover, the navigation has to include motion controls, for the robot to move along the sequence of states, which corresponds to a lower level of control. The motion control is based on a sum of behaviors, an attractive behavior to the state goal and a repulsive behavior to avoid obstacles or not desired directions [14, 21, 34, 84, 105].

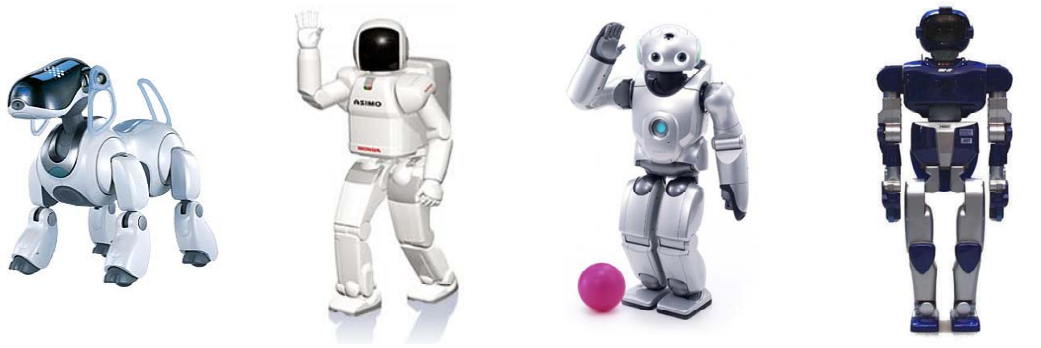
The construction of the topological map, defined by a set of states (each represented by a sum of n-dimensional Gaussian pdfs that model the features), requires a feature extraction procedure. The features are extracted from the sensors' rawdata. Given the sensors available on the robot, the essential features are edges, color-histograms, image segmentation, image components, range-data, GPS and orientation. Since it is possible to obtain large amounts of features extracted from the available sensors, most of the features are redundant in certain scenarios, it is important to implement a feature selection criteria. Thus, we used a correlation matrix to remove the redundant features.

The rawdata used for the map building is not necessarily obtained from the sensors in a single robot. Underlining the importance of cooperative navigation, it is possible to extract features from the observations of heterogeneous robots in a team (for instance, terrestrial, aerial, legged, wheeled, etc). In a cooperative topological implementation, the map can be processed by multiple robots, using the different points of view of the scenario

and perception by each robot and sharing the updated map by all the robots of the team.

1.2 Target Applications

Nowadays, several companies are developing and implementing models of animal-like and human-like robots (humanoids), as those illustrated in Figure 1.9. Their movement becomes similar to the human walking, climbing, picking something and consequently, they are equipped with reactive sensors, as touch, force (including inertial), vision and others. Given this growing proximity of humans and machines, it is quite important that they share the "same language" and are able to operate in the same scenarios.



a) b) c) d)

Figure 1.9: Recent robots available on the market: a) Aibo from Sony [8], b) Asimo from Honda [2], c) Qrio from Sony [9] and d) Hrp-2p from Kawada Industries [4]

These machines are able not only to use a metric navigation, but also a high level of abstraction. For instance, the task "pick up this box and put it in the room". This task requires metric procedures to identify the distance to the box and how to pick up it but more important, requires the knowledge of what a box is and what and where the room is. Moreover, it requires the knowledge of where the robot is at the current time instant and how to reach "that" room.

Not only these commercial humanoids but also other types of robots, for example wheeled robots, developed for testbed, repetitive and/or dangerous tasks, or simply for entertainment, are identified as possible targets to the research described in the thesis. The scenarios where these robots operate cover important issues of mobile robot navigation, such as unstructured environment or even scenario changes. For that reason, the thesis mainly addresses outdoor navigation, that shares the same properties.

Outdoor scenarios represent high unstructured environments with large physical area, where several times humans can not interfere or even be present, leaving the robot with a high level of autonomy, as the Spirit [5] (see Figure 1.10) and Opportunity Mars rovers, developed by the NASA, that reached the planet in 2004. Given the physical limitations on communication and with GPS not available, it is strictly necessary a high level of abstraction to control the machine. The robot has to adapt, record the main features of

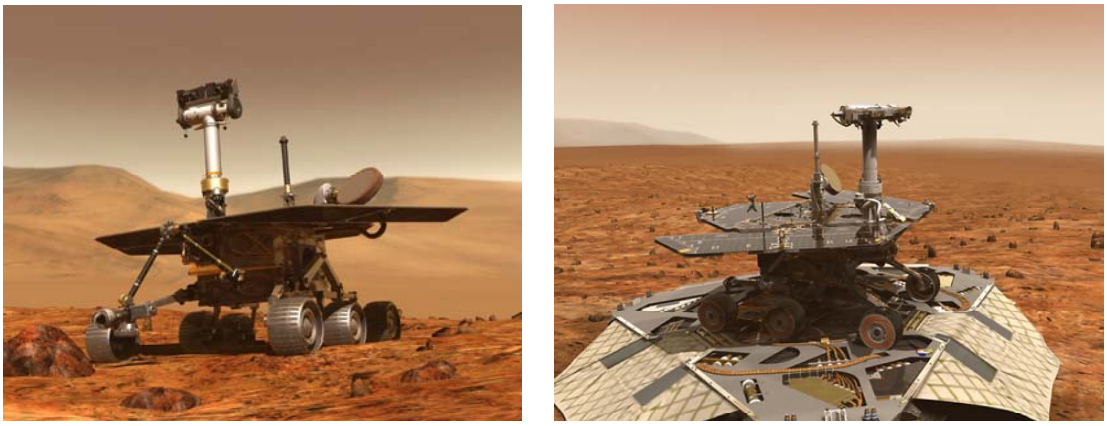


Figure 1.10: Spirit: the Mars rover, developed and implemented by NASA, that reached to the planet at January 2004

the environment and start to navigate to accomplish the mission.

1.2.1 Search and Rescue like applications

The challenging applications do not exist only in other planets, as in Mars. There are thousands of situations on Earth, where the robots intervention in outdoors scenarios could return important profits. The topological approach is developed concerning real situations, where the human intervention would not be possible. Most of the envisaged situations require emergency response, as search, rescue, extrication hazardous material, incident command, medical interventions, rapid sampling, plume tracking, hazard monitoring and information technology.

Disaster rescue is one of the most serious social issues involving large numbers of heterogeneous agents in hostile environments. Several researchers are working on this socially significant domain at various levels, involving multi-agent team work coordination, physical robotic agents for search and rescue, information infra-structures, personal digital assistants, simulators and decision support systems, evaluation benchmarks for rescue strategies and robotic systems.

The search and rescue area has been attracting an increasing number of researchers of the robotic community. Moreover, scientific events have been gathering these researchers, as the RoboCup-Rescue - a robotic approach to the disaster mitigation problem [108] and as noticed by [53] is permanently leading new robotic components. This challenge domain bears to the platform development, including sensors, control, communication, power, user interfaces, reach-back and visualization of data and to a more sophisticated and high-level of discussion - systems integration - human-robot integration, distributed robot/sensors teams, metrics and standards and insertion of technology into workspace.

Most of the researchers use simulations or artificial scenarios provided on these scientific events as a testbed for their prototypes. They are intended to be used in real situations of disaster, as those illustrated in Figure 1.11.

a) b) c) d)

Figure 1.11: Hazardous scenarios and challenging applications: a) Fire b) Demining c) Earthquake and d) Chernobyl disaster

Earthquakes are one of the most critical and emergent situations where the scenarios are completely unstructured, unpredictable and still unsafe, even in the presence of scenarios under the ruins. Earthquakes lead to fire, floods, similar situations that result from terrorism, like the one occurred on Federal Building in Oklahoma, May 1998. But the most catastrophic was the 9/11 incident in New York city at World Trade Center, where thousands of people died and the search and rescue operations were a rasping task to accomplish. People tried to get inside the ruins, but this was quite difficult, if not impossible. The mission was not only navigation, but also to discover possible victims and deliver assistance to them.

Shortly after the World Trade Center collapsed, a team of researchers at University of South Florida, supervised by Robin Murphy, tele-operated their robots and expertise to assist the search for bodies and survivors at ground zero [26, 88, 89]. She noticed that "robots can save lives and make the world a better place".

Unfortunately, wars have always book-marked the history and left awful scenarios. In spite of complex environments, it is often simple to navigate through. However, these scenarios commonly hide a big problem: the mines. Several mines, randomly distributed over the place, make the human intervention risky, time consuming and would require a lot of people to detect and remove them. This job can be accomplished by a team of robots to inspect the area to detect and demining, if possible.

Nevertheless, the hazardous scenarios are not necessarily caused by human intention. Catastrophes also occur due to unforeseen circumstances. One example of unpredictable catastrophes occurred in April 1986: one of the reactors at the Chernobyl Nuclear Power Station, 100 km north from Kiev, blew up during a routine daily operation. Nearly nine tons of radioactive material - 90 times as much as the Hiroshima bomb - were hurled into the sky. Winds over the following days, mostly blowing north and west, carried a fallout into large area. About 135 thousand of people were evacuated from a 30-km radius around the plant, with the peripheral areas remaining at a high risk of radioactive exposure. The reactor was enclosed in a concrete-and-steel sarcophagus. Over the following years about 600,000 people known as "the liquidators" worked on clean-up operations inside the 30-

km zone. This is another example of mission that could have been executed by a team of robots, where the human intervention is a risk.

1.2.2 The Rescue Project

The thesis was developed under the framework of the search and rescue like project, "The Rescue Project – Cooperative Navigation for Rescue Robots", whose main goal is to provide integrated solutions for the design of teams of cooperative robots operating in outdoor environments. It focus, with special detail, in the short and mid-terms on perception and world representation issues, as well as cooperative navigation, and, in the mid to long-terms, on task modeling, planning and coordination. The project developments are implemented in two platforms depicted in Figure 1.12. For a full project description see [77], with a brief summary in Appendix A.

Figure 1.12: The two platforms of the Rescue Project: on the left side the aerial blimp/zeppelin robot and, on the right side, the ATRV-Jr (a land indoor/outdoor robot)

1.3 Novelties and Major Contributions

The main contribution of the thesis is the definition and demonstration of the applicability of a high level of abstraction on mobile robot navigation in unstructured environments. For this purpose, complete algorithms were developed and tested in real scenarios with a mobile robot.

The main contributions of the thesis include:

- Environment representation - a topological map is more powerful to apply in outdoor environments (large and unstructured scenarios), when compared with geometric maps. A new topological representation is defined as a set of nodes or states modeled by sum of Gaussians, representing the main features that characterize the states. The connection between two states is given by an orientation value that indicates the direction that the robot should follow to move from one state to the other;

- Map building - a dynamic version of the Expectation and Maximization algorithm to build the world representation as a topological map. The importance of this dynamic version of EM consists on adjusting the number of states in order to compute the best environment representation. The map building algorithm estimates the number of states and the respective sum of Gaussian pdfs (the mean vectors and the covariance matrices). The algorithm updates the current map based on the observed features along the robot motion, removing superfluous states (if the current representation covers few observations) or adding new states to increase the precision of the representation;

- Probabilistic approach for localization and navigation - it is covered by an optimized version of Forward-Backward algorithm. In the overall framework, it is necessary to estimate the localization of the robot given the current topological map and the observed features. To deal with the presence of uncertainty associated to the perception and motion, the localization algorithm is implemented with a probabilistic approach, an optimized version of Forward-Backward algorithm that estimates the current robot location in the topological map. This estimation is performed based on the previous state, the observed features and the transition probabilities between states. The navigation, also based on a probabilistic approach, provides the best sequence of states starting in the current estimated state to reach the target state. The algorithm, a revisited Forward-Backward, is used to estimate the best sequence, whose length (the number of states) could be constant or variable according to the time restrictions imposed at the initialization. The sequence of states is translated in motion commands by a behavior approach: an attractive behavior to reach the next state and a repulsive behavior to avoid obstacles;

- Feature extraction and selection - the features are relevant information extracted from rawdata acquired by the sensors. Different types of feature extraction procedures are endowed with a selection criteria. The robot is equipped with different types of sensors that retrieve rawdata, from which the essential information is extracted, leading to the features. In the thesis we use different feature extraction procedures proposed by some authors, since free-area is measured by range sensors and vertical edges, histograms of colors, important regions on the image, are measured by a video camera. Some features retain similar information in same scenarios.

Therefore, the correlation between features is evaluated and the less correlated features are selected to reduce the ambiguity. The correlation is resumed to a single matrix that evaluates the correlations between all the available features.

1.4 Outline of Dissertation

The thesis is structured as follows. After this first chapter of introduction, the thesis focus on the various components of mobile robot navigation. Chapter 2 presents an overview of the three main problems of robot navigation: Localization, Navigation and Map Building. It is explained the importance of each block and the respective order. Since these three main problems are based on an environment representation, this chapter includes a subsection that discusses the possible representations and justifies the adopted one, i.e., the topological representation.

Chapter 3 discusses the localization problem, assuming that the topological map is already known and the navigation is not required at this point. Since the localization consists on estimating the current robot's state in the map, this chapter presents a changed version of the Forward-Backward algorithm that minimizes the uncertainty given the observations.

Still assuming that the map is already known and given the current robot's position, Chapter 4 describes how to get the best path to reach a goal in the map. The chapter also includes the linkage between this high level of abstraction - topological navigation - and a motion control, that supports the motion of the robot along the sequence of states evaluated at the topological navigation. The motion control is based on a sum of behaviors, an attractive behavior to the state goal and a repulsive behavior to avoid obstacles or not desired directions. The algorithm is illustrated with simulated results.

The adopted environment representation, a topological map relies on a set of nodes that represent the states that are characterized by a set of relevant features modeled by mathematical functions: a sum of the n-dimensional Gaussian pdfs, where the dimension corresponds to the number of different features. Chapter 5 describes the algorithm to estimate the number of states and the parameters that define the sum of n-dimensional Gaussian pdfs. A Dynamic Expectation and Maximization algorithm and the Markov Models are the framework used in this chapter.

The algorithms are tested using a real robot of the Rescue Project (described with more detail in Appendix A) and the mobile vehicle ATRV-Jr (see Appendix B). The experimental results in real indoor and outdoor scenarios are shown and discussed in Chapter 6.

The thesis is concluded in Chapter 7 with an evaluation of the approach, underlining the relevant issues of the novelties and discussing the advantages and limitations of the proposed topological approach. The chapter is concluded describing the perspectives of further research.

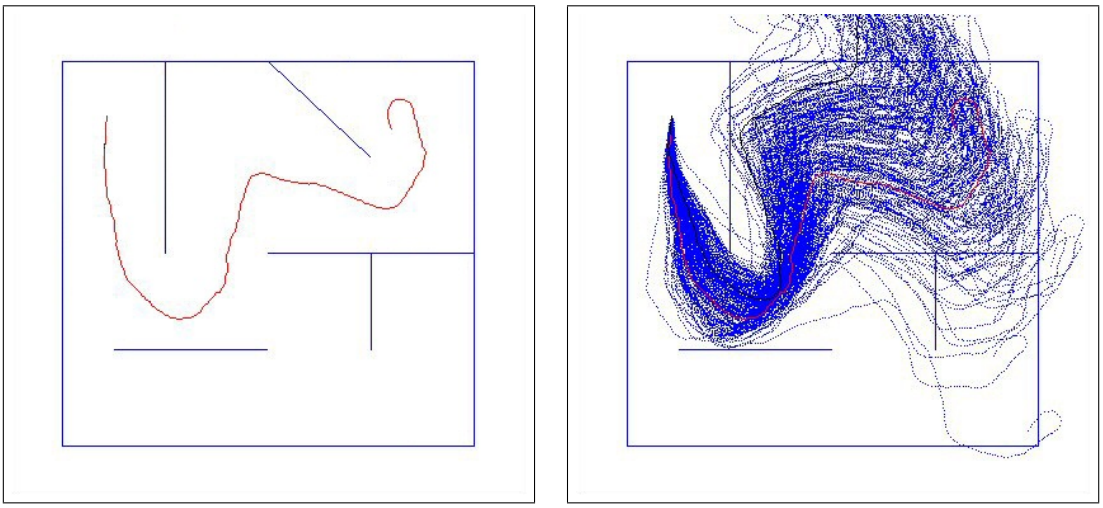
Chapter 2

Mobile Robot Navigation

The approach implemented in a mobile robot is developed to accomplish a given mission in a well known, partially known or unknown environment. As underlined before, the world is not perfect and the mobile robot navigation problem has to deal with the uncertainty included in the world, in the perception of the world (observations and consequently features) and also in its own motion.

Assuming that the robot has no perception of the world, the localization is provided using only the robot's motion. For a robot, with its dynamic and/or kinematic model well known, the scientific explanation is easy: the position estimation is given by inertial systems or by the odometric information, integrating the velocity commands over the time. However, from the command injection to the action result, the uncertainty is always present (wheel slippage, irregularity of the floor, sampling, etc). So, the position estimation given by the odometry integrates the uncertainty over the time. Even for a small uncertainty, it increases uninterruptedly along time and becomes so large than it is impossible to estimate the robot's position. A simple illustration of uncertainty propagation is shown in Figure 2.1 obtained by a particle filter simulation described in [101]. During a period of time it is injected in the mobile robot a sequence of velocity commands to follow a pre-defined trajectory, as presented on the left side of the image. The estimated trajectory, knowing the initial position and using only the dead-reckoning is illustrated on the right side of the Figure 2.1, where the increase of the uncertainty is visible. Each particle simulates the mobile robot position. After a period of time, the cloud is so large that the uncertainty on the real position of the vehicle is very large. If the robot's position is estimated based on that cloud, where the uncertainty is so large as the size of the scenario, it is rough to accomplish any kind of mission.

In the referred relative localization procedures, there is no connection between the robot and the world's referential. The robot's position is relative to the point when the robot was switched on. The localization approach based on odometry and/or inertial systems is commonly defined as dead-reckoning, a method of navigation used in ships, aircrafts, and also mobile robots, when poor information is available. Essentially, it is used to estimate the position based on the distance traveled during short periods of time,



a) b)

Figure 2.1: Illustration of the uncertainty: a) a planned path and b) some estimations of the path followed

where the uncertainty in the odometry acquisitions is still acceptable. In the example illustrated in Figure 2.1 only the odometry information was used and the robot does not have any perception of the world. There is no way to correct the position estimation, as there is no type of environment perception feedback. A trajectory described by the robot and estimated by odometry (even assuming there is no uncertainty) is not anchored to the world. Consequently, the robot can not generally accomplish a mission, where the knowledge of its location relative to a the world's referential is required. The other sensors installed on the robot (e.g., range sensors, image sensors), even though providing noisy data, are important tools to receive feedback from the world. These sensors installed on a robot, acquire measurements of the environment, which are used to build and update the world representation. To accomplish this mission, it is necessary to quantize and correct the error in the observations, to bound the uncertainty increasing and to link the estimated trajectory to the world.

The localization problem is associated with the world representation. Therefore, when using the perception given by the sensors according to the selected environment representation, the localization has to provide the information to bound the increasing uncertainty.

Having selected an appropriate world representation, the next problem to solve in a robotic navigation framework is to build the map that endows the environment representation. This is achieved from sensorial data acquired by the mobile robot during its motion. The mobile robot navigation may start without a map. In this situation, the map building algorithm starts from a single state that corresponds to the place where the robot started. The map construction continues along the robot motion, while it acquires new measurement and discovers new places. However, the mobile robot navigation can start with a previous map. In this case, it is necessary to understand how to start the

localization on the initial map and to progressively update that map. If the localization does not identify the current robot's position in the map, i.e., all the possible locations are equiprobable, the mapping algorithm generates a new location corresponding to the current position. When the localization identifies the current state belonging to the initial map, the map algorithm establishes the connection between the new state (or several states) with the old map.

The environment representation is commonly associated with a geometrical description and a global or local referential, to which the robot's pose (metric information) is related. In most of the SLAM approaches, the world is represented by a set of landmarks identified in a referential. However, the same sensors used to measure the metric distances to the landmarks can also provide different types of information.

The rawdata retrieved directly from the available sensors could be processed and transformed into features, which endow different types of information. The map can be supported by a referential, but with a large number of dimensions and a completely different meaning, with each axis representing a different feature. This map provides a high level of abstraction of the environment, a topological representation. Using the same example described in the Introduction of Chapter 1, to move the robot from "room A" to "room B", it is necessary to identify the rooms, which is done using the features that characterize them, for instance, the color of the walls or patterns. The representation resolution is accomplished by the type and the amount of features extracted. The same features used to navigate between rooms may be insufficient and/or inappropriate to navigate between chairs and tables of a room or even between buildings, where the rooms are located. The appropriated resolution used on the map is a function of the robot, dependent on the target mission.

The topological maps are deeply explained in the Section 2.2 and compared with other types of representations. Before explaining the different types of representations and the adopted one, it is important to understand how the mobile robot navigation problems are related and the importance of each one.

2.1 The three main problems: Localization, Navigation and Mapping

Mobile robot navigation has three main associated problems: Localization, Navigation and Mapping. When the robot is moving, it is necessary to estimate its location (the localization problem), and to compute a path to the goal (the navigation problem). Moreover, if the robot moves, it changes to a new location, where the new acquired observations could be used to improve the current map (the mapping problem). From these three problems emerges the loop in Figure 2.2, that is executed while the robot performs its mission.

This loop does not require a specific sequence, i.e., the three problems can be accomplished at different times. Moreover, each problem can be accomplished at a different

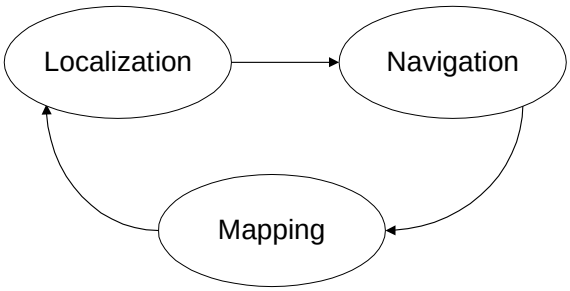


Figure 2.2: The main loop of mobile robot navigation

ratio, which could change dynamically, this meaning that the three components may have different priorities. Given that the robot is moving to reach a specific target or goal, it is important to recognize its position at each time instant and to update the planned path if necessary. Both localization and navigation are based on a map, an environment representation, which is built by the mapping algorithm. It is assumed that the robot does not travel long distances during short periods of time. So, it is expected that the small distance followed by the robot between to consecutive sensor data acquisition leads to a location still covered by the current map. Therefore, the map should be updated when the localization ambiguities occur repeatedly and new types of features are added or any type of feature is removed. Consequently, it is not necessary to run the mapping procedure so often. Moreover, the priority of the mapping component may decrease along time, since the environment becomes well represented by the current map.

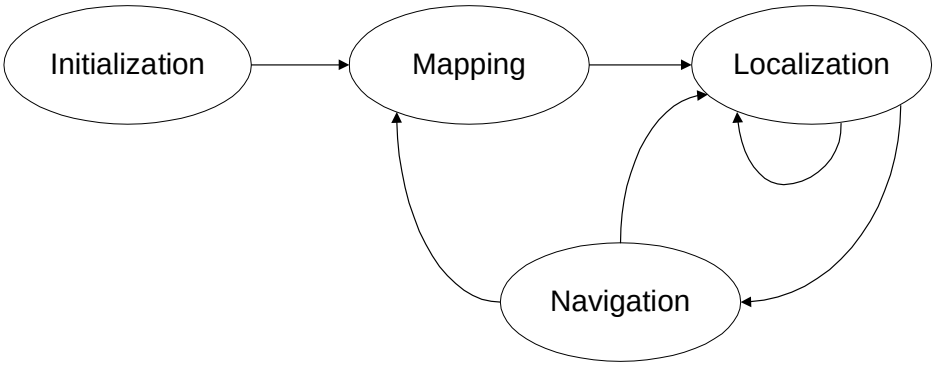


Figure 2.3: The three steps of mobile robot navigation at different rates

The localization requires high priority, since other blocks are dependent on it. The navigation is the next priority, to evaluate the best way to reach the main goal of the mission. Since all the procedures are based on the environment representation, it should be updated even though it is not necessary to run it every time the robot acquires a new observation. Based on this concept, the procedure implemented with the highest rate is the localization, followed by the navigation and finally the mapping, as illustrated in Figure 2.3. If there is no map available at the beginning, the loop must start with the

mapping procedure to provide a first map, or loading a previous estimation of the scenario.

The mobile robot navigation loop includes different elements, which require an initialization procedure. The map, the most import element, is initialized by loading a previous version, if available. Otherwise, it is necessary to select which are the best suited features to represent the scenario (as explained in Section 5.4.3). In the localization, the most important component is the initial robot position in the current map. If not known and assuming a probabilistic approach, the initial position is assumed as equiprobable in the entire map. In the navigation, a target goal is set according to the mission. If there is a map available at the beginning, the target goal can be pointed out in the map, otherwise, it has to be characterized by a set of features understandable by the robot. After the initialization, the loop starts, following the priorities illustrated in Figure 2.3. The localization is running permanently, sometimes interrupted by the navigation algorithm, imposing an updated path based on the current map. The map building algorithm occurs at a slower rate, when compared with the localization and navigation. During the first iterations, i.e., when the robot is switched on and no initial map is available, it is necessary to build a minimal map. In this case, it is imposed the loop closure of Figure 2.2 that all the components should have the same priority, to built an initial map.

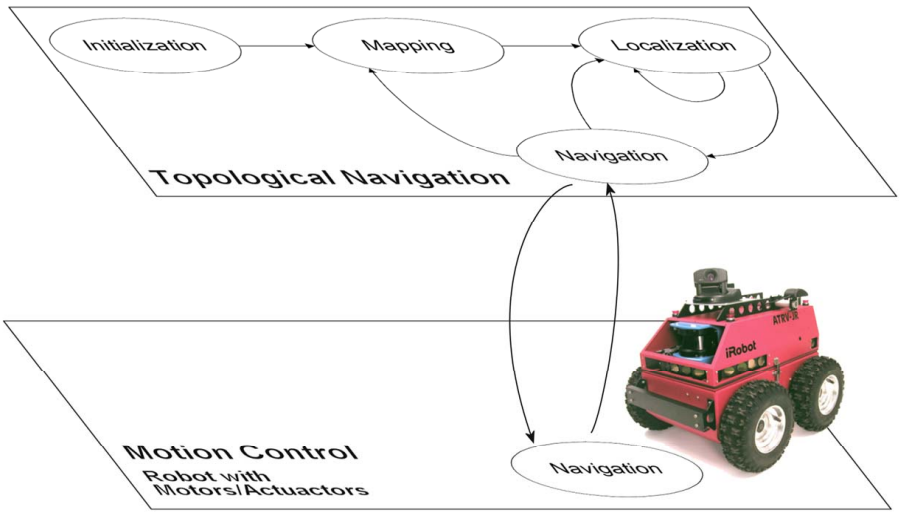


Figure 2.4: The linkage between the topological navigation and motion control

The loop, available at any level of abstraction, can be accomplished by a metric or by a topological approach. However, when a specific approach is selected, the three main problems of mobile robot navigation must be addressed in this same way. If the environment is represented by landmarks in respect to a referential, the localization and navigation have to be accomplished in the same referential. When the localization-navigation-mapping loop is implemented at a high level of abstraction, e.g., at a topological approach, there should exist also an implementation of the loop or part of the loop at a lower level of abstraction, a geometric level, responsible for the local robot motion control. In the thesis the

loop is addressed as a topological approach, but the navigation has also a motion control component to implement the resulted path planned at a topological level, as illustrated in Figure 2.4

The adopted approach strongly depends on the type of the environment representation. Since the loop is based on the environment representation, it is important to take in account the available representations. In particular, the topological approach is the selected representation in the thesis, justifying the selection according to the target scenarios and the main applications.

2.2 Environment Representation

One of the key issues on the research on navigation has been centered on the problem of building full or partial 2D or 3D representations of the environment, which are then used to support the navigation of autonomous robots. However, the navigation problem based on full or partial 2D or 3D maps requires large computational power. This is a severe limitation for systems with limited sensory and computational resources.

Studies of animal navigation suggest that most species utilize a parsimonious combination of perceptual, action and representational strategies that lead to much more efficient solutions when compared to those implemented on today's robots. Many animals combine the local/accurate *vs* global/coarse navigation strategies, as referred in Chapter 1. For instance in [126], there is evidence of animals using landmark-based navigation and (approximate) route integration methods. The distance/accuracy trade-off between long-distance/low-precision and short-distance/high-accuracy mission segments plays an important role in finding efficient solutions to the robot navigation problem. We denote these navigation modes as Global and Local.

However, the navigation depends on the type of world representation adopted. When we mentioned before the landmark-based navigation, the landmarks can be represented by the metric position (as it is common in SLAM) or by specific features (as used by the animals that memorize places by colors, temperatures or magnetic information). The information is sometimes recorded as metric information and other times by particular features.

Current global methods for environment representation can be classified as topological, geometric or hybrid:

- A Topological Map is an adjacency-graph based representation of the environment composed by nodes (or states) and links. Nodes or states represent distinctive places that the system must be able to recognize using external sensors. Links represent ways or motions between nodes and contain procedural information about going from node to node, using either internal and/or external sensors. System localization is defined as the state of the map containing properties that best suit the measurements acquired by the external sensors.

- A Geometric Map is a metric representation of the environment, where the position of landmarks (for instance, corners, walls and doors) are relative to a referential (2D/3D, or even 6D with 3 axis for positioning plus 3 axis for orientation roll/pitch/yaw). The grid maps (two/three dimension cells) are also a common geometric approach, where each cell contains information on its occupancy. System localization is defined in precise metric terms and it is best suited for use with range sensors (laser, ultrasound) and for accurate motion planning.

- Hybrid Maps are grid based local maps connected by topological relations. Each node of a topological map has an associated metric map. For example, an house is an hybrid map: the topological map is defined by the enclosed areas (rooms, corridors, bathrooms) and each state has a metric map, which can be an occupancy grid map to represent the free-area and the obstacles.

2.2.1 Topological Maps

Topological maps make possible the exploration of unknown and unstructured environments in the absence of metric information, being mostly adequate for the representation of outdoor environments. The world is represented by a graph (not necessarily 2D) consisting of a finite number of nodes representing regions, connected by bi-directional links representing ways of motions between regions.

A topological map is also a structured feature map, i.e., a featured-based map, which combines the optimal tracking capabilities of feature maps with the scalability of a topological map. The optimal tracking capabilities result from the fact that the features include more information than the metric location, which reduces the uncertainty perception [68]. The scalability of a topological map depends on the type and amount of features used, which defines the accuracy of the coverage of a large or small physical area. Each node is represented by a set of "nearby" features (e.g., distinct places, regions or landmarks) and each link connects a pair of nodes. When traversing such a graph, the current node is changed when the majority of the observed features belongs to another node. Topological approaches do not require the exact determination of the geometric localization of the robot.

The resolution of a map is proportional to the complexity of the environment representation. Compactness is a key advantage of this type of representation and low computer processing allows fast planning and facilitates interfacing to symbolic planners and problem-solvers. However, with no metric information available, the topological representation requires feature (or landmark) selection/detection/recognition. This means that topological representations are heavily dependent on a powerful system to identify key elements of the environment. As a result, one of the most known localization problems using topological representations occurs when the robot traverses two places that look alike. The topological mapping often has difficulty determining if theses places are the

same or not, particularly if these places have been reached via different motion commands, actions or paths.

Topological maps explore the acquired observations efficiently, because information is represented with accuracy levels adapted to the application needs, whereas geometric maps represent detailed metric information in the map. The resolution of the topological maps is related to the type and amount of information used to represent the nodes and links. Based on a poor selection or small amount of features, the map building algorithm can provide poor topological maps to accomplish a given mission.

Most robotic systems to date use geometric maps, mainly because the commonly used sensors acquire range data (e.g., laser, ultrasound sensors, infra-red). Vision can be an important complementing sensor, motivated by the richness of the visual information, which is extremely important in some animals, as the ants, the honeybees and the migratory birds. In [42] it is shown that many biological systems, ranging from honeybees to humans, use several visual behaviors in their interaction with the environment, but in general visual information is not used to acquire accurate range information. Instead, visual information is used in a more qualitative way with a strong integration in motion control loops, without explicit use of metric information. Some relevant vision based behaviors found in biological systems are trail following aiming at targets, appearance and landmark based guidance, place recognition, among others. For most of these behaviors, environment representations expressed by topological maps are best suited for target applications in outdoor environments. The information involved in topological maps can be expressed qualitatively with relation to explicit sensor data (targets, landmarks, pictorial appearance, tracks, paths, etc) instead of metric information. Furthermore, when considering large complex outdoor environments, geometric maps have an exponential growth in complexity, while topological maps are more computationally efficient.

In [44], a topological map is built by driving a robot through an indoor environment and acquiring grey-scale omni-directional images along the path. A principal component analysis (PCA) is performed on the images to compress the data set, retaining only the image components. At place recognition time, the current image is projected on the components of the PCA space and a qualitative localization is obtained by detecting the nearest neighbors. To move from node to node, local navigation methods are used, such as following corridor guidelines and positioning relative to landmarks.

In [115], the image content is enriched by using a color omni-directional camera. The topological map is acquired manually by driving a robot and grabbing images at a given rate. These images are analyzed and some representative images are selected to represent the nodes of the graph. The recognition phase is done by comparing images acquired online with the images of neighbor nodes, by histogram matching on individual color bands of components HLS and normalized RGB. Histograms are compared with Jeffrey's divergence criteria [97] and images are classified by unanimous voting. This method is inspired in some image-database retrieval techniques but is more efficient because comparison is only

made with images in a neighborhood of the current location.

Works [16, 44, 115] show distinct ways of defining distinctive places. While [16] represents places by the geometric relations between environment features (points, edges) [44, 115] use the global appearance of the scene for comparison with run-time acquired images. Additionally, [44] represents appearance as a PCA basis, which retains geometrical organization of tokens in the images, whereas in [115] that information is lost by computing the color histogram of images. The former may have difficulties when dealing with geometric changes in the environment, while the latter may give ambiguous results on environments with uniform color and luminance characteristics. The combination of both methods in a hierarchical scheme may provide a better solution: the latter would classify the current location in terms of broad classes that present distinct luminance and color distributions (the nodes identifying distinct scenario profiles, e.g., urban, forest, beach).

Many topological maps are built manually in a training phase, without major concerns on the definition of states. Nevertheless, map construction should match the capabilities of the system, i.e., the definition of distinctive places should conform with the system's ability to recognize and navigate between them. Both temporal and spatial distinctiveness are important characteristics for selecting "good" nodes for topological maps. Global vision-based approaches do not have to concern with spatial distinctiveness since the whole image information is used. Local (feature) based approaches must select spatially distinct landmarks for node definition and matching. The following referenced papers address issues concerning the problem of distinctive place definition.

In [43], the temporal distinctiveness problem is addressed. Panoramic images are used in a non-structured environment. Distinctive Places (DP) are identified whenever an image is distinct enough from the previously stored images (snapshots). The distinctiveness is computed from the cross correlation between the current image and the snapshots on the database. A homing behavior is presented that drives the robot towards a DP provided that the robot is close enough to it (catch area).

In [62], the problem of selecting salient and distinctive features from grey-scale images is addressed. Task independence, domain independence and view point invariance are desirable features for the system. Salient features are selected with the Harris Corner Detector (a combined corner and edge detector [50]), which is robust to small changes in view point. Potential landmarks are characterized by a feature vector containing the brightness and its first and second derivatives. They define functions F of the feature vector that are invariant to view point small changes (differential invariant). The matching function is given by the Mahalanobis distance of F at each template point, where the covariance matrix is computed in a training phase. Recognition is then performed with the Nearest Neighbor criteria. The most distinctive landmarks are the ones that have large Mahalanobis distance from all the others.

In [109], panoramic images are also used but instead of using all image data to describe the places, some landmarks are selected from the image. Landmarks selected in this way

can also be used for in-place localization. Landmarks are represented by 16x16 windows. Good landmarks are defined as those having good static and dynamic reliability and that are uniformly distributed in the image. Static reliability is evaluated by a measure of uniqueness of the landmark in a neighborhood, given by the ratio of its auto-correlation and the maximum correlation in the neighborhood. Dynamic reliability is measured by the average of the static reliability along a test trajectory involving view point changes. Uniform distribution is enforced by dividing the image in four sections (front, back, left, right) and selecting four landmarks on each section. The best landmarks, according to this criteria, are used to represent the place. Matching is performed by a normalized correlation to gain some robustness to illumination changes.

A context-free attentional operator based on symmetry is proposed in [99]. This work is not related to navigation but presents a psychologically motivated operator for focusing attention on the visual field. According to the Gestalt psychology [64], symmetry is considered one of the basic principles of perception. Human sense of symmetry is so strong that almost every man-made object is symmetric. Symmetry has been suggested as one of the non-accidental properties of objects which should trigger attention and guide higher level processes. In this sense, symmetry can be used as a property of landmark distinctiveness.

2.2.2 Geometric Maps

A Geometric Map is a metric representation of the environment, relative to a referential, where the information is referred to physical distances and dimensions of places or landmarks (for instance, corners, walls and doors). The referential could be global or local. The referential is global whenever the reference is equal in all maps and anchored to the world and local if it covers a particular place where the origin and orientation of the referential are chosen according to the initial position of the robot. The geometric maps are commonly based on a 2D/3D referential, where the robot pose, position and orientation (or even 6D where 3 axis for positioning plus 3 axis for orientation roll/pitch/yaw) and the position of landmarks are related. Another type of environment representation, also included in the class of geometric maps, are the grid maps. This type of representation is equivalent to a sampling of the world in cells according to a referential. The grid maps consist on two/three dimension cells, where each cell contains information on its occupancy (deterministic or stochastic).

The robot's location based on geometric maps is defined in precise metric terms and is best suited for use with range sensors (laser, ultrasound) and for accurate motion planning. The probabilistic approach to the localization problem is typically more robust than the deterministic approach due to sensor limitations, sensor noise and environment dynamics, as referred in [29]. In [111], Thrun identifies the global localization problem as particularly challenging. Approached probabilistically, the localization problem is a density estimation problem computed using Bayes' rule, the theorem of total probability and

the Markov assumption. Within the context of mobile robot localization, the keywords is often referred as Markov Localization (ML) or Hidden Markov Models (HMM). In addition, the Kalman Filter (KF) is the most popular approach to uncertainty handling, representing the belief on the current location by Gaussian pdfs. As referred in [111], it has also been developed the Bayesian Landmark Learning (BaLL) algorithm, using Bayes' rule for indoor environment applications. This algorithm enables mobile robots to learn what features/landmarks are best suited for the localization process.

In [103], inspired by traditional ship navigation, a coastal navigation algorithm is presented, using Markov Localization (ML) as a probabilistic approach. The motivation for coastal navigation is, more than generating trajectories for the robot that reduce the likelihood of localization error, to build a geometric map where the coast is represented. However, proximity sensors such as laser or ultrasound range-finders have finite range and can not reach landmarks on the coast. Consequently, the geometric maps require odometry and/or inertial systems for dead-reckoning, given the sensors limitations. Experimental results were obtained with the Minerva robot in a museum tour-guide robot (indoor environment). These results demonstrated that a probabilistic approach is robust enough for the navigation using a coastal representation in dynamic environments such as a museum with a significant people density.

In [63], a partially observable Markov decision process (POMDP) is used to estimate the position of the robot in the form of a probability distribution. The algorithm adjusts the probabilities of the initial Markov model by passively observing the robot interactions with its environment representation. Learning is unsupervised and passive, meaning that the robot gets no information from a teacher while it is performing other tasks. This unsupervised passive method is based on the Baum-Welch algorithm, described in [98]. This is a simple expectation-maximization algorithm for learning POMDP from observations.

The most common geometric representation is the occupancy grid maps [35], a high-dimensional space, maintaining all dependencies between neighboring cells. Existing occupancy grid mapping algorithms decompose the high-dimensional mapping problem into a collection of one-dimensional problems, where the occupancy of each grid cell is estimated independently. This induces conflicts that may lead to inconsistent maps, even for noise-free sensors.

Most of the work developed using geometric representations was applied to indoor environments. In [92], Olson describes a probabilistic self-localization technique for mobile robots that is based on the principle of maximum-likelihood estimation. The map may be generated by any method to detect features in the robot's surroundings, using vision, ultrasound sensors, and laser range-finder. The basic idea is to compare a map generated using any method to detect features at the current robot position with a previously generated map. This method is able to operate in both indoor and outdoor environments using geometric maps, mainly of occupancy grids type. A drawback is that the method requires a previous map of the environment, as illustrated by the experimental localization results

for the Sojourner rover on Mars [92, 127].

2.2.3 Hybrid Maps

Hybrid maps aim at combining the advantages of topological and geometric maps. Both approaches exhibit strengths and weaknesses. In [103] an approach that integrates topological maps and grid-based maps is described. In [36], the author presents an approach where a topological based map is extracted from a grid map.

Since the intrinsic geometry of a grid corresponds to the environment geometry, the position and orientation of the robot in the real world can be determined. Topological maps determine the location of the robot relative to the model based on landmarks or distinct sensor features and provides the essential information for a global navigation. However, each node of a topological map may represent a large physical area, where the stated information could not be enough to compute trajectories to operate the robot inside a node. Hybrid maps combine both representations, where the topological component is located at a high level of abstraction and each one of its nodes contains a different and independent geometric map. When the localization algorithm identifies the robot in a different node of the topological map, the metric navigation also changes to a different geometric map associated to this new node of the topological map.

A recent work presented in [16] describes an outdoor vehicle navigating in a park and suburban street environment based on a topologically structured feature map. A laser range finder is used to detect features in the environment (points and edges) and topological nodes are defined as spatial arrangements of features. The topological map is selected manually on a training phase. A data association algorithm allows the system to recognize previously trained places and the navigation inside places is made by metric localization using both odometry and feature tracking data (the features belonging to each state).

The main idea under hybrid map building consists on two steps, shown in Figure 2.5. The first step is the definition of a topological map as a set of nodes and links to accomplish the mission. Second, each node (as a set of nearby features) has an associated geometric map. Each geometric map has a coordinate system, a global/local referential that provides the parametric information to the local navigation, local path planning and obstacle avoidance. Moreover, each navigation planner requires a specific level of environment representation (more abstract, as a topological map or more objective, as a set of landmarks represented in a metric map) [29]. Both topological and geometric maps require features/landmarks processing, which are entirely sensor dependent. In addition, it is necessary to define the links between all the nodes. That connection could be made by motion commands, like go ahead until you find something or move in this direction.

It is common to apply Markov Localization, a probabilistic approach that requires the model of the sensors (often non-linear but assuming a Gaussian approximation) in grid maps. Each grid cell is identified by a combination of feature values (set of features).

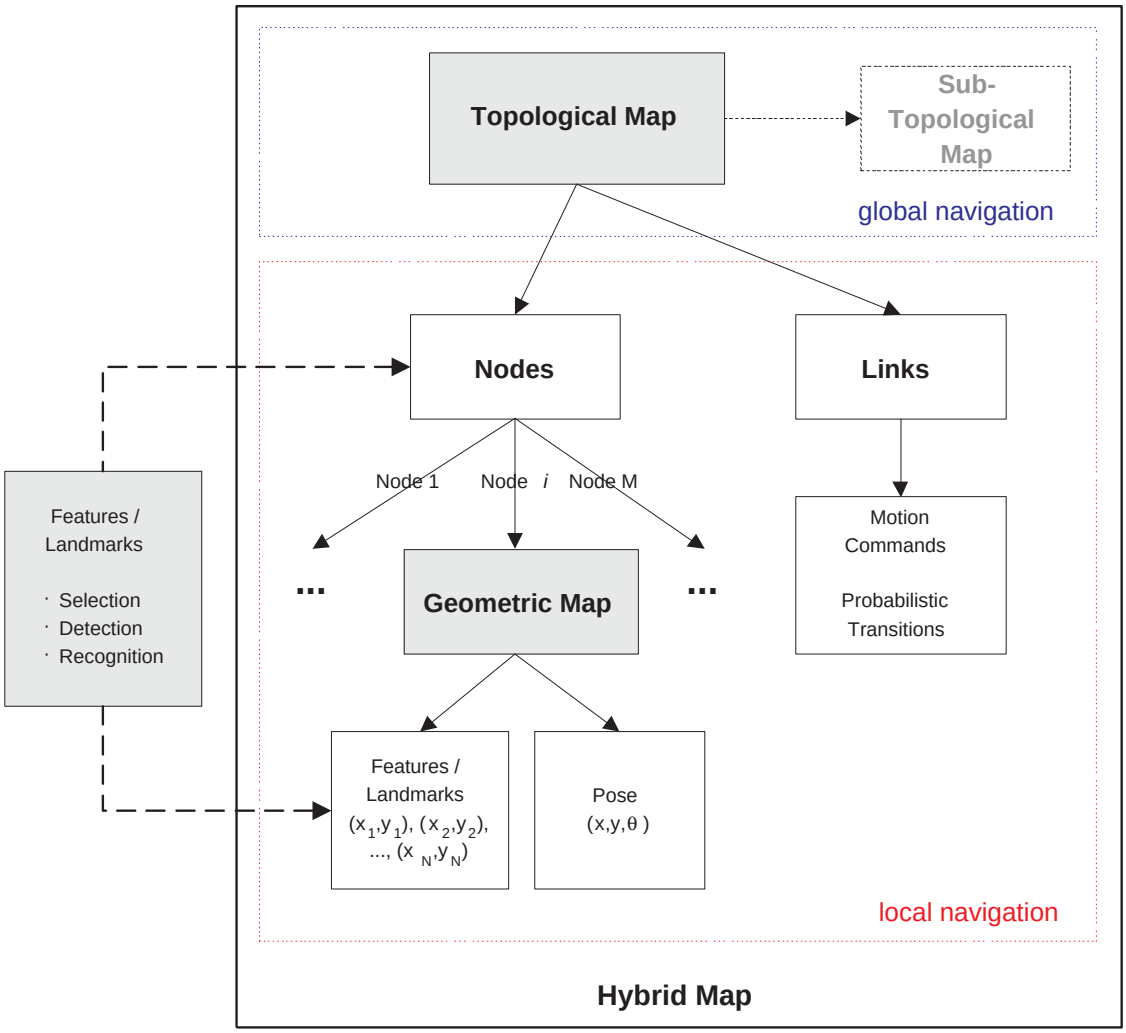


Figure 2.5: Block diagram of topological map building

This type of representation is different from the topological maps, since each grid cell represents a physical area with the same dimensions, connected only with a small number of neighbor cells, while in topological maps the states, representing different places, can all be connected. However, the topological maps still require a know-how to identify and update the Gaussian parameters that define each state, as referred in [98]. Transitions between cells are defined by motion commands (as previously mentioned) and have an associated probability. The transition probabilities can be related to metric distances between nodes unless there is no metric information among nodes. In that case (and without previous information) the probability density function of transitions starts as an uniform distribution that is updated through time, according to the robot's transitions between states.

Therefore, the integration of both map representations (hybrid maps) and the combination of the probabilistic approaches becomes a challenging problem, mainly in outdoors applications. It requires a high capacity of autonomy, when started without an a priori

map. The most promising probabilistic approach is ML, that solves the localization problem based on a probability density function in hybrid representations, as proven by the large number of papers and bibliography. In spite of the large number of experimental results documented, most of them belong to indoor results. In addition, the outdoor environments still require a previous knowledge of the environment to determine (or learn) the optimal parameters of a mathematical support. These approximations define the probability density functions that belong to the Markov model. Using Gaussian approximation is an assumption to overcome the non-linear models.

The features/landmarks selection problem also occurs in hybrid maps. From the different types of available sensors, such as ultrasound sensors, laser, vision, odometry, GPS, compass, inertial, orientation and temperature, and from the observation measurements it is strictly necessary to extract the fundamental information to choose the appropriate features/landmarks and build/update/spread the map. On the other side, in each node of the map is also present metric information

2.2.4 The adopted representation: Topological Maps

A topological map is a representation of an environment with no metric information available, showing physical (natural or artificial) features that characterize particular locations or places. The map expresses a functional relationship among relevant features with a resolution that is proportional to the complexity of the environment's representation. The structure of a topological map relies on a set of nodes that, in this work, represent places in an outdoor environment. Each node is defined as a state of the map and is characterized by a set of relevant features to support the state identification and to avoid mismatching.

One of the most ambitious goals of robotic research is to develop an approach for mobile and completely autonomous robots to operate in outdoor environments, similar scenarios where animals and humans survive. The topological maps are complex enough to provide the crucial information for mobile robots to travel long distances according to the appropriate way and simple to avoid the incumbency of recording all the information over the physical location covered by the map. This type of representation, topological maps, is prepared to cope with the diversity of scenarios, the large spectrum of information acquired by the available sensors, the unexpected events that occur during the operation and, moreover, is robust to scenario changes and the same maps can be shared by different robots.

Furthermore, it is possible to extend a topological map to a hybrid map, building a geometric map associated to each state, providing local navigation. However, the thesis is mainly focused in the capabilities of building a topological representation of the environment and the algorithms to navigate through the map with a topological navigation approach. The novelties are inner to the topological navigation, while the possible geometric navigation in each state could be addressed by one of the several techniques available in the references.

Given the choice of the topological maps as the appropriate environment representation, it is important to describe the mathematical framework that is used from this point forward. The notation used to define a topological map is the following:

- s_i is the state i of the map,
- $S = \{s_1, \ldots, s_N\}$ is the set of states of the map considered herein with N states,
- v_j is the j^{th} feature or attribute, $j = 1, \ldots, M$, that may classify any state s_i,
- $v_j \in V_j$, i.e., the feature j takes values in the set V_j,
- $s_i(v_j)$ is the value of the attribute v_j at state s_i. $s_i(v_j) = \emptyset$ means that the attribute v_j is not existent at state s_i.

In a traditional map of a city, the topological representation is often used as a support for path planning. The relevant interesting places, considered as map states, are emphasized with their most important features like buildings, monuments, shops, stations. The states are linked by roads, streets and rails.

The notation used for the link between states is the following:

- a_{ij} is the transition probability between state i and state j,
- θ_{ij} is the direction between state i and state j, for instance, north, south, east, or other direction.

An example of a topological representation of the map of a city is depicted in Figure 2.6 where seven different states ($S = \{s_1, s_2, \ldots, s_7\}$) are displayed. Three features (v_1, v_2, v_3) are used: transport, building and leisure. The values taken by each feature are, for example, $V_1 = \{$airport, underground, boat, train, parking$\}$, $V_2 = \{$university, castle, church, statue$\}$ and $V_3 = \{$camping, garden, restaurant$\}$.

The main goal of localization, in the context of topological navigation, is related with identifying the state of the map in the closest vicinity of the mobile robot, which is a concept that will be formalized in Chapter 3. To perform topological localization the robot perceives the environment with its on-board sensors and the acquired data is processed aiming at extracting the most relevant features. The robot localizes by a matching between the observations acquired by the sensors and the features represented in the map. This points to the key role played by the sensing processing that collects observations, for which the following related notation will be used:

- r_t is the rawdata or unprocessed data acquired by the robot's sensors at time instant t,
- $o_t = [\ o_t(v_1)\ \ o_t(v_2)\ \ \ldots\ \ o_t(v_M)\ ]$ is a M-dimensional vector of features extracted from r_t,

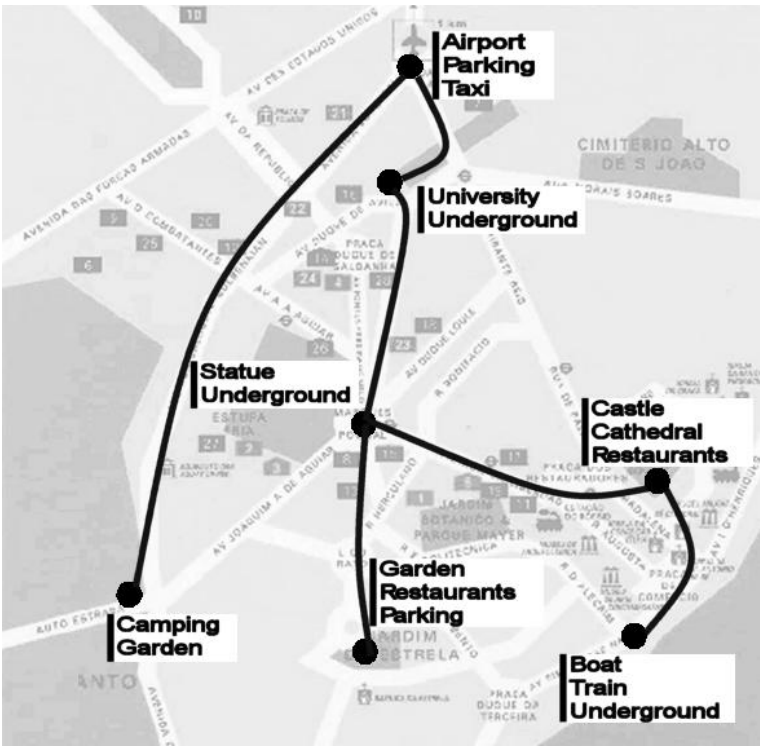


Figure 2.6: An example of a topological map in a city

- $o_t(v_j)$ is the value of the attribute v_j on the feature o_t,
- $o_t(v_j) = \emptyset$ means that observation of the attribute v_j at time instant t is not achieved,
- $O_t = \{o_1, o_2, \ldots, o_t\}$ is a feature sequence up to the time instant t.

As a result of the localization procedure, the current state of the mobile robot is evaluated. The following notation is used:

- q_t is the robot location (state) at time instant t,
- $Q_t = \{q_1, q_2, \ldots, q_t\}$ is a sequence of robot location states up to the time instant t,
- $q_t \in S$.

The localization, explained in Chapter 3, estimates the current robot position, $\hat{q}_t$, which is a state of the topological map. This is notably different from the usual localization procedures that aims at providing a pose estimation in a local or global frame. In fact, when the proposed localization procedure yields a robot estimated location $\hat{q}_t = s_i$ this does not mean that the robot physical location (pose) coincides exactly with that of the environment place from where the features that characterize that state were acquired. This situation is illustrated in Figure 2.7, where, by human perception, three different snapshots taken in different positions identify the same state, the Central Building of IST campus of Alameda.

Figure 2.7: Three different snapshots identify the same state, the Central Building of IST campus of Alameda

2.3 Initial Conditions

The mobile robot navigation is based on the topological representation and consequently, beyond the three problems represented in Figure 2.2, it also includes an initialization procedure to provide an initial map. There are two possibilities to create an initial map:

- loading a previous or predefined map and
- starting from the scratch, with no map.

When no a priori map is available, it is necessary to initialize the number of states, N, and the corresponding states. Since the states are represented by mathematical functions that model the features, it is also necessary to select the initial features, procedure that is described in Subsection 5.4.3. The relevant information present in the rawdata acquired by the robot, i.e., the features, are consequently used by the map algorithm to build and update a topological map, as described in Chapter 5.

Given an initial topological map, it is necessary to understand how to initialize the localization algorithm, the next component of mobile robot navigation represented in the loop of Figure 2.2. Also accomplished with a topological and probabilistic approach, the localization problem is developed over the states that define the map. The initial state probability, π_i, represents the probability that the robot initial state is s_i, this procedure being described in Subsection 5.1.

To start the localization, it is necessary to know the initial robot's position on the map, or the initial state. Similarly to the mapping problem, the initialization of the localization algorithm also includes two possible ways. One, when the initial state is known, which requires an a priori map. The probability π_i is defined as equal to 1, when the initial state is s_i and 0 for the other states. The other possible way for initialization occurs when the initial state is unknown, whether a map is available or not. In the last case, π_i is defined as an uniform distribution for all the states present in the map.

The localization is the first problem addressed in the thesis. Consequently, it is assumed that an initial topological map is available and the localization is presented in Chapter 3,

followed by the navigation, discussed in Chapter 4 and finally the mapping algorithm, in Chapter 5, which provides the topological map.

Chapter 3

Localization

3.1 Problem Statement

As described in Chapter 2, a topological map is a representation of an environment with no metric information available, showing physical features that characterize particular locations or places, defined as states. At this point forward, the mathematical notation becomes more important and it is listed in the Chapter List of Symbols, at the beginning of the thesis. In the current chapter and Chapter 4 it is assumed that the topological map is already known to describe and to simulate the localization and the navigation algorithms. The localization procedure proposed in the thesis considers that, at each time instant, the robot location, q_t, is equal to the map's state in its closest vicinity using a probabilistic approach to decide on this proximity function. The robot estimated location is the map's state that is most likely to have produced the observations acquired by the robot sensors during a given time interval in its operation along which it is necessary to localize it. This is notably different from the usual localization procedures that aim at providing a pose (position and orientation) estimation in a local or global frame associated to a geometric map. In fact, when the proposed localization procedure yields a robot estimated location $\hat{q}_t = s_i$ this does not mean that the robot has always the same pose (physical or geometric location) in the environment place that lead to the map state s_i. This difference between localization in a geometric and in a topological map is illustrated in Figure 3.1.

As a result of the measurements uncertainty, the current robot state estimation can not be performed using a deterministic criteria, as referred in [118]. Consequently, the main issue of the topological localization problem is to find the state that minimizes the uncertainty, given the observations. The state estimation at each time instant t is evaluated using all the available observations during the interval T, $O_T = \{o_1, o_2, \ldots, o_t, \ldots, o_T\}$. The time instant t is not necessarily the current time instant t_a, i.e., $t \leq t_a$ since the localization also estimates the previous robot's locations.

According to a probabilistic approach, the current state estimation, $\hat{q}_t$, is the argument that maximizes the probability of the state given the observation sequence acquired in the time interval T,

$$P(q_t = s_i \mid o_1, \ldots, o_T), \tag{3.1}$$

i.e.,

$$\hat{q}_t = \arg\max_{q_t} P(q_t = s_i \mid o_1, \ldots, o_T). \tag{3.2}$$

According to (3.2), the current state estimation procedure only returns topological map states. Consequently, the robot may have different geometric locations, i.e., be placed in different geometric poses (position and orientation) and be localized in the same topological state, i.e., have the same topological localization. Figure 3.1 illustrates this situation: the robots A, B and C, having different geometric locations but are all in State 1, i.e., have the same topological location, while robot D and E are in different states.

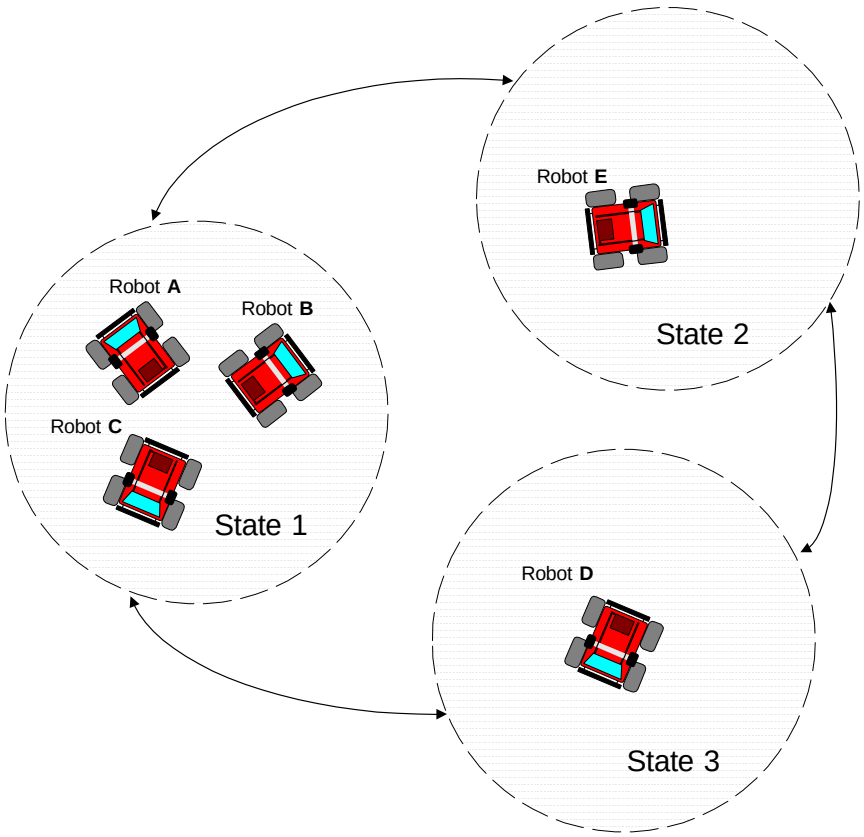


Figure 3.1: Illustration of a topological map with three states where five robots are place with different geometric locations (poses)

Mobile robot sensors acquire rawdata whose nature depends on the types of sensors. For example, vision cameras provide intensity images, while laser range sensors or ultrasound sensors provide range data. Regardless of the types of sensor, it is assumed that rawdata is processed and that the relevant features are extracted. These features are those used as attributes for the characterization of the states of the topological map. In

the context of the localization methodology described in the thesis, (3.2), observations are considered to be the state attributes extracted from sensor data along the mobile robot trajectory. In others words, o_j in (3.2) is assumed to be the features extracted from the sensors rawdata.

To evaluate (3.2) it is necessary to compute the pdf of the state given the observation sequence. The following subsections introduce the Markov Models and the Forward-Backward algorithm that support the evaluation of (3.2).

3.2 Markov Models

The localization process requires the information that results from the processing of the data acquired by the sensors installed on the robot, $(o_1, o_2, o_3, \ldots, o_T)$, since this is the only way to perceive the environment where the robot moves. This knowledge is used to solve (3.2) at each time instant t. Given the large amount of information, the maximization in (3.2) becomes a rough problem to solve. However, given the natural conditions of outdoor environments and the robot dynamics, all the past information can be condensed just into the previous estimation, as depicted in Figure 3.2, which represents the strong Markov assumption. Valid in this context, the Markov assumptions overcomes the problem of solving (3.2), given the large amount of information.

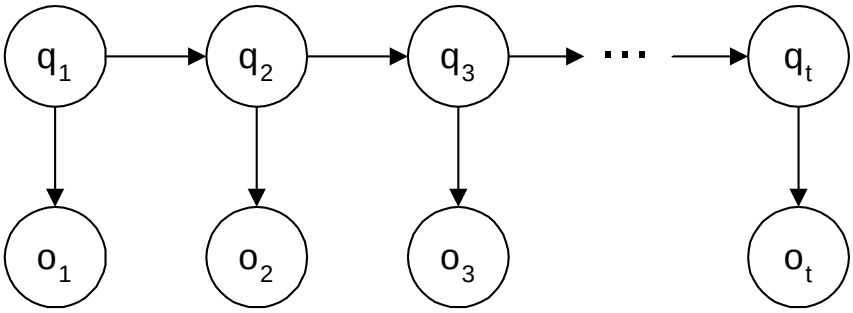


Figure 3.2: Markov Model: states and observations assumptions

The Markov property [61], states that the knowledge of the state at time instant t, q_t, is enough to determine the pdf of the state in future instants, or, in different words, all the information acquired before the time instant t is already reflected in the state estimate q_{t-1}. It important to remember that each state, s_i, does not correspond to a particular pose, but rather to a place characterized by a set of relevant features. In addition, the observation o_t depends only on the state of the mobile robot at the time instant t. This Markov property is graphically illustrated in Figure 3.2, where the variable q_t corresponds to the mobile robot location estimation at time instant t, this corresponding to a state of the topological map.

A Markov Model (MM) [11, 17, 18, 69, 81], is the framework that supports the study of the state evolution along time. The MM is completely characterized by three parameters,

(3.3), (3.5) and (3.8). The first one is the state transition probability density function, a_{ij},

$$a_{ij} \quad = \quad P(q_{t+1} = s_j \mid q_t = s_i), \qquad\qquad i, j = 1, \ldots, N \tag{3.3}$$

changing from the state i to the state j, as illustrated in Figure 3.3. The state transition probability a_{ij} characterizes the edge of the topological map between states s_i and s_j. The value of each transition depends mainly on the distance between the states. However it is important to retrieve particular situations, for instance, two near states with a river in the middle, as an obstacle. The robot can not overcome this obstacle and consequently, the transition probability between these two states is zero. As a first approximation, it is possible to model a_{ij} by

$$a_{ij} = \begin{cases} c & , \quad i = j \\ \frac{1-c}{N-2}\left(1 - \frac{\|s_i - s_j\|}{\sum_{j=1}^{N}\|s_i - s_j\|}\right) & , \quad i \neq j \end{cases} \tag{3.4}$$

where c corresponds to the probability of the robot remaining in the same state and $\| s_i - s_j \|$ is the distance between the features that characterize the states. This distance definition is explained in Chapter 5. The parameters a_{ij} are learned during the navigation procedure, in Chapter 4, according the most probable transitions between the current states of the map.

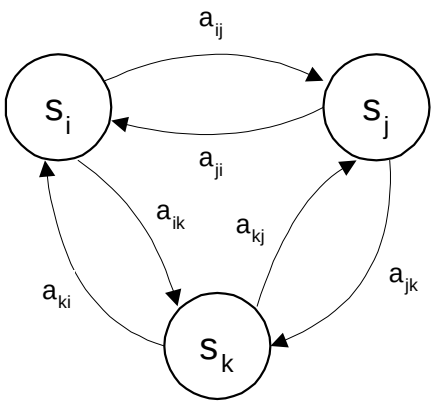


Figure 3.3: Markov Model with 3 states s_i, s_j and s_k and transition probabilities

The second parameter that characterizes the MM is the observation probability density function $b_i(o_t)$, that represents the pdf of observing o_t at time instant t given that the state is s_i,

$$b_i(o_t) \quad = \quad P(o_t \mid q_t = s_i), \quad i = 1, \ldots, N. \tag{3.5}$$

The observation pdf depends on the sensors' model. As proposed in [79], this pdf can

be approximated by a sum of Gaussian pdfs, written as

$$b_i(o_t) = \sum_{l=1}^{G} k_{il}\mathcal{N}(o_t, \mu_{il}, R_{il}) \tag{3.6}$$

with,

$$\mathcal{N}(o_t, \mu_{il}, R_{il}) = \frac{1}{(2\pi)^{\frac{M}{2}}\sqrt{\|R_{il}\|}} \cdot \exp\left\{-\frac{1}{2}(o_t - \mu_{il})^T R_{il}^{-1}(o_t - \mu_{il})\right\}, \tag{3.7}$$

where M is the dimension of the observation vector, which corresponds to the number of distinct features.

To determine the parameters that define the pdfs in (3.7) and also the number of Gaussian pdfs, G (a large value of G corresponds to a good approximation of the model) it is necessary to carry out a statistical characterization of the sensors.

Finally, the initial state probability, π_i, represents the probability that the robot initial state q_1 is s_i, i.e.,

$$\pi_i \quad = \quad P(q_1 = s_i), \quad i = 1, \ldots, N, \tag{3.8}$$

that is defined *a priori*.

The compact notation for the MM characterization is represented by $\lambda = (A, B, \pi)$, where A is a square matrix containing all the probability transitions, $a_{ij}, i, j = 1, \ldots, N$, B is a vector of the observation probability in all possible states and π is a vector that contains the initial localization probability in all possible states, π_i,

$$A \quad = \quad \begin{bmatrix} a_{11} & \ldots & a_{1N} \\ \vdots & a_{ij} & \vdots \\ a_{N1} & \ldots & a_{NN} \end{bmatrix}, \quad B \quad = \quad \begin{bmatrix} b_1(o_t) \\ b_2(o_t) \\ \vdots \\ b_N(o_t) \end{bmatrix}, \quad \pi \quad = \quad \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_N \end{bmatrix}. \tag{3.9}$$

Based on Markov Models, the localization procedure in (3.2) is similar to the high-dimensional maximum likelihood estimation problem, as referred in [98]. This problem is efficiently solved using the Baum-Welch algorithm, as well as the Forward-Backward (FB) algorithm or simply the Alpha-Beta algorithm. The same problem is referred in [113] as a special version of the Expectation and Maximization algorithm.

3.3 Maximization Criteria

The localization problem requires the evaluation of the argument that maximizes

$$\hat{q}_t = \arg\max_{q_t} P(q_t = s_i \mid o_1, \ldots, o_T), \qquad t \leq T. \tag{3.10}$$

To accomplish this goal, the FB algorithm, described in this subsection, is applied along the lines described in [98]. The conditional probability of the current state given the observation sequence during the time interval T is decomposed in $P(o_1, \ldots, o_t, q_t = s_i)$ and $P(o_{t+1}, \ldots, o_T \mid q_t = s_i)$, using the Bayes rule,

$$\begin{aligned} P(q_t = s_i \mid O_T) &= P(q_t = s_i \mid o_1, \ldots, o_T) \\ &= \frac{P(o_1, \ldots, o_T, q_t = s_i)}{P(o_1, \ldots, o_T)} \\ &= \frac{P(o_1, \ldots, o_t, (o_{t+1}, \ldots, o_T), q_t = s_i)}{P(o_1, \ldots, o_T)} \\ &= \frac{P(o_1, \ldots, o_t, q_t = s_i) \cdot P(o_{t+1}, \ldots, o_T \mid o_1, \ldots, o_t, q_t = s_i)}{P(o_1, \ldots, o_T)} \\ &= \frac{P(o_1, \ldots, o_t, q_t = s_i) \cdot P(o_{t+1}, \ldots, o_T \mid q_t = s_i)}{P(O_T)}. \end{aligned} \tag{3.11}$$

Therefore, the probability (3.1) is divided into two main components: one containing the past information, $P(o_1, \ldots, o_t, q_t = s_i)$ and the other the future information, $P(o_{t+1}, \ldots, o_T \mid q_t = s_i)$, as shown in (3.11).

3.3.1 Forward-Backward Algorithm

From inspection, the two probabilities in (3.11) are the keywords in the Forward-Backward algorithm, described in [98]. Let,

$$\alpha_t(i) = P(o_1, \ldots, o_t, q_t = s_i) \tag{3.12}$$

$$\beta_t(i) = P(o_{t+1}, \ldots, o_T \mid q_t = s_i) \tag{3.13}$$

with which the equality (3.11) may be rewritten as,

$$P(q_t = s_i \mid O_T) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(O_T)}. \tag{3.14}$$

In the sequel, we show that $P(O_T)$, in the denominator of (3.14), may be written as a function of the parameters $\alpha_t(i)$ and $\beta_t(i)$ in (3.12) and (3.13). Applying the Bayes rules in the denominator of (3.14) and using (3.12) and (3.13), we successively obtain:

$$\begin{aligned} P(O_T) &= P(o_1, \ldots, o_t, \ldots, o_T) \\ &= \sum_{i=1}^{N} P(o_1, \ldots, o_t, \ldots, o_T, q_t = s_i) \\ &= \sum_{i=1}^{N} P(o_1, \ldots, o_t, q_t = s_i) \cdot P(o_{t+1}, \ldots, o_T \mid o_1, \ldots, o_t, q_t = s_i) \end{aligned}$$

$$= \sum_{i=1}^{N} P(o_1, \ldots, o_t, q_t = s_i) \cdot P(o_{t+1}, \ldots, o_T \mid q_t = s_i)$$
$$= \sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i). \tag{3.15}$$

$P(O_T)$ corresponds to a normalization factor in t, which is defined as $1/\eta_t$. Using (3.15) in (3.14), $P(q_t = s_i \mid O_T)$ becomes:

$$P(q_t = s_i \mid O_T) = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)} = \eta_t \cdot \alpha_t(i) \cdot \beta_t(i). \tag{3.16}$$

One of the probabilities in (3.14), the component $\alpha_t(i)$, is relative to all the observations from the past up to the time instant t. On the other hand, all the future observations are in the $\beta_t(i)$ parameter. The Figure 3.4 illustrates the influence of the past, α, and the future, β, on (3.1).

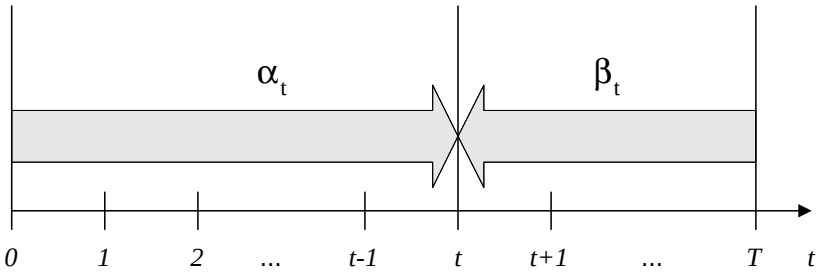


Figure 3.4: Influence of the past, α, and the future, β, at time instant t

The main result expressed in (3.16) is that there is a complete decoupling of the observation sequence relative to the time instant t. The observations prior to t (past observations) and the observations from t to T (future observations) appear in different factors. This decomposition is illustrated in Figure 3.4, where the pdf represented by α and β parameters are represented during a time interval T.

Whenever t is the current time instant and $t \leq T$, it is reasonable to consider that there is no future observations up to T, and therefore $\beta_t(i)$ is considered to have an uniform distribution, this expressing the absence of future information. On the contrary, whenever the current time instant is t_a with $t < t_a \leq T$, the state estimate at time instant t can profit from the observations from t to t_a.

To evaluate the pdf (3.16) for each time instant t, it is necessary to evaluate $\alpha_t(i)$ and $\beta_t(i)$. The Forward-Backward algorithm [98], provides an iterative solution to this problem. The forward and backward iterations of the algorithm are represented in (3.18) and (3.19), respectively,

$$\begin{aligned}\alpha_t(j) &= P(o_1,\ldots,o_t,q_t=s_j)\\ &= \left[\sum_{i=1}^{N} P(o_1,\ldots,o_{t-1},q_{t-1}=s_i)\cdot P(q_t=s_j \mid q_{t-1}=s_i)\right]\cdot P(o_t \mid q_t=s_j)\\ &= \left[\sum_{i=1}^{N}\alpha_{t-1}(i)a_{ij}\right]\cdot b_j(o_t), \end{aligned} \tag{3.17}$$

with $b_j(o_t)$ defined in (3.5). For time instant $t+1$, (3.17) is written as,

$$\begin{aligned}\alpha_{t+1}(j) &= \left[\sum_{i=1}^{N} P(o_1,\ldots,o_t,q_t=s_i)\cdot P(q_{t+1}=s_j \mid q_t=s_i)\right]\cdot P(o_{t+1} \mid q_{t+1}=s_j)\\ &= \left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]\cdot b_j(o_{t+1}). \end{aligned} \tag{3.18}$$

In (3.18) it was used j in $\alpha_{t+1}(j)$ to keep the notation of the a_{ij} parameter in the sum inside the square brackets.

$$\begin{aligned}\beta_t(i) &= P(o_{t+1},\ldots,o_T \mid q_t=s_i)\\ &= \sum_{j=1}^{N} P(q_{t+1}=s_j \mid q_t=s_i)\cdot P(o_{t+1} \mid q_{t+1}=s_j)\cdot P(o_{t+2},\ldots,o_T \mid q_{t+1}=s_j)\\ &= \sum_{j=1}^{N} a_{ij}\cdot b_j(o_{t+1})\cdot\beta_{t+1}(j) \end{aligned} \tag{3.19}$$

Equation (3.18) evolves forward in time while (3.19) evolves backwards. The forward and backward iterations are initialized at time instant $t=1$ and $t=T$, respectively. The initialization of (3.18) requires *a priori* information in the variable π, as expressed in (3.20). The initialization of the variable β corresponds to an uniform distribution, as stated in (3.21),

$$\alpha_1(i) = P(q_1=s_i)\cdot P(o_1 \mid q_1=s_i)=\pi_i\cdot b_i(o_1), \tag{3.20}$$

$$\beta_T(i) = 1,\quad i=1,\ldots,N. \tag{3.21}$$

As shown in the second line of (3.15),

$$P(O_T)=\sum_{i=1}^{N} P(o_1,\ldots,o_t,\ldots,o_T,q_t=s_i),$$

the probability of the observation sequence, $P(O_T)$, can be evaluated by

$$P(O_T) = \sum_{i=1}^{N} \alpha_T(i), \tag{3.22}$$

when no future information is available, i.e., t is equivalent to T.

In summary, the localization procedure consists on finding the best state estimate, given the observations, (3.2). Replacing the pdf (3.2) by (3.16) and applying the FB algorithm, (3.18), (3.19), (3.20) and (3.21), the equation (3.2) is modified to,

$$\hat{q}_t = \arg\max_{q_t = s_i} \left[\alpha_t(i) \cdot \beta_t(i)\right]. \tag{3.23}$$

The best state estimate for each t, $\hat{q}_t$, during a time interval T, is the state s_i, i.e., $\hat{q}_t = s_i$, that maximizes $\alpha_t(i) \cdot \beta_t(i)$.

For a sequence of estimations, $\hat{Q}_T = \{\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_T\}$, incongruent sequences might occurs, when some of the probability transitions are equal to zero. To solve this problem, the optimization criteria must be changed. In (3.23) the argument is a sequence of states, $\hat{Q}_T$, that maximizes the probability in (3.24), similar to the Viterbi algorithm, see [98].

$$\hat{Q}_T = \arg\max_{q_1,\ldots,q_T} P(Q_T \mid o_1, \ldots, o_T) \tag{3.24}$$

3.3.2 Forward-Backward Algorithm Revisited

In the FB algorithm described in the previous subsection, the time interval of length T from $t = 0$ to $t = T$ has a fixed length, while t is the unique variable of time. For long time intervals, corresponding to large operating periods, the FB algorithm implementation becomes too time consuming. For this reason, it is necessary to understand the dependence of the algorithm on the length of the observation sequence and to find an iterative procedure to implement the localization in a feasible way.

Given the importance of the length of the observation sequence, we introduce a new notation on associated to the parameters α and β, explicitly including the time interval T in superscript. Rewriting the equations (3.12), (3.13) and the iterations, (3.18) and (3.19), yields,

$$\alpha_t^T(j) = P(o_1, \ldots, o_t, q_t = s_j) \tag{3.25}$$

$$\beta_t^T(i) = P(o_{t+1}, \ldots, o_T \mid q_t = s_i) \tag{3.26}$$

$$\alpha_{t+1}^T(j) = \left[\sum_{i=1}^{N} \alpha_t^T(i) a_{ij}\right] \cdot b_j(o_{t+1}) \tag{3.27}$$

$$\beta_t^T(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}^T(j), \tag{3.28}$$

that will be used along this section.

To evaluate $\alpha_t^T(j)$ and $\beta_t^T(i)$ as a function of T, consider, as an example, two distinct values for T, T_1 and T_2, with $T_1 \leq T_2$.

Result: Consider $T_1, T_2 \in I\!N$, $T_1 \leq T_2$ and $t \leq T_1$. Then $\alpha_t^{T_1}(j) = \alpha_t^{T_2}(j), \forall j$.

Proof: Consider the sequence of observations up to time instants T_1 and T_2, $O_{T_1} = \{o_1, \ldots, o_{T_1}\}$ and $O_{T_2} = \{o_1, \ldots, o_{T_2}\}$, respectively. Using the FB initialization, (3.20) and (3.21),

$$\alpha_1^T(i) = P(q_1 = s_i) \cdot P(o_1 \mid q_1 = s_i)$$

from where,

$$\alpha_1^{T_1}(i) = \alpha_1^{T_2}(i),$$

which is equivalent to

$$\alpha_1^{T_1}(j) = \alpha_1^{T_2}(j).$$

Moreover, by mathematical induction, if $\alpha_t^{T_1}(j) = \alpha_t^{T_2}(j)$ and taking into account the iteration (3.18), it is shown that,

$$\begin{aligned}
\alpha_{t+1}^{T_1}(j) &= \left[\sum_{i=1}^{N} P(o_1, \ldots, o_t, q_t = s_i) \cdot P(q_{t+1} = s_j \mid q_t = s_i)\right] \cdot P(o_{t+1} \mid q_{t+1} = s_j) \\
&= \left[\sum_{i=1}^{N} \alpha_t^{T_1}(i) a_{ij}\right] \cdot b_j(o_{t+1}) \\
&= \left[\sum_{i=1}^{N} \alpha_t^{T_2}(i) a_{ij}\right] \cdot b_j(o_{t+1}) \\
&= \alpha_{t+1}^{T_2}(j), \qquad t \leq T_1
\end{aligned}$$

which concludes the proof, i.e.,

$$\alpha_t^{T_1}(j) = \alpha_t^{T_2}(j), \qquad t \leq T_1, T_1 \leq T_2. \tag{3.29}$$

□

Result: Consider $T_1, T_2 \in I\!N$ with $T_1 < T_2$ and $t \leq T_1$. Then $\beta_t^{T_1}(i) = \beta_t^{T_1}(i) \cdot \beta_{T_1}^{T_2}(i), \forall i$.

Proof: Consider $\beta_t^{T_1}(i)$ for $t \leq T_1$ and $\beta_t^{T_2}(i)$ for $t \leq T_2$, given by:

$$\begin{aligned}
\beta_t^{T_1}(i) &= P(o_{t+1}, \ldots, o_{T_1} \mid q_t = s_i) \\
\beta_t^{T_2}(i) &= P(o_{t+1}, \ldots, o_{T_1}, \ldots, o_{T_2} \mid q_t = s_i) \\
&= P(o_{t+1}, \ldots, o_{T_1}, (o_{T_1+1}, \ldots, o_{T_2}) \mid q_t = s_i) \\
&= P(o_{t+1}, \ldots, o_{T_1} \mid o_{T_1+1}, \ldots, o_{T_2}, q_t = s_i) \cdot P(o_{T_1+1}, \ldots, o_{T_2} \mid q_t = s_i) \\
&= P(o_{t+1}, \ldots, o_{T_1} \mid q_t = s_i) \cdot P(o_{T_1+1}, \ldots, o_{T_2} \mid q_t = s_i)
\end{aligned}$$

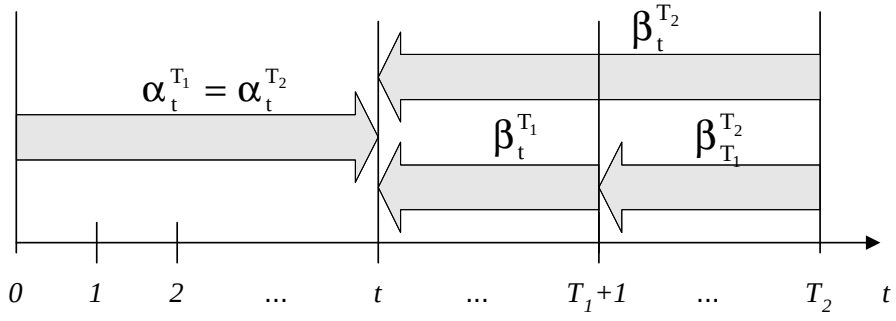


Figure 3.5: The past and future influence at time instant t. The future is divided in two sub-intervals

$$= \beta_t^{T_1}(i) \cdot \beta_{T_1}^{T_2}(i)$$

which concludes the proof, i.e.,

$$\beta_t^{T_2}(i) = \beta_t^{T_1}(i) \cdot \beta_{T_1}^{T_2}(i), \qquad t \leq T_1, T_1 \leq T_2. \tag{3.30}$$

□

As illustrated in Figure 3.5, $\alpha_t^{T_1}$ is equal to $\alpha_t^{T_2}$, for $t \leq T_1$, since both consider the same past information, while $\beta_t^{T_2}$ is decomposed in two components: the first contains the information of the interval from t to T_1 and the second results from the observations acquired during the fixed interval from T_1 to T_2.

If T_1 is constant and T_2 changes, to determine $\beta_t^{T_2}(i)$ in (3.30) only requires the evaluation of $\beta_{T_1}^{T_2}(i)$, since $\beta_t^{T_1}(i)$ remains unchanged. Assuming T_1 as a fixed time interval T and T_2 as $T + k\Delta T$, then (3.30) is replaced by:

$$\beta_t^{T+k\Delta T}(i) = \beta_t^{T}(i) \cdot \beta_T^{T+k\Delta T}(i) \tag{3.31}$$

where ΔT corresponds to the time interval between successive observations and $k = 1, 2, 3, \ldots$. The decomposition of β in two components reduces the computational costs, as it is only necessary to evaluate the innovative component of β, i.e., what is new between T and $T + k\Delta T$.

Moreover, it is possible to generalize (3.31) for several observations, i.e., for different values of k:

$$\begin{aligned}
k = 1, \quad \beta_t^{T+\Delta T}(i) &= \beta_t^{T}(i) \cdot \beta_T^{T+\Delta T}(i) \\
k = 2, \quad \beta_t^{T+2\Delta T}(i) &= \beta_t^{T}(i) \cdot \beta_T^{T+2\Delta T}(i) \\
&= \beta_t^{T}(i) \cdot \beta_T^{T+\Delta T}(i) \cdot \beta_{T+\Delta T}^{T+2\Delta T}(i) \\
k = 3, \quad \beta_t^{T+3\Delta T}(i) &= \beta_t^{T}(i) \cdot \beta_T^{T+3\Delta T}(i)
\end{aligned}$$

$$
\begin{aligned}
&= \beta_t^T(i) \cdot \beta_T^{T+\Delta T}(i) \cdot \beta_{T+1\Delta T}^{T+3\Delta T}(i) \\
&= \beta_t^T(i) \cdot \beta_T^{T+\Delta T}(i) \cdot \beta_{T+1\Delta T}^{T+2\Delta T}(i) \cdot \beta_{T+2\Delta T}^{T+3\Delta T}(i)
\end{aligned}
$$

$$\cdots$$

$$k = n, \quad \beta_t^{T+n\Delta T}(i) = \beta_t^T(i) \cdot \prod_{k=0}^{n} \beta_{T+(k-1)\Delta T}^{T+k\Delta T}(i) \tag{3.32}$$

or

$$k = n, \quad \beta_t^{T+n\Delta T}(i) = \beta_t^{T+(n-1)\Delta T}(i) \cdot \beta_{T+(n-1)\Delta T}^{T+n\Delta T}(i). \tag{3.33}$$

Replacing $\beta_{T+(k-1)\Delta T}^{T+k\Delta T}(i)$ in (3.32) by (3.26) and consequently by (3.28) yields,

$$
\begin{aligned}
k = n, \quad \beta_t^{T+n\Delta T}(i) &= \beta_t^T(i) \cdot \prod_{k=0}^{n} P(o_{T+k\Delta T} \mid q_{T+k\Delta T} = s_i) \\
&= \beta_t^T(i) \cdot \prod_{k=0}^{n} \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{T+k\Delta T}) \\
&= \beta_t^T(i) \cdot \sum_{j=1}^{N} a_{ij} \prod_{k=0}^{n} b_j(o_{T+k\Delta T})
\end{aligned} \tag{3.34}
$$

or

$$
\begin{aligned}
k = n, \quad \beta_t^{T+n\Delta T}(i) &= \beta_t^{T+(n-1)\Delta T}(i) \cdot P(o_{T+n\Delta T} \mid q_{T+n\Delta T} = s_i) \\
&= \beta_t^{T+(n-1)\Delta T}(i) \cdot \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{T+n\Delta T}).
\end{aligned} \tag{3.35}
$$

Given

$$P(o_{T+n\Delta T} \mid q_{T+n\Delta T} = s_j) = b_j(o_{T+n\Delta T}) \leq \delta_j < 1,$$

the equation (3.34) and (3.35) are rewritten as:

$$k = n, \quad \beta_t^{T+n\Delta T}(i) \leq \beta_t^T(i) \cdot \sum_{j=1}^{N} a_{ij} \delta_j^n \tag{3.36}$$

or

$$k = n, \quad \beta_t^{T+n\Delta T}(i) \leq \beta_t^{T+(n-1)\Delta T}(i) \cdot \sum_{j=1}^{N} a_{ij} \delta_j. \tag{3.37}$$

It is important to understand the relation between β and δ. From (3.3), it is obvious

that $\sum_{j=1}^{N} a_{ij} = 1$. Since $\delta_j < 1, \forall j$, yields

$$\sum_{j=1}^{N} a_{ij}\delta_j = \tau < 1,$$

where τ is a factor that represents the importance of β through time. From (3.37) it is clear that

$$\begin{aligned} \beta^{T+n\Delta T}(i) &\leq \beta_t^{T+(n-1)\Delta T}(i) \cdot \sum_{j=1}^{N} a_{ij}\delta_j \\ &\leq \beta_t^{T+(n-1)\Delta T}(i) \cdot \tau \\ &\leq \tau^n, \end{aligned} \tag{3.38}$$

which is bounded by an exponential function.

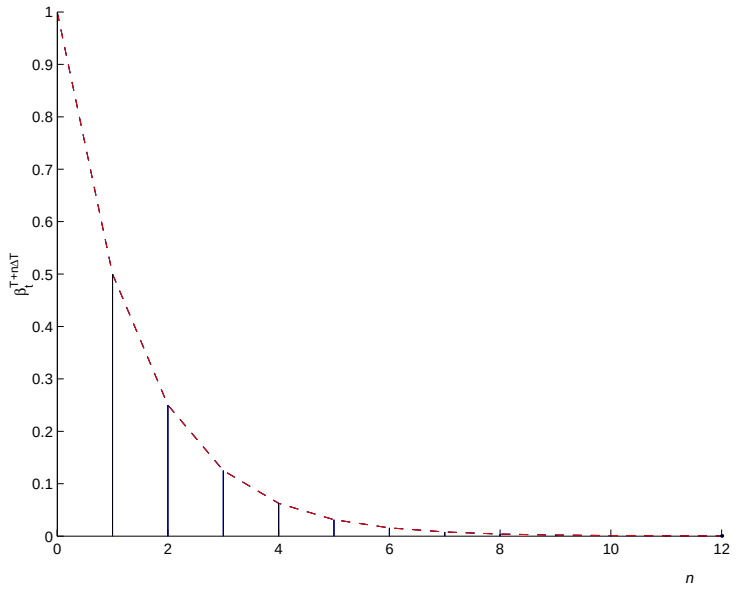


Figure 3.6: The evaluation of $\beta_t^{T+n\Delta T}$ during 12 time intervals, with $\tau = 0.5$

The influence of the observations acquired up to the time instant t decreases exponentially. Figure 3.6 illustrates the evolution of $\beta_t^{T+n\Delta T}$ with $\tau = 0.5$. After 4 time intervals the effects of the parameter β is less than 10%. Therefore, the time interval T has to be selected according to the decay referred in (3.38).

3.3.3 Time Interval Dimension

As explained in the previous section, the localization is accomplished using the information returned by the observed features. However, the influence of that information decreases

along time, i.e., an observation acquired in a far away instant of time is less important than an observation acquired in a closer instant. Therefore, it is not suitable to equally integrate all the information acquired up to the time instant t to estimate the robot's location at t. There is an interval where the information is more relevant for the localization algorithm. As illustrated in Figure 3.6, the influence may be less then 10% after 4 iterations or less than 1% after 8 iterations.

Consider now that the overall observation sequence interval is sampled in time intervals of length T, as illustrated in Figure 3.7, with each window represented by $kT + T$, for $k = 1, 2, \ldots$.

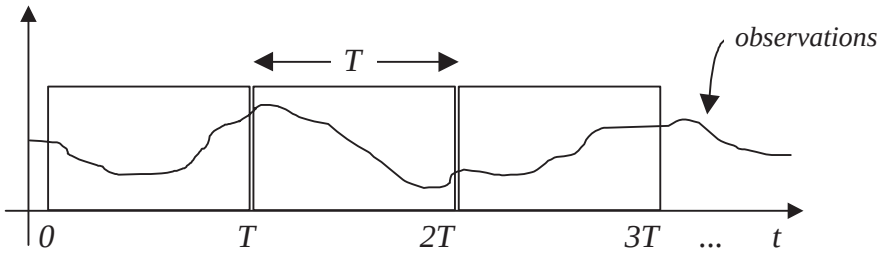


Figure 3.7: Observation sequence divided in time intervals of length T

Using the introduced notation, this yields:

$$\begin{aligned} O_{kT+T} &= \{o_1, \ldots, o_{kT}, o_{kT+1}, \ldots, o_{kT+T}\} \\ &= \{O_{kT}, o_{kT+1}, \ldots, o_{kT+T}\}. \end{aligned} \tag{3.39}$$

Defining

$$O_{kT}^{kT+T} = \{o_{kT+1}, \ldots, o_{kT+T}\},$$

the observation sequence up to $kT + T$ is written as

$$O_{kT+T} = \{O_{kT}, O_{kT}^{kT+T}\}. \tag{3.40}$$

The probability of the current state given the observation sequence, $\{O_{kT}, O_{kT}^{kT+T}\}$, along the acquired time interval from kT to $kT + T$ is given by:

$$\begin{aligned} P(q_t = s_i \mid O_{kT+T}) &= P(q_t = s_i \mid o_1, o_2, \ldots, o_{kT}, o_{kT+1}, o_{kT+2}, \ldots, o_{kT+T}) \\ &= P(q_t = s_i \mid o_{kT}, o_{kT+1}, o_{kT+2}, \ldots, o_{kT+T}) \\ &= P(q_t = s_i \mid o_{kT}, O_{kT}^{kT+T}) \\ &= \frac{\alpha_t^{kT+T}(i) \cdot \beta_t^{kT+T}(i)}{P(O_{kT+T})} \end{aligned} \tag{3.41}$$

where α and β are expressed by

$$\alpha_t^{kT+T}(i) = P(o_{kT}, \ldots, o_t, q_t = s_i) \tag{3.42}$$
$$\beta_t^{kT+T}(i) = P(o_{t+1}, \ldots, o_{kT+T} \mid q_t = s_i) \quad . \tag{3.43}$$

Both equations require that $kT \leq t \leq kT + T$. To evaluate the pdf (3.41) for each time instant t, it is necessary to evaluate $\alpha_t^{kT+T}(i)$ and $\beta_t^{kT+T}(i)$ in (3.42) and (3.43), respectively.

At each interval of length T, the initialization of both parameters is required. The initialization values for (3.42) and (3.43) are presented in (3.44) and (3.45), where we use the fact that the $P(q_{kT+1} = s_i) = P(q_{kT} = s_i)$,

$$\begin{aligned} \alpha_{kT+1}^{kT+T}(i) &= P(q_{kT+1} = s_i) \cdot P(o_{kT+1} | q_{kT+1} = s_i) \\ &= \pi_i^{kT+T} \cdot b_i(o_{kT+1}) \qquad (3.44) \\ \beta_{kT+T}^{kT+T}(i) &= 1\,, \quad \leq i \leq N. \qquad (3.45) \end{aligned}$$

The equations (3.42) and (3.43) are similar to the forward and backward iterations in (3.18) and (3.19) and consequently, (3.42) and (3.43) can be rewritten as follows:

$$\alpha_{t+1}^{kT+T}(j) = \left[\sum_{i=1}^{N} \alpha_t^{kT+T}(i) \cdot a_{ij}\right] \cdot b_j(o_{t+1}) \tag{3.46}$$
$$\beta_t^{kT+T}(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}^{kT+T}(j) \quad . \tag{3.47}$$

Further developments on the parameter β in (3.47) are still possible. By inspection on (3.35) and (3.33), (3.47) can be written as

$$\beta_t^{kT+T}(i) = \beta_t^{(k-1)T+T}(i) \cdot \beta_{(k-1)T+T}^{kT+T}(i) \tag{3.48}$$
$$\beta_t^{kT+T}(i) = \beta_t^{(k-1)T+T}(i) \sum_{j=1}^{N} a_{ij} b_j(o_{kT+T}). \tag{3.49}$$

Since the parameter $b_j(o_{kT+T})$ is equal to $P(o_{kT+T} \mid q_{kT+T} = s_j) = \delta_j$, (3.49) is rewritten as shown in (3.51), where $\delta_j < 1, \forall j$. From (3.3), $\sum_{j=1}^{N} a_{ij} = 1$, and so

$$\sum_{j=1}^{N} a_{ij} \cdot \delta_j = \tau < 1. \tag{3.50}$$

Equivalently to (3.36) and (3.37),

$$\begin{aligned} \beta_t^{kT+T}(i) &\leq \beta_t^{(k-1)T+T}(i) \cdot \sum_{j=1}^{N} a_{ij} \cdot \delta_j \qquad (3.51) \\ &\leq \beta_t^{(k-1)T+T}(i) \cdot \tau \\ \beta_t^{kT+T}(i) &\leq \tau^k, \qquad (3.52) \end{aligned}$$

which means that β is bounded by a decreasing power series with time constant τ. This is the clue to choose the appropriate time interval T, which is defined according to the values of a_{ij} already known.

The necessary tools to evaluate the localization probability were described. To implement this methodology, it is first necessary to define a time interval T. Second, along each time interval the α and β parameters are calculated using the observations and iterative equations. The localization probability for each state, $q_t = s_i, \;\; i = 1, 2, \cdots, N$ is evaluated using the FB algorithm. The state that maximizes the probability is defined as the current robot location estimate.

3.4 Simulation Results

This section presents experimental simulated results on the localization procedure described in this chapter with the revisited FB algorithm. In the experiments the environment is represented by a topological map with 6 states, $s_1, s_2, \ldots, s_6$, each one characterized by a set of five different attributes, $v_1, v_2, \ldots, v_5$. Two different paths are considered in this map, as represented in Figures 3.8 and 3.20.

Even though each state may, in general terms, be characterized by a sum of Gaussian pdfs, the present simulations consider, for simplicity, that each state is defined by a single Gaussian pdf. The corresponding mean is a 5-dimensional vector collecting the attribute values, represented by colored bars as shown in Figure 3.8. The covariance matrix is diagonal to express a null correlation among the attributes.

To implement the simulation and assess the localization performance it is necessary to define a metric referential that provides the exact position and orientation of the mobile robot along its path and the exact position of each map's state. It should be reinforced, that the topological localization procedure proposed in the thesis does not require this metric information, which is exclusively used to simulate the observations, to accomplish the path planning and to evaluate the results. The tests consider that the mobile robot is equipped with a range sensor whose statistical model is also one dimensional Gaussian pdf. It is considered that the probability that the mobile robot observes the attributes of a state is proportional to the inverse of the Euclidean distance to the state. Additionally, it is assumed that the robot's sensor measures, in each time instant, all the attributes of the closest state.

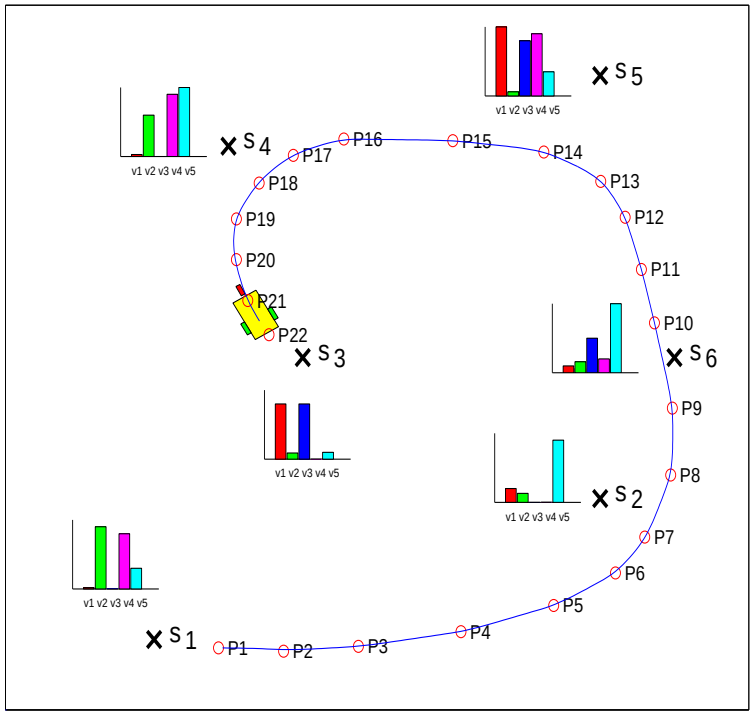


Figure 3.8: Experiment 1: Map states, via points, attributes of each state and the executed path

At this stage, the performance analysis of the localization procedure requires the *a priori* definition of the path the mobile robot will follow. This path is specified by the set of via points, P_i, illustrated in Figures 3.8 and 3.20. In the complete application of the algorithm, the path is defined as a sequence of states that result from the topological navigation procedure and described in Chapter 4.

As a consequence of the disparity of the probability values returned by the localization (determined by the exponential decreasing referred in (3.52), one or two states with probability while the other states have low probabilities) the experimental results are displayed in logarithmic scale, $\log P(\cdot)$. At each time instant t, the current location estimate corresponds to the state with the highest probability value. The results displayed in Figures 3.9-3.19 and 3.21 represent the evolution of $\log P(q_t = s_i \mid o_1, \ldots, o_T)$ for $i = 1, 2, \ldots, 6$ along the simulation iterations. The small circles in each function are numbered according to the via points displayed in Figures 3.8 and 3.20.

The first simulated results, obtained with a standard deviation σ on the observation, are shown in Figure 3.9. The value of σ is not mentioned, since the goal is to compare the algorithm performance with different standard deviations, namely 2σ, 4σ and 8σ compare with the first experiment. The $P(q_t = s_i)$ is higher when the mobile robot is in the close vicinity of state s_i. However, in this situation, the probability function displays some peak values. These peaks are periodic with period related with the length of the observation time interval, T (defined in Subsection 3.3.1), as illustrated in Figures 3.9-3.12 with T

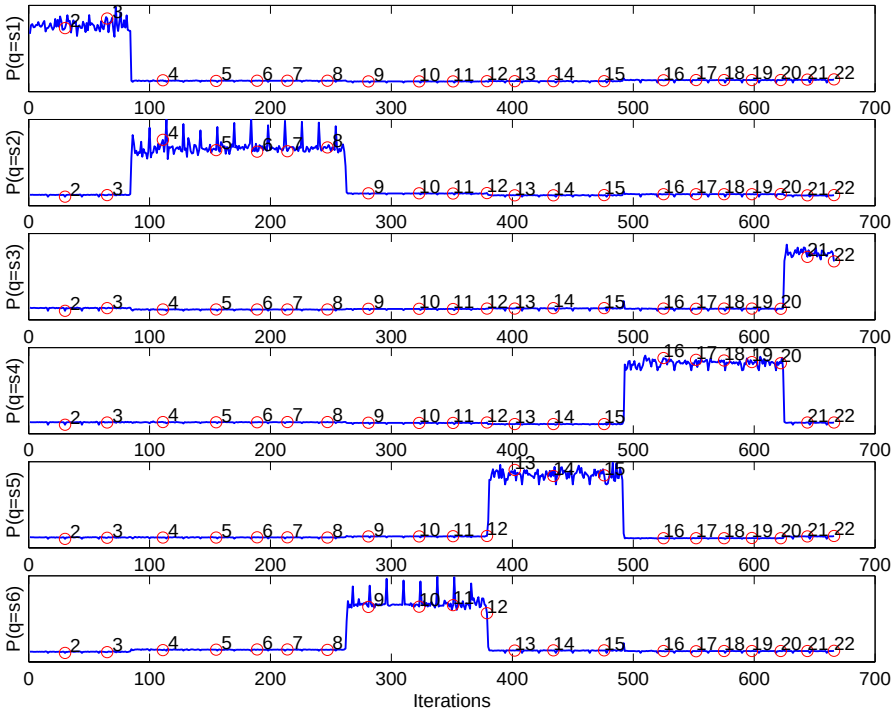


Figure 3.9: Experiment 1-a): Log of the localization probability in each state with $T = 14$ iterations. The time interval between iterations is 0.125s. Observation variance noise σ^2

equal to 14, 10, 15 and 22, respectively. They occur at the beginning of each interval T as consequence of the initialization referred in (3.44), without any connection with the previous α_{kT}^{kT} value. The smoothing of these peaks is achieved by normalizing the parameter $\alpha_{kT+1}^{kT+T}(i)$ in (3.44) and replacing it by

$$\alpha_{kT+1}^{kT+T}(i) \quad = \quad \frac{\alpha_{kT}^{kT}(i)}{\sum_{j=1}^{N} \alpha_{kT}^{kT}(j)}. \tag{3.53}$$

Figure 3.13 displays the results obtained as in Figure 3.9 but after the referred normalization. It is clear that the peaky evolution of the probability decreased, this resulting from the implemented normalization.

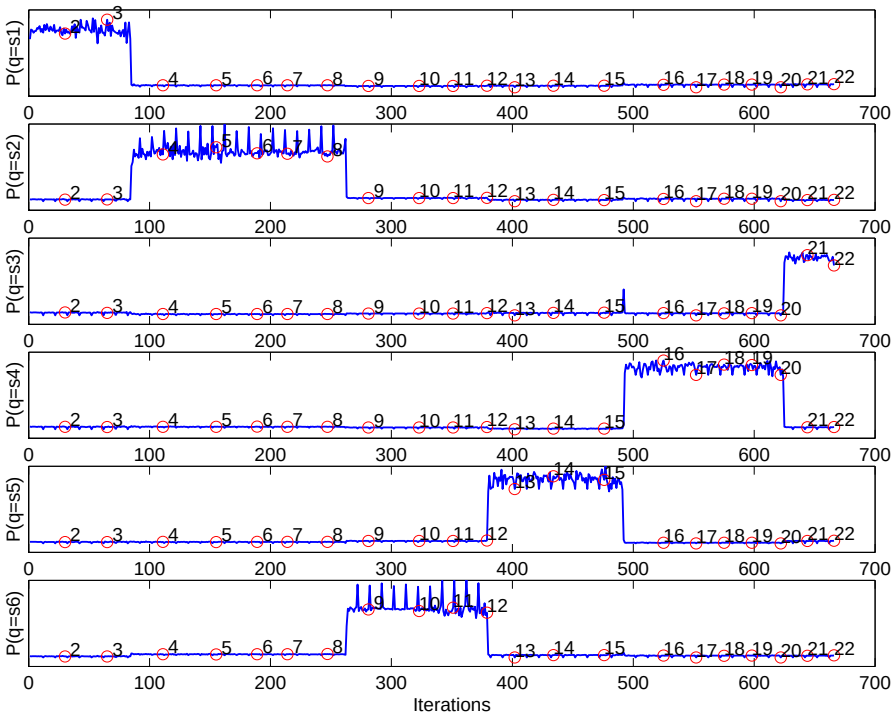


Figure 3.10: Experiment 1-b): Log of the localization probability in each state with $T = 10$ iterations. The time interval between iterations is 0.125s. Observation variance noise σ^2

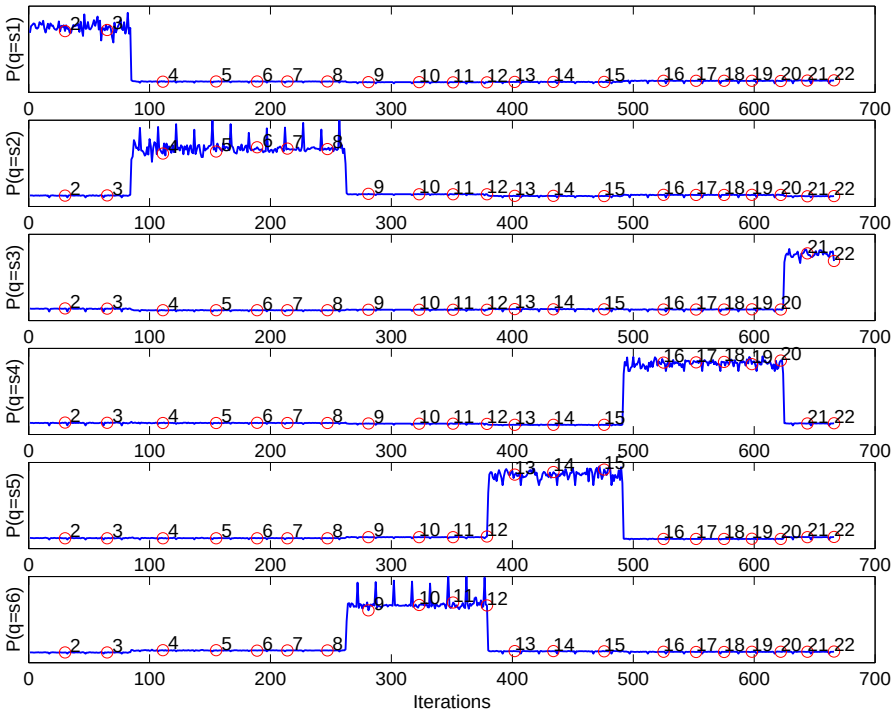


Figure 3.11: Experiment 1-c): Log of the localization probability in each state with $T = 15$ iterations. The time interval between iterations is 0.125s. Observation variance noise σ^2

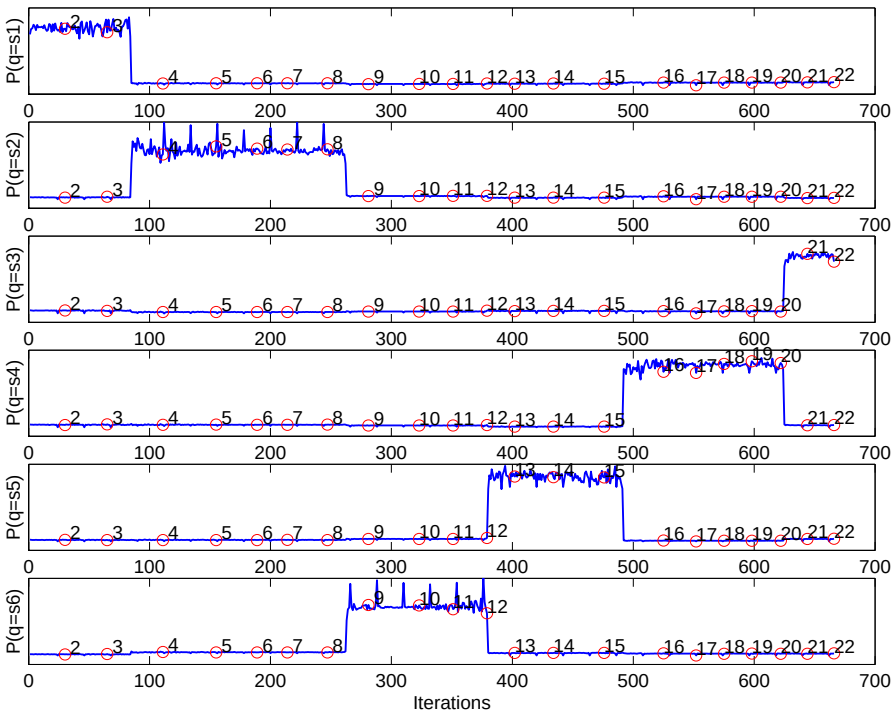


Figure 3.12: Experiment 1-d): Log of the localization probability in each state with $T = 22$ iterations. The time interval between iterations is 0.125s. Observation variance noise σ^2

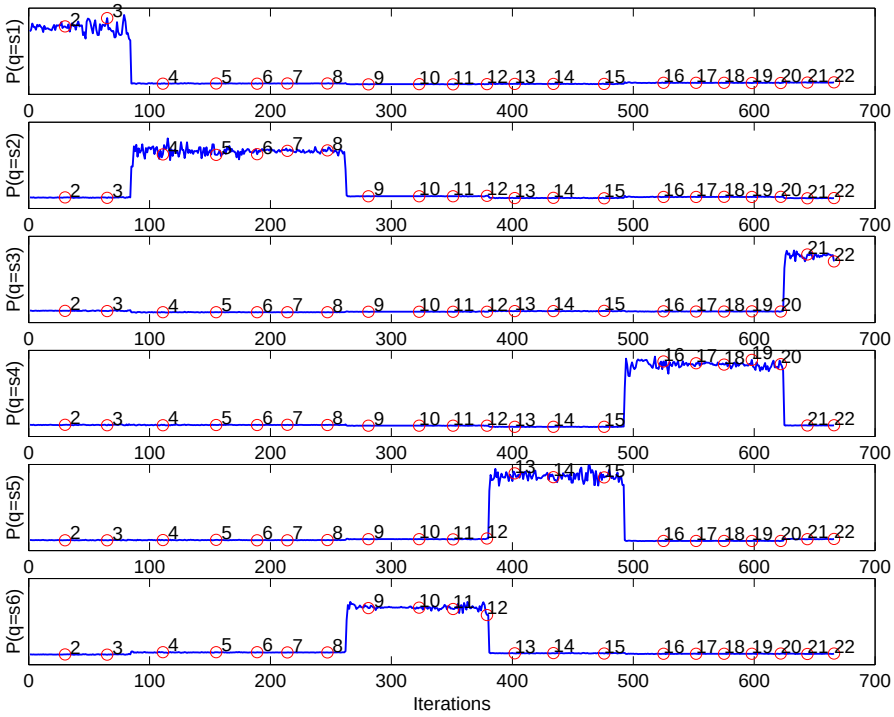


Figure 3.13: Experiment 2-a): Log of the localization probability in each state with $T = 14$ and normalization in α. Observation variance noise σ^2

To test the localization algorithm robustness, the standard deviation of the sensor noise was increased. The results in Figures 3.14-3.16 were obtained with the normalization in (3.53) and an observation standard deviation of 2σ, 4σ and 8σ, respectively, where σ refers to the value used in Figures 3.9 - 3.13.

It is clear the performance degradation in the localization estimate as the observation noise increases. In particular, in Figure 3.16, the localization result oscillates when the mobile robot travels between the states s_1 and s_2. The localization results show some activity in states s_4, s_5 and s_6, between iterations 0 and 200, which reflects that the high uncertainty presented in the observations leads to an increasing probability of the robot to localize in states with similar attributes. When the states are represented by attributes without metric information and the observation noise is high, the localization results present some uncertainty between states that are physical distant. This is a result of topological maps. In these situations, the navigation algorithm explained in Chapter 4 has to re-estimate a new trajectory from the current state to the main goal. Given the uncertainty, the current location estimate oscillation imposes the navigation to compute a new sequence of states to the goal. The map improvement, identifying or updating new states, can also reduce the ambiguities, even with high uncertainty in the observations, as demonstrated in Chapter 5.

In Experiment 4 the path planning presented in Figure 3.20 provides ambiguities. In some parts of the trajectory, the mobile robot is equally distant from two or more states. For instance, in Figure 3.21 the ambiguity situation among the states s_1, s_2 and s_3 is evident between the via points P_3 and P_4 (between the iterations 100 and 200). Also between P_4 and P_6 the ambiguity between s_2 and s_3 remains (iterations 200 to 300). After crossing the via point P_{12} the mobile robot becomes equidistant to the states s_3 and s_4 this creating, due to sensor noise, an ambiguity situation evident in Figure 3.21 (iterations 800 to 950). As illustrated in Figure 3.16, the localization algorithm may return oscillations (uncertainty) between two or more states. In Figure 3.16, the oscillations were caused by the uncertainty of the observation, whereas in Figure 3.21 the oscillations occur due to the trajectory shape relative to the states. The robot follows the trajectory that was defined in the middle of states and, therefore, the robot observes simultaneously the attributes of some state.

The navigation has to deal with these situations of uncertainty in the localization, updating the sequence of states to reach the goal. In the second situation, shown in Figure 3.21, the oscillations occur when the trajectory moves throw near states (small physical distances, as s_2 and s_3, s_2 and s_6 and, s_3 and s_4). The navigation may return similar sequences, only changing the first states of the sequence, as explained in Chapter 4.

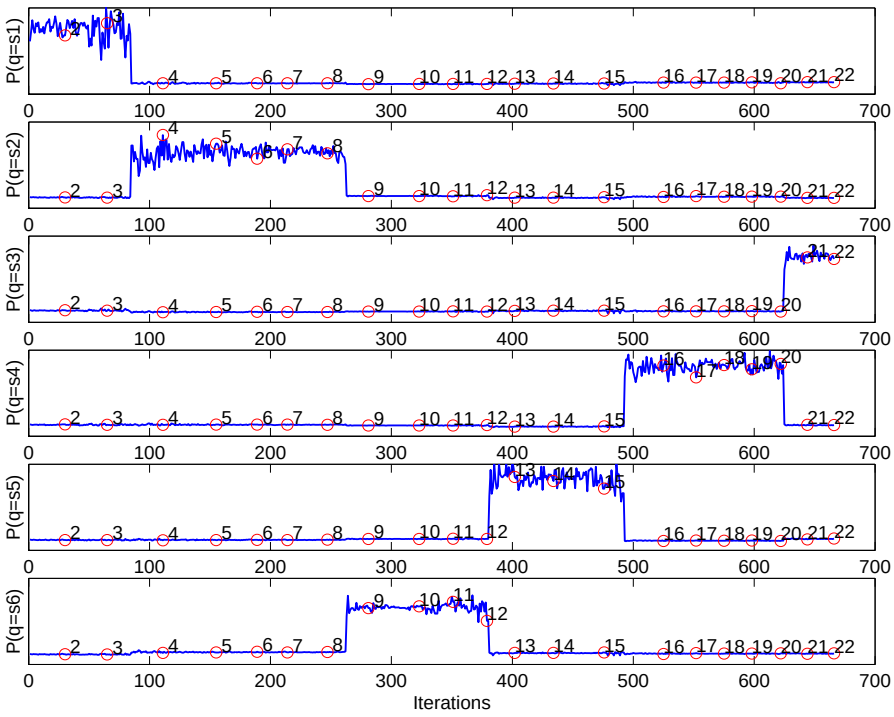


Figure 3.14: Experiment 2-b): Log of the localization probability evolution with normalization in α. Observation variance is $4\sigma^2$

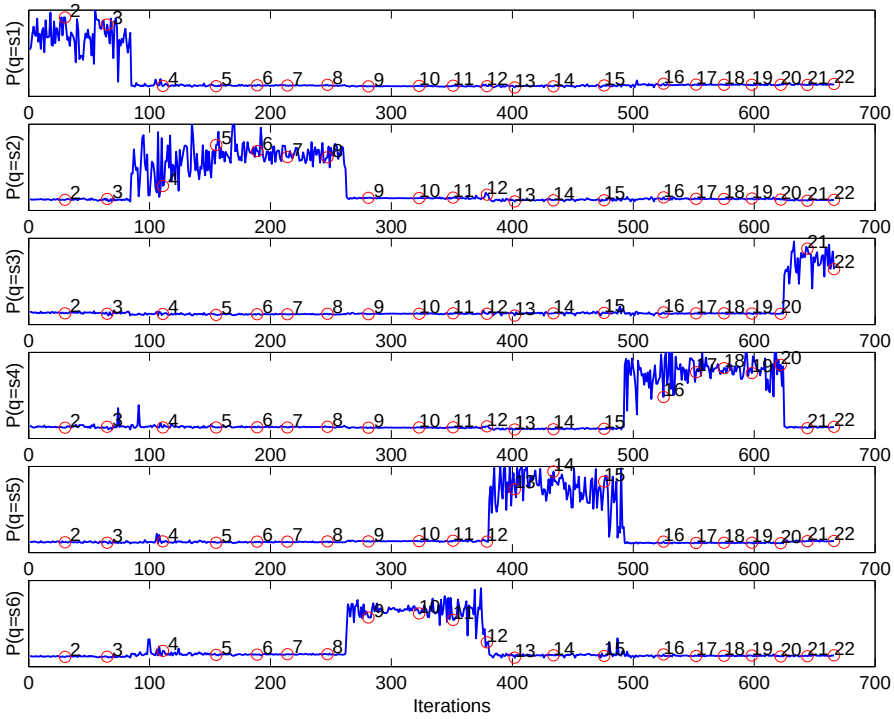


Figure 3.15: Experiment 2-c): Log of the localization probability evolution with normalization in α. Observation variance is $16\sigma^2$

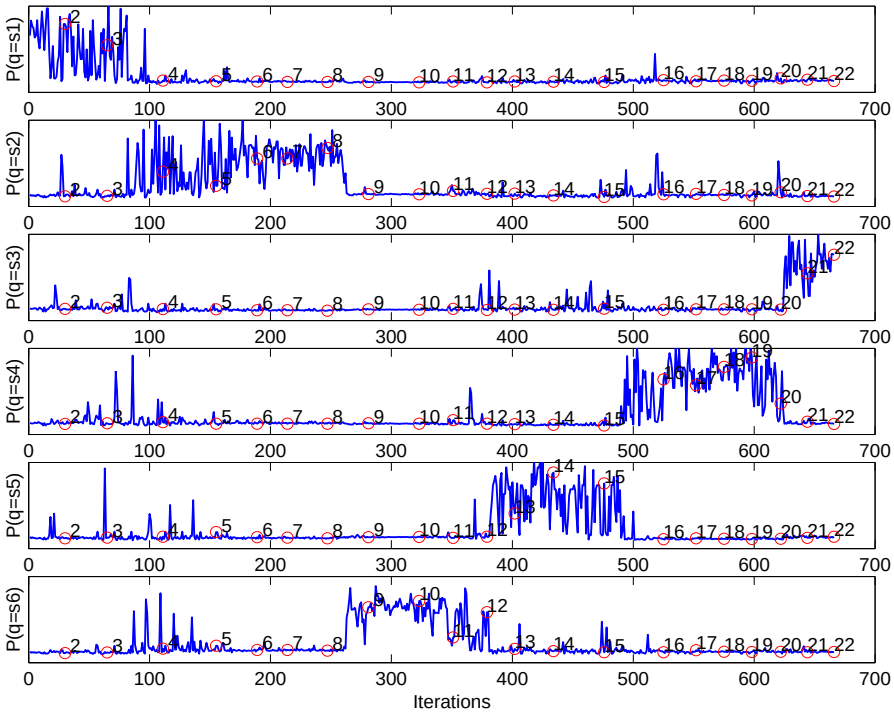


Figure 3.16: Experiment 2-d): Log of the localization probability evolution with normalization in α. Observation variance is $64\sigma^2$

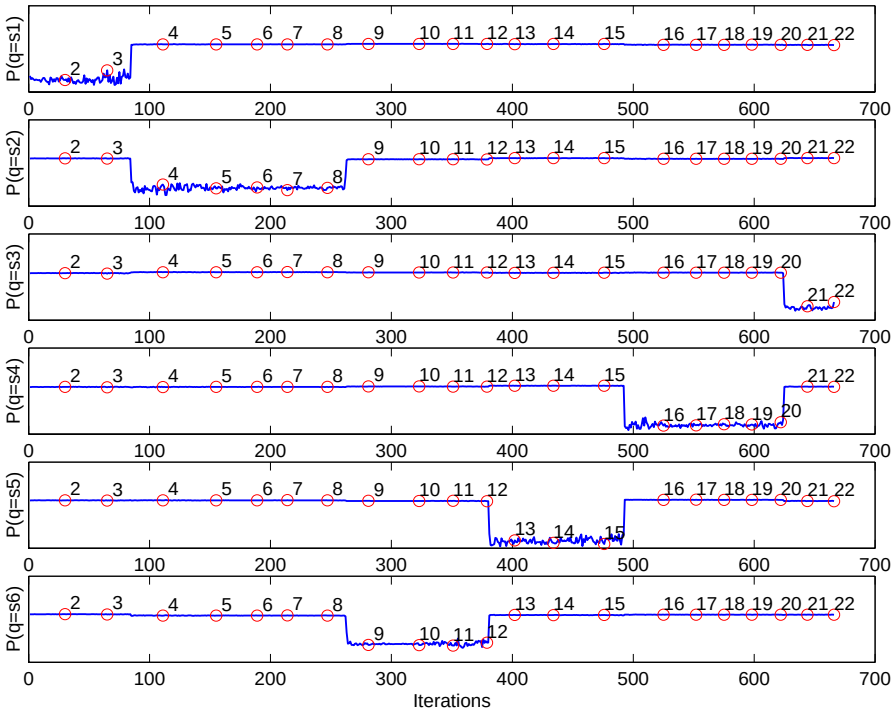


Figure 3.17: The localization probability evolution $P \cdot \log(P(q_t = s_i \mid O_t))$

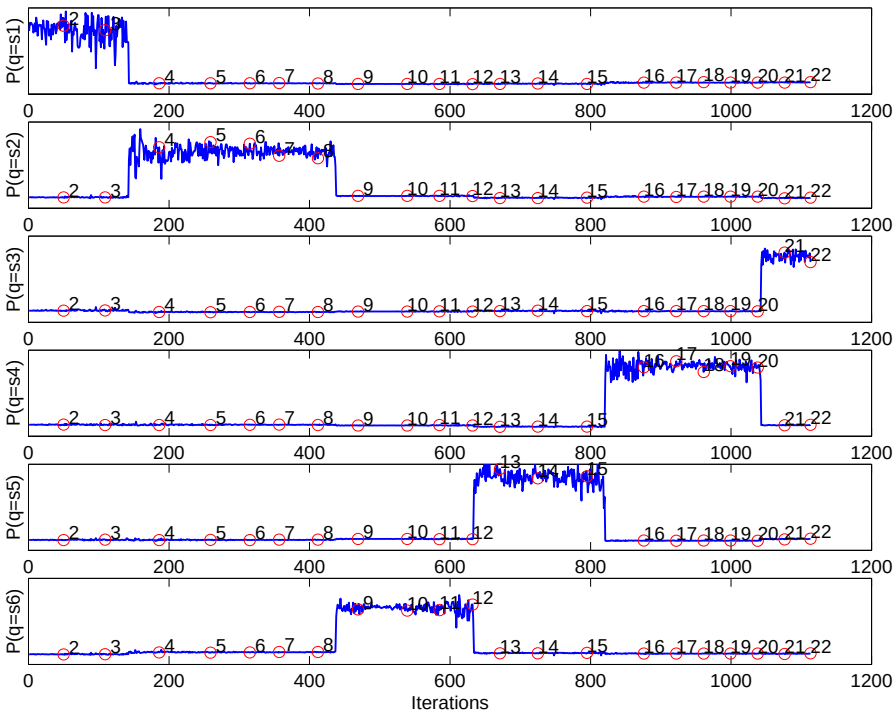


Figure 3.18: Experiment 3-a): Log of the localization probability in each state. The time interval between iterations is 0.075s

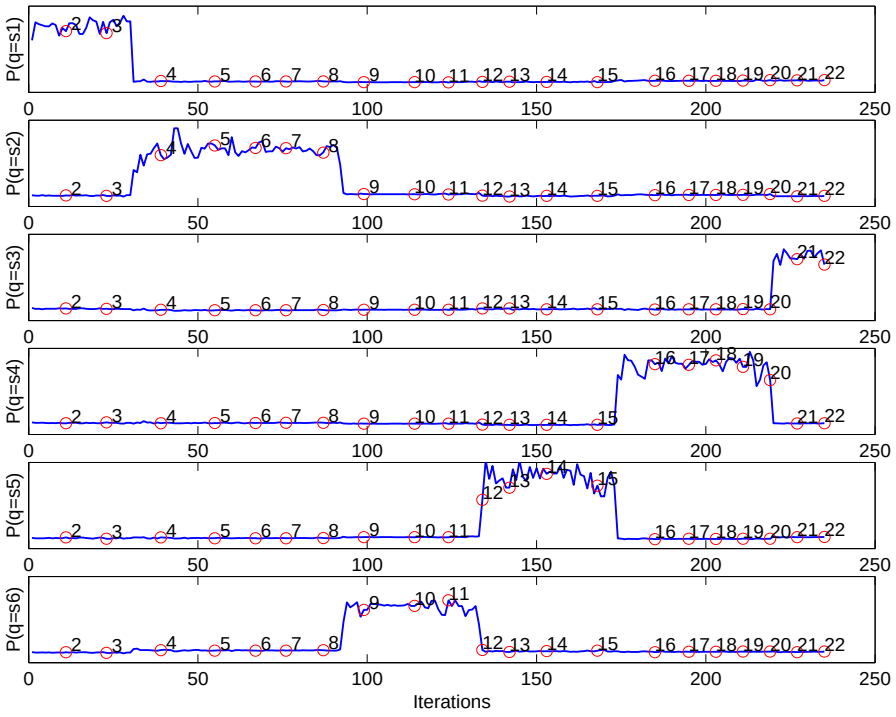


Figure 3.19: Experiment 3-b): Log of the localization probability in each state. The time interval between iterations is 0.350s

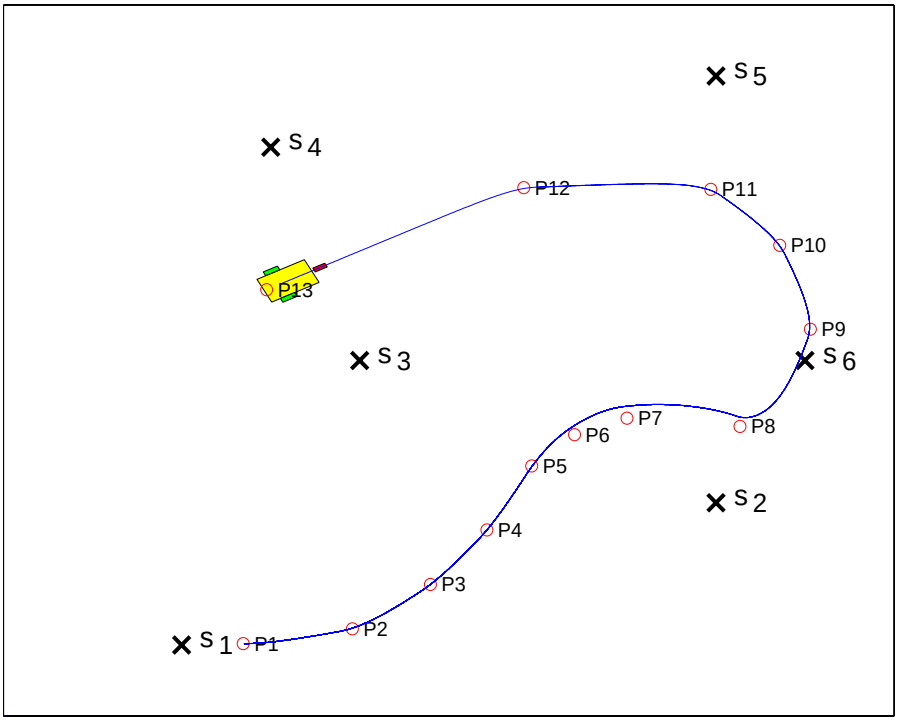


Figure 3.20: Experiment 4: Map states, via points and executed path

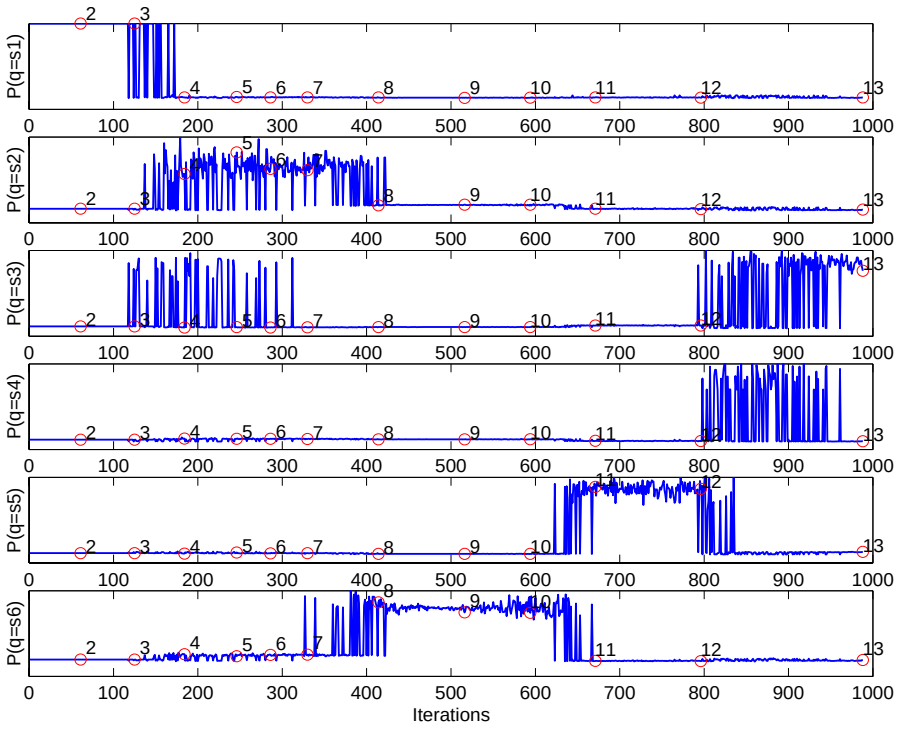


Figure 3.21: Experiment 4-a): Log of the localization probability evolution

Chapter 4

Navigation

As described in Chapter 2, a topological map is a representation of an environment with no metric information available, showing physical features that characterize particular locations or places that constitute the nodes of the map and that, in the context of this work, are named as states. The navigation procedure proposed in this work is also developed using a topological approach and is based on the robot location at each time instant and on the topological map, both assumed to be known for the navigation purposes. When navigation is herein cited, it only refers to a topological path planning, while the reference to the general problem of mobile robot navigation includes the three main issues of, localization, navigation and mapping.

The robot estimated location is equal to the map's state in its closest vicinity using a probabilistic approach to decide on this proximity function. The navigation procedure consists on finding the best way to reach a goal, a state in the topological map, given the current robot's state.

To reach the target state, the robot moves through other states, facing the uncertainty on the world perception and on the motion commands. The localization algorithm, described in the Chapter 3, deals with that uncertainty by a probabilistic approach to estimate the robot's location on the map. The navigation has also to deal with uncertainty and, consequently, a topological and probabilistic approach is developed to support it.

Given the localization result at each time instant, the navigation algorithm provides the best sequence of states from the current state to the goal's state. The robot has to follow these sequence of states, where the sequence's length or the states that defines the sequence could change along time. However, if the robot fails the sequence, i.e., if the robot reaches a state not included in the sequence, the topological navigation has to compute a new sequence, starting at the actual robot's state to the goal. This possible failure could be caused by the localization (as described in previous chapter, for instance, caused by the uncertainty in the observations) and by dynamical obstacles forcing a new trajectory. The navigation algorithm has to be robust to deal with these situations, updating the previous sequence to a new sequence of states.

The topological navigation is similar to the navigation adopted by human beings, that

try to be anchored to the well known features of the environment. When we are walking on the street and someone ask us how to reach a specific place, we describe a path as a sequence of places where the person has to walk through. Furthermore, it is important to describe how to obtain the best sequence of states given a map.

As referred in Section 2.1, the three main problems of mobile robot navigation are the localization, the navigation and the mapping. Similar to the localization, the navigation is developed assuming that the topological map is already known. Chapter 5 describes the map building algorithm, which runs permanently, but with a low frequency when compared to the localization or to the navigation components. However, for each time instant when the map algorithm updates the current topological map, the localization has to retrieve the new robot's state. Therefore, for the new robot's state and an updated topological map, the navigation has to compute a new sequence of states to reach the goal.

4.1 Problem Statement

Given the robot's state at time instant t, $q_t = s_i$, retrieved by the localization algorithm, (3.2) in Chapter 3, and assuming that the topological map is known, the navigation determines the sequence of states from q_t to the goal state, s_j, so that the objective function

$$P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) \tag{4.1}$$

is maximized. This corresponds to place the robot in a state s_j, after a period of time Δ, which is equivalent to have $q_{t+\Delta} = s_j$.

Each state of the topological map is associated to a combination of features and, when the robot is in the proximity of this state, the retrieved observations should be similar to those features. So, each observation should be associated to a particular state. The key-question in solving (4.1) is what the robot should observe since the current time instant t up to $t + \Delta$, when the final state s_j is reached, or equivalently,

$$\max_{o_{t+1},\ldots,o_{t+\Delta}} P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i). \tag{4.2}$$

If it is important to reach the goal in the minimum time interval, (4.2) has to be minimized relative to Δ, yielding the objective function,

$$\min_{\Delta} \left\{ \max_{o_{t+1},\ldots,o_{t+\Delta}} P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) \right\}. \tag{4.3}$$

However, at this point, the most important issue is how to determine a sequence of states to reach the goal, regardless of the time restriction. Consequently, in the approach

taken in the thesis, the minimization on the Δ parameter is disregarded and the topological navigation resumes to solve (4.2).

To solve (4.2) it is necessary to develop some research in (4.1), since the pdf is not known. Applying the Bayesian rules on (4.1), leads to:

$$\begin{aligned} P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) = \\ &= \frac{P(q_{t+\Delta} = s_j, q_t = s_i, o_{t+1}, \ldots, o_{t+\Delta})}{P(o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i)} \\ &= \frac{P(q_{t+\Delta} = s_j, o_{t+1}, \ldots, o_{t+\Delta} \mid q_t = s_i) \cdot P(q_t = s_i)}{P(o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i)} \\ &= \frac{P(q_{t+\Delta} = s_j, o_{t+1}, \ldots, o_{t+\Delta} \mid q_t = s_i)}{P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_t = s_i)}. \end{aligned} \tag{4.4}$$

The probability $P(q_{t+\Delta} = s_j, o_{t+1}, \ldots, o_{t+\Delta} \mid q_t = s_i)$ on the numerator of (4.4) can be decomposed in two other probabilities as derived in the sequel:

$$\begin{aligned} P(q_{t+\Delta} = s_j, o_{t+1}, \ldots, o_{t+\Delta} \mid q_t = s_i) = \\ &= \frac{P(q_{t+\Delta} = s_j, o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i)}{P(q_t = s_i)} \\ &= \frac{P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j, q_t = s_i)}{P(q_t = s_i)} \\ &= P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i). \end{aligned} \tag{4.5}$$

The two probabilities in (4.5) have different meanings. The first corresponds to the probability of acquiring a sequence of observations given the initial and final states restrictions, while the second represents the transition probability between the initial and the goal states.

Replacing (4.5) in (4.4) yields,

$$\begin{aligned} P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) = \\ &= \frac{P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i)}{P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_t = s_i)}, \end{aligned} \tag{4.6}$$

whose the denominator can also be written as,

$$\begin{aligned} P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_t = s_i) = \\ &= \sum_{n=1}^{N} \left[P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_n, q_t = s_i) \cdot P(q_{t+\Delta} = s_n \mid q_t = s_i) \right]. \end{aligned} \tag{4.7}$$

This decomposition is similar to the one presented in (4.5), where the target goal assumes all the possible states of the map. Replacing (4.7) in (4.6) yields,

$$
\begin{aligned}
&P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) = \\
&\quad = \frac{P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i)}{\sum_{n=1}^{N} \left[P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_n, q_t = s_i) \cdot P(q_{t+\Delta} = s_n \mid q_t = s_i)\right]} \\
&\quad = \eta \cdot P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i),
\end{aligned} \tag{4.8}
$$

whose the denominator corresponds to the all possible final states available in the map. As it does not influence the maximization, it is consequently assumed as a normalization factor, η.

As suggested in [61], (4.8) is evaluated by

$$
\begin{aligned}
&P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) = \\
&= \eta \cdot \sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, q_{t+1} = s_{l_1}, q_t = s_i) \cdot \\
&\quad \cdot P(q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i).
\end{aligned} \tag{4.9}
$$

The decomposition of (4.8) in (4.9) consists on $\Delta - 1$ sums corresponding to all the possible combinations of states between q_{t+1}, the current state, and $q_{t+(\Delta-1)}$. Based on the Markov assumption referred in Chapter 3, in the first term of (4.9), the observations are independent of $q_t = s_i$, yielding

$$
P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, q_{t+1} = s_{l_1}). \tag{4.10}
$$

The second term can be decomposed in two other probabilities,

$$
\begin{aligned}
&P(q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1} \mid q_{t+\Delta} = s_j, q_t = s_i) = \\
&\quad = \frac{P(q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}, q_{t+\Delta} = s_j, q_t = s_i)}{P(q_{t+\Delta} = s_j, q_t = s_i)} \\
&\quad = \frac{P(q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}, q_t = s_i)}{P(q_{t+\Delta} = s_j, q_t = s_i)} \\
&\quad = \frac{P(q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1} \mid q_t = s_i) \cdot P(q_t = s_i)}{P(q_{t+\Delta} = s_j, q_t = s_i)} \\
&\quad = \frac{P(q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1} \mid q_t = s_i)}{P(q_{t+\Delta} = s_j \mid q_t = s_i)}.
\end{aligned} \tag{4.11}
$$

Using (4.10) and (4.11) in (4.9) yields,

$$
\begin{aligned}
&P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) = \\
&\quad = \eta \cdot \sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1})
\end{aligned}
$$

$$\cdot P(q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1} \mid q_t = s_i). \tag{4.12}$$

The second term of (4.12) corresponds to the transition probability, or the probability of the robot crossing a given combination of states, starting in $q_t = s_i$, the initial condition for the navigation problem. At this point, it is important to identify in (4.12) the known parameters, as the transitions probabilities, a_{ij} and the observation pdfs, $b_i(o_t)$. This entire expression in (4.12) can also be expanded to

$$\begin{aligned}
&P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i) = \\
&= \sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}) \\
&\quad \cdot P(q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1} \mid q_t = s_i) \\
&= \sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}) \\
&\quad \cdot P(q_{t+\Delta} = s_j \mid q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}, q_t = s_i) \cdot \\
&\quad \cdot P(q_{t+(\Delta-1)} = s_{l_{\Delta-1}} \mid q_{t+(\Delta-2)} = s_{l_{\Delta-2}}, \ldots, q_{t+1} = s_{l_1}, q_t = s_i) \cdot \\
&\quad \cdots \\
&\quad \cdot P(q_{t+2} = s_{l_2} \mid q_{t+1} = s_{l_1}, q_t = s_i) \cdot \\
&\quad \cdot P(q_{t+1} = s_{l_1} \mid q_t = s_i) \\
&= \sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}) \\
&\quad \cdot P(q_{t+\Delta} = s_j \mid q_{t+(\Delta-1)} = s_{l_{\Delta-1}}) \cdot \\
&\quad \cdot P(q_{t+(\Delta-1)} = s_{l_{\Delta-1}} \mid q_{t+(\Delta-2)} = s_{l_{\Delta-2}}) \cdot \\
&\quad \cdots \\
&\quad \cdot P(q_{t+2} = s_{l_2} \mid q_{t+1} = s_{l_1}) \cdot \\
&\quad \cdot P(q_{t+1} = s_{l_1} \mid q_t = s_i).
\end{aligned} \tag{4.13}$$

Replacing the transition probabilities in (4.13) by the variable a_{ij} defined in (3.3) in Section 3.2, (4.13) results in (4.14),

$$\begin{aligned}
&P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i) = \\
&= \sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}) \\
&\quad \cdot a_{l_{\Delta-1}j} \cdot a_{l_{\Delta-2}l_{\Delta-1}} \cdot \ldots \cdot a_{l_1 l_2} \cdot a_{i l_1} \\
&= \sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1} \mid o_{t+2}, \ldots, o_{t+\Delta}, q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}) \\
&\quad \cdot P(o_{t+2}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1})
\end{aligned}$$

$$
\begin{aligned}
&\cdot a_{l_{\Delta-1}j} \cdot a_{l_{\Delta-2}l_{\Delta-1}} \cdot \ldots \cdot a_{l_1 l_2} \cdot a_{i l_1} \\
= &\sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1} \mid o_{t+2}, \ldots, o_{t+\Delta}, q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}) \\
&\cdot P(o_{t+2} \mid o_{t+3}, \ldots, q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}, q_t = s_i) \cdot \\
&\cdot P(o_{t+3} \mid o_{t+4}, \ldots, q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}, q_t = s_i) \cdot \\
&\ldots \\
&\cdot P(o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_{t+(\Delta-1)} = s_{l_{\Delta-1}}, \ldots, q_{t+1} = s_{l_1}, q_t = s_i) \cdot \\
&\cdot a_{l_{\Delta-1}j} \cdot a_{l_{\Delta-2}l_{\Delta-1}} \cdot \ldots \cdot a_{l_1 l_2} \cdot a_{i l_1} \\
= &\sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} P(o_{t+1} \mid q_{t+1} = s_{l_1}) \cdot \\
&\cdot P(o_{t+2} \mid q_{t+2} = s_{l_2}) \cdot \\
&\cdot P(o_{t+3} \mid q_{t+3} = s_{l_3}) \cdot \\
&\ldots \\
&\cdot P(o_{t+\Delta} \mid q_{t+\Delta} = s_j) \cdot \\
&\cdot a_{l_{\Delta-1}j} \cdot a_{l_{\Delta-2}l_{\Delta-1}} \cdot \ldots \cdot a_{l_1 l_2} \cdot a_{i l_1}.
\end{aligned} \tag{4.14}
$$

The first probability in (4.12), developed in (4.14), presents the observation probability that can be also replaced by a variable defined in Section 3.2, $b_i(o_t)$, yielding

$$
\begin{aligned}
&P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i) = \\
&= \sum_{l_{\Delta-1}=1}^{N} \cdots \sum_{l_1=1}^{N} b_{l_1}(o_{t+1}) \cdot b_{l_2}(o_{t+2}) \cdot \ldots \cdot b_{l_{\Delta-1}}(o_{t+(\Delta-1)}) \cdot b_j(o_{t+\Delta}) \cdot \\
&\quad \cdot a_{l_{\Delta-1}j} \cdot a_{l_{\Delta-2}l_{\Delta-1}} \cdot \ldots \cdot a_{l_1 l_2} \cdot a_{i l_1} \\
&= \sum_{l_1=1}^{N} a_{i l_1} b_{l_1}(o_{t+1}) \left[\sum_{l_2=1}^{N} a_{l_1 l_2} b_{l_2}(o_{t+2}) \ldots \left[\sum_{l_{\Delta-2}=1}^{N} a_{l_{\Delta-3}l_{\Delta-2}} b_{l_{\Delta-2}}(o_{t+(\Delta-2)}) \right.\right. \\
&\quad \left.\left. \left[\sum_{l_{\Delta-1}=1}^{N} a_{l_{\Delta-2}l_{\Delta-1}} b_{l_{\Delta-1}}(o_{t+(\Delta-1)}) \left[a_{l_{\Delta-1}j} b_j(o_{t+\Delta}) \right] \right] \right] \right].
\end{aligned} \tag{4.15}
$$

It is important to remember that the main problem of navigation consists on finding the best topological path (sequence of states) from the current location to the state goal, or, equivalently, to solve (4.2). The work developed through this section developed a way to obtain a solution, computationally feasible, to determine the desired sequence of states.

To simplify (4.15), define $\beta_{t+\Delta}^{(j)}(l_\Delta)$ as,

$$
\beta_{t+\Delta}^{(j)}(l_\Delta) = \begin{cases} 1 & , \quad l_\Delta = j \\ 0 & , \quad l_\Delta \neq j \end{cases}, \tag{4.16}
$$

where s_j is the goal state. Therefore, the parameters $a_{l_{\Delta-1}j} b_j(o_{t+\Delta})$ in (4.15) become,

$$a_{l_{\Delta-1}j} b_j(o_{t+\Delta}) = \sum_{l_\Delta=1}^{N} a_{l_{\Delta-1}l_\Delta} b_{l_\Delta}(o_{t+\Delta}) \beta_{t+\Delta}^{(j)}(l_\Delta). \tag{4.17}$$

Replacing (4.17) in (4.15), the navigation probability (4.15) is written as:

$$\begin{aligned} &P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i) = \\ &= \sum_{l_1=1}^{N} a_{il_1} b_{l_1}(o_{t+1}) \left[\sum_{l_2=1}^{N} a_{l_1 l_2} b_{l_2}(o_{t+2}) \ldots \left[\sum_{l_{\Delta-2}=1}^{N} a_{l_{\Delta-3} l_{\Delta-2}} b_{l_{\Delta-2}}(o_{t+(\Delta-2)}) \right.\right. \\ &\quad \left.\left. \left[\sum_{l_{\Delta-1}=1}^{N} a_{l_{\Delta-2} l_{\Delta-1}} b_{l_{\Delta-1}}(o_{t+(\Delta-1)}) \left[\sum_{l_\Delta=1}^{N} a_{l_{\Delta-1} l_\Delta} b_{l_\Delta}(o_{t+\Delta}) \beta_{t+\Delta}^{(j)}(l_\Delta) \right] \right] \right] \right]. \end{aligned} \tag{4.18}$$

Replacing (4.16) in (4.18), leads to

$$\begin{aligned} &P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i) = \\ &= \sum_{l_1=1}^{N} a_{il_1} b_{l_1}(o_{t+1}) \left[\sum_{l_2=1}^{N} a_{l_1 l_2} b_{l_2}(o_{t+2}) \ldots \left[\sum_{l_{\Delta-2}=1}^{N} a_{l_{\Delta-3} l_{\Delta-2}} b_{l_{\Delta-2}}(o_{t+(\Delta-2)}) \right.\right. \\ &\quad \left.\left. \left[\sum_{l_{\Delta-1}=1}^{N} a_{l_{\Delta-2} l_{\Delta-1}} b_{l_{\Delta-1}}(o_{t+(\Delta-1)}) \left[\beta_{t+(\Delta-1)}^{(j)}(l_{\Delta-1}) \right] \right] \right] \right] \\ &= \sum_{l_1=1}^{N} a_{il_1} b_{l_1}(o_{t+1}) \left[\sum_{l_2=1}^{N} a_{l_1 l_2} b_{l_2}(o_{t+2}) \ldots \right. \\ &\quad \left. \ldots \left[\sum_{l_{\Delta-2}=1}^{N} a_{l_{\Delta-3} l_{\Delta-2}} b_{l_{\Delta-2}}(o_{t+(\Delta-2)}) \beta_{t+(\Delta-2)}^{(j)}(l_\Delta - 1) \right] \right] \\ &= \ldots \\ &= \sum_{l_1=1}^{N} a_{il_1} b_{l_1}(o_{t+1}) \beta_{t+1}^{(j)}(l_1) \\ &= \beta_t^{(j)}(i), \end{aligned} \tag{4.19}$$

where $\beta_t^{(j)}(i)$ allows a recursive evaluation, as in the common Forward-Backward algorithm, i.e., similar to the algorithm referred in the Chapter 3, the parameter β in (3.26) address the future information relative to a given time instant t.

$$\beta_\tau^{(n)}(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{\tau+1}) \cdot \beta_{\tau+1}^{(n)}(j), \qquad t \leq \tau < t + \Delta, \tag{4.20}$$

with the restriction that, at time instant $t + \Delta$, the robot has to be placed in the goal

state, s_j.

Returning to the initial probability, the navigation problem is stated in (4.2), where the probability can be rewritten as

$$P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) =$$

$$= \frac{P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_j, q_t = s_i) \cdot P(q_{t+\Delta} = s_j \mid q_t = s_i)}{\sum_{n=1}^{N} \left[P(o_{t+1}, \ldots, o_{t+\Delta} \mid q_{t+\Delta} = s_n, q_t = s_i) \cdot P(q_{t+\Delta} = s_n \mid q_t = s_i)\right]}. \quad (4.21)$$

The equality (4.21) is simplified to (4.22), where the numerator is equal to (4.19) and, by inspection, the denominator equals $\sum_{n=1}^{N} \beta_t^{(n)}(i)$, leading to

$$\frac{\beta_t^{(j)}(i)}{\sum_{n=1}^{N} \beta_t^{(n)}(i)}. \quad (4.22)$$

In summary, the topological navigation consists on computing the best sequence of states starting in the current robot's position s_i, retrieved by the localization algorithm and ending into the goal state s_j. The sequence of states is calculated, finding the sequence of observations that maximize

$$\max_{o_{t+1}, \ldots, o_{t+\Delta}} \frac{\beta_t^{(j)}(i)}{\sum_{n=1}^{N} \beta_t^{(n)}(i)}, \quad (4.23)$$

with the restriction referred in (4.16) and where the parameter $\beta_t^{(j)}$ is evaluated recursively, as shown in (4.20). These equations, (4.16) and (4.20) are rewritten for simplicity,

$$\beta_{t+\Delta}^{(j)}(l_\Delta) = \begin{cases} 1 & , \quad l_\Delta = j \\ 0 & , \quad l_\Delta \neq j \end{cases}, \quad (4.24)$$

$$\beta_\tau^{(n)}(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{\tau+1}) \cdot \beta_{\tau+1}^{(n)}(j), \qquad t \leq \tau < t + \Delta. \quad (4.25)$$

The parameter $\beta_t^{(j)}$ addresses the future observations, or equivalently, what the robot should observe to be anchored on the states of the topological map to reach the target goal. The past information is all enclosed in the current state retrieved by the localization. Therefore, the parameter α, described in the previous chapter is not present in the topological navigation procedure.

However, the result from (4.23) is a sequence of observations. The question arises, on how to convert a sequence of observations in a sequence of states. The answer is explained in the following section.

4.1.1 Maximization Implementation

The navigation solution is the best sequence of observations $o_{t+1}, \ldots, o_{t+\Delta}$ that result from the maximization of (4.23). The sequence of observations has to be converted in a sequence of states, that the robot has to follow to reach the main goal.

The maximization in (4.23) returns the best sequence of observations, or equivalently, for a specific value of Δ, the navigation algorithm returns $O^*_\Delta = \{o^*_{t+1}, o^*_{t+2}, \ldots, o^*_{t+\Delta}\}$, with an uncertainty equal to $P(q_{t+\Delta} = s_j \mid o^*_{t+1}, o^*_{t+2}, \ldots, o^*_{t+\Delta}, q_t = s_i)$.

The observations are relative to the states, since these are identified by particular features. Each state is defined by a single or a set of multi-dimensional Gaussian pdfs, as explained in Section 2.2.4, each state s_i is mainly represented by two parameters, the mean vector μ_i and the covariance matrix R_i (for a set of Gaussians, it is also necessary a third parameter, the weight for each Gaussian). According to this assumption, each observation o^* is associated to the state with the nearest mean vector,

$$s^*_i = \arg \min_{i=1,\ldots,N} \|o^* - \mu_i\|. \tag{4.26}$$

For a sequence of observations

$$\{o^*_{t+1}, o^*_{t+2}, \ldots, o^*_{t+\Delta}\}$$

corresponds a sequence of states

$$\{s^*_{t+1}, s^*_{t+2}, \ldots, s^*_{t+\Delta}\},$$

where each state s_i is identified through (4.26). An association criteria between observations and states in (4.26) may also include the covariance matrices, as the Mahalanobis distance as described in [102].

Given the sequence of states that defines a topological path, it is also necessary to explain how the robot can follow that path. Therefore, it is necessary to establish and implement a methodology on the robot to follow that sequence of states. The adopted methodology requires the orientation between different states. One possible way to estimate the orientation between two different states is based on the results of the localization algorithm. The localization algorithm estimates the robot position on the topological map, or, the current state where the robot is, as explained in the previous chapter. The robot can stay in the same state with different poses, as illustrated in Figure 3.1. When the robot follows a direction θ_d, moving from state s_i to state s_j, it is defined that the orientation error is $\theta_d - \theta$. The orientation is referred to the North-South and West-East referential, the common information retrieved by a compass.

In Figure 4.1 the robot is located in state s_i, i.e., $q_t = s_i$, estimated by the topological localization algorithm, where the orientation between the two states, $\theta_{ij} = \theta_d$ is known. Assuming that the best sequence returned by the navigation algorithm to reach the goal

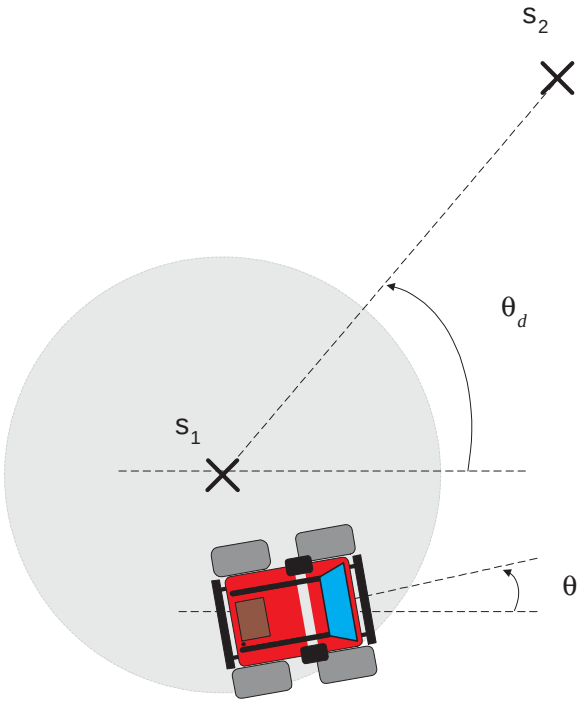


Figure 4.1: Orientation error: the robot is moving from the state s_1 to state s_2

state is $S^* = \{s_i^*, s_j^*\}$, the robot has to follow this sequence, changing its actual metric orientation θ to the θ_{ij}. This orientation adjustment is accomplished by controlling the velocities applied to the wheels, this corresponding to the connection between the high level of navigation (topological navigation) with a low level or motion control. Section 4.2 describes how to implement that low level of navigation, the motion control, including the solution to the undesirable situations created by the obstacles.

4.1.2 Computational Requirements

To evaluate the expression presented in (4.19) it is necessary $(2N)^{\Delta}$ multiplications and N^{Δ} sums. Since the multiplications require more computational time, the order of the problem is $O(2^{\Delta}N^{\Delta})$. For the maximization procedure described in (4.23), it is also necessary to evaluate the corresponding denominator. It is similar to repeat (4.19) N times. However, the navigation algorithm remains with the same complexity,

$$O(N \cdot (2N)^{\Delta}) \approx O((2N)^{\Delta}).$$

For the case where the maximization also concerns the dimension of Δ, where it assumes integers values $1, 2, \ldots, L$ as mentioned in (4.3), the navigation algorithms complexity is equal to $O(L \cdot (2N)^{\Delta})$ since Δ only assumes L different integer values imposed in the initialization of the algorithm.

4.2 Low Level Driving Methodology

The topological navigation algorithm returns a sequence of states, starting on the current state and ending into the target one. Since the robot is a mobile vehicle, it is necessary to convert the sequence of states into velocities to apply on the wheels to have the robot moving along this sequence of states. This is equivalent to change from topological navigation (high level of abstraction) to a metric navigation (low level of abstraction), as illustrated in Figure 4.2.

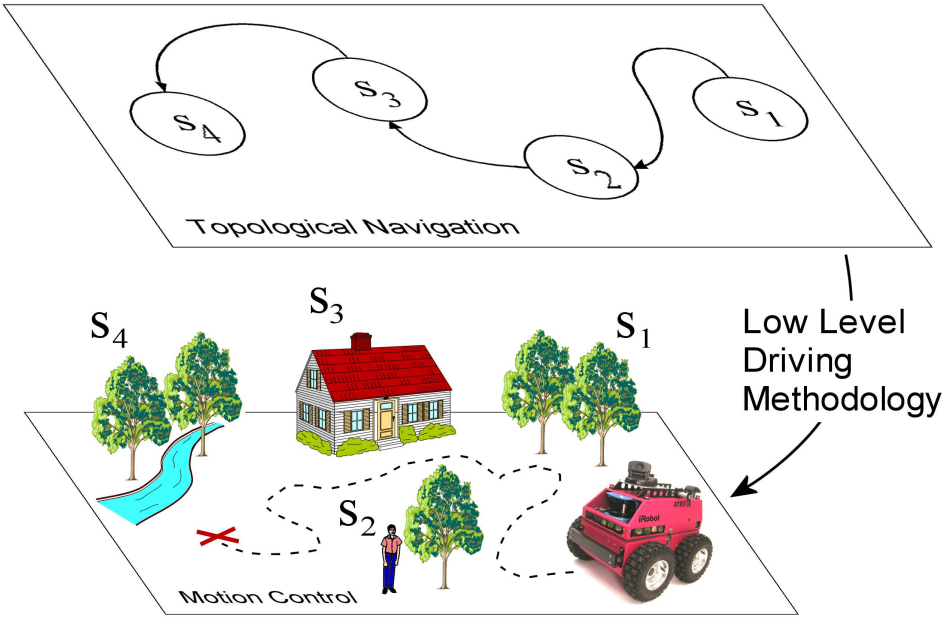


Figure 4.2: The connection between the topological navigation and the motion control

To accomplish this step it is necessary to know the states that compose the map, the angles between them and the geometric robot's orientation.

For the sake of clarity, it is important to underline that the map, the topological map, composed by a set of states, is computed by the algorithm described in Chapter 5. The direction between states is estimated as described in the previous section, by saving the orientation assumed by the robot between the transition of two consecutive states. For instance, the robot is in state s_i and following NW direction, it reaches the state s_j after a couple of iterations. So the direction between s_i and s_j, θ_{ij}, is NW. However this process of estimation is repeated several times and the direction between states is refined with the transitions between these two same states. The directions between states may have more precision (for instance, between -180º and 180º with 0.5º of precision) despite of few directions (the traditional ones: north, south, west, east, northwest, ...). Even when the robot is trying to follow that direction it can reach another state, not s_j. If the robot reaches a state not contained in the sequence, the navigation algorithm returns a new sequence given the new current state, as explained in Section 4.1.1.

To follow a given direction, the orientation goal to reach a state, θ_d, it is necessary to know the current robot's orientation, θ. The procedure to follow θ_d sis equivalent to follow a path defined by a set of landmarks, defined as landmark to landmark control, as explained in [55], where the orientation error is given by

$$\theta_e = \theta_d - \theta. \tag{4.27}$$

The estimation of the robot's orientation is possible using the robot's kinematics and sensors or by an appropriate sensor. The kinematics approach is similar to the pose estimation using only the odometric. After a few velocity commands it is impossible to estimate the robot's orientation using the kinematics, since the associated errors are cumulative. Using only an appropriate orientation sensor like a compass is a better choice, but not enough given the traditional time-delay of these type of sensors. The selected approach for orientation estimation is based on integrating both techniques through an Extended Kalman Filter (EKF) methodology [25, 45], as described in Appendix B.3.

Given the robot's orientation, θ, retrieved by an EKF, the next problem is to target the robot to the desired direction, θ_d, equal to θ_{ij},

$$\theta_d = \theta_{ij},$$

which corresponds to the direction between the current state s_i, yielded by the topological localization estimation, and the next goal state s_j, obtained from the topological navigation.

4.3 Behavior Approach

The topological navigation approach discussed through this chapter provides a sequence of states to reach a goal. The orientation estimated, referred in the previous section, gives the error between the current orientation of the robot and the direction between the current state and the next one in the sequence, as described in (4.27). It is not required any metric information on the position of the robot in a referential, but only on the orientation.

Given the sequence of states, it is necessary to develop an approach to follow this sequence, using the orientations between states, θ_{ij}. This is similar to a path following procedure implemented at a low level of navigation, providing the velocities for the motors. It is important to be aware that there are unexpected situations that could occur and that influence the performance of the planned path (sequence of states), as illustrated in Figure 4.3. It is expected that the transition probability between two states s_i and s_j is low if there are known, STATIC obstacles between them. Therefore, the sequence of states provided by the navigation algorithm is computed to avoid that s_i is immediately followed by s_j, given the low probability transition. However, temporary obstacles may appear, as people walking or moving objects.

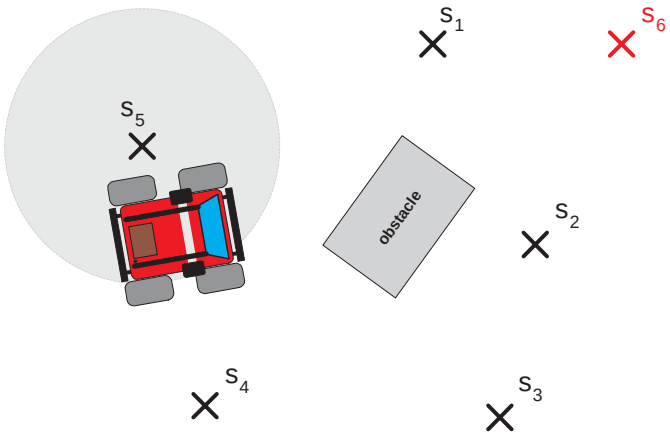


Figure 4.3: Unexpected obstacles along the trajectory, where the target state is s_6

Given the state goal and the unexpected situations it is necessary to develop a process that controls the robot to the target avoiding the undesirable obstacles. One common approach is using behaviors for autonomous robots, a component of an agent architecture extensively studied in autonomous agent research [14, 84, 105].

The selected approach is oriented to the use of attractive and repulsive behaviors [34], with the action leading the robot to the target while avoiding obstacles. Since the main goal is to avoid obstacles (people walking, cars moving, etc) the observations are retrieving from range sensors. Based on [21], the behavior approach used in the thesis is defined by combining two behaviors, whose weight influence is a function of the sensors observation, resulting on velocity commands. It is important to point out that the connection between the topological navigation (the sequence of states to follow) and a low level navigation (velocity control) is being accomplished exactly at this point.

4.3.1 Attractive Behavior

The selected behaviors are extracted from the work of Estela Bicho [21]. The robot is driven according to the speed of the wheels, which are converted into angular and translation velocities, Ω and V, as described in Appendix B.2. To follow the orientation to a target, it is most important to evaluate the angular velocity, Ω, leaving the translation speed, V, constant and non-zero. The translation speed can also be managed by other type of control not important at this point, since the orientations is the main issue.

The desired direction, θ_d, is given by the orientation between the current state, where the robot is and the next target state. The orientation error, θ_e, is the difference between θ_d and θ, the orientation of the robot that is time variant (4.27). An attractive behavior retrieves an angular velocity as a function of the orientation error. Based on [21], the attractive behavior is described as the odd function, $f_a(\theta_e)$,

$$f_a(\theta_e) = -\sin(k_0\theta_e), \tag{4.28}$$

where k_0 defines the slope of the function sin, as illustrated in Figure 4.4. The angular velocity generated only by the attractive behavior results from multiplying this function by a reference angular velocity,

$$\Omega = f_a(\theta_e) \cdot \Omega_{max}. \tag{4.29}$$

The point where the orientation error is equal to zero works as an attractive point with this behavior. When the orientation error is positive, the resulted angular velocity is negative leading θ_e to decrease to zero. When the orientation error is negative, the resulted angular velocity is positive and θ_e converges to zero. It is important to underline that when the direction error to the target is higher than 90º, the attractive behavior must be also high to drive the robot to the goal. When this occurs, Ω should be equal to Ω_{max}, the maximum angular velocity.

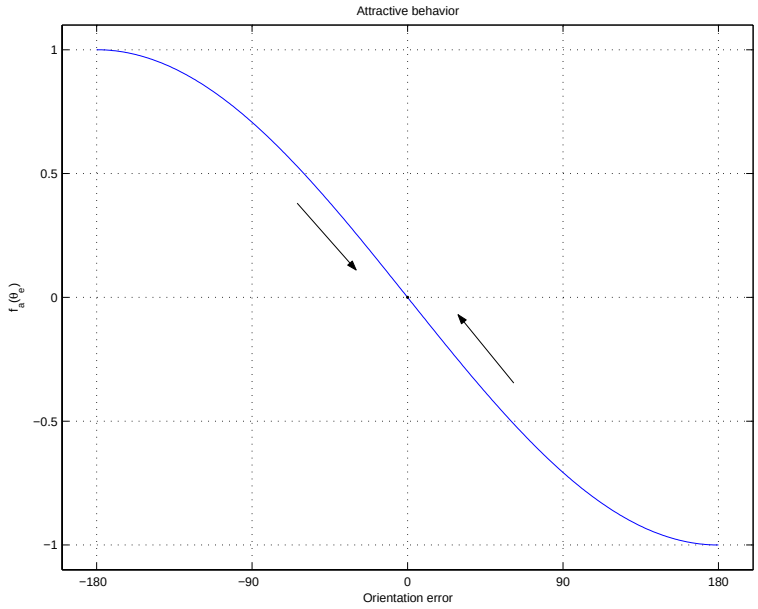


Figure 4.4: Attractive behavior

4.3.2 Repulsive Behavior - Obstacle Avoidance

To avoid unexpected or dynamic obstacles along the robot path a repulsive behavior, dependent on the obstacle's orientation, is implemented. To estimate the orientation of the nearest obstacle, θ_o (see Figure 4.5), the sensors available on the robot are used together with a selected approached described in [27]. The angle θ_o is given by the difference between the robot's orientation and the orientation of the nearest point of the obstacle to the robot.

A repulsive behavior retrieves an angular velocity that depends on θ_o, where the sign of Ω is equal to the sign of θ_o. In [21], a collision avoidance implementation is based on a repulsive behavior, described as the odd function, $f_r(\theta_o)$,

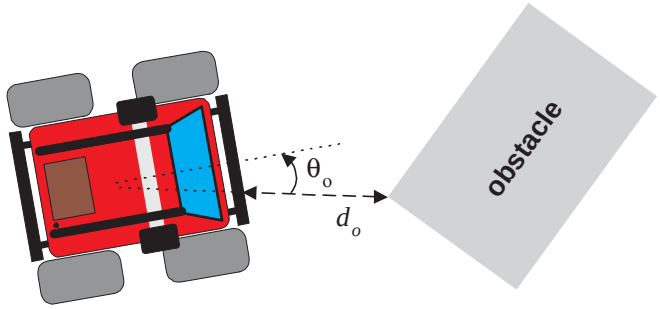


Figure 4.5: Unexpected obstacles along the trajectory

$$f_r(\theta_o) = k_1 \theta_o e^{-k_2 \theta_o^2}, \tag{4.30}$$

where k_1 corresponds to the normalization of f_r, i.e., the maximum value of $|f|$ is 1, which occurs when the derivative of f_r is equal to zero, or $k_1 = \sqrt{2k_2 \cdot e}$, as illustrated in Figure 4.6. The parameter k_2 is relative to the function slope, or the point where the maximum value of the derivative of f_r is reached. The maximum value of f_r is determined when the derivative of f_r is zero, or, when the orientation to the obstacle is equal to $\frac{\pm 1}{\sqrt{2k_2}}$. The angular velocity is determined similarly as in the attractive behavior, multiplying f_r by the maximum velocity, i.e.,

$$\Omega = f_r(\theta_o) \cdot \Omega_{max}. \tag{4.31}$$

In this behavior, the situation where the orientation to the obstacle is zero acts as a repulsive point. When the orientation is positive, the resulted angular velocity is also positive, increasing the orientation to the obstacle. When the orientation is negative, the resulted angular velocity is also negative, driving the robot away to the obstacle.

Figure 4.7 illustrates some repulsive behaviors for different values of k_2. For high values of k_2, the robot turns faster when it is in the vicinity of the obstacle.

The repulsive behavior may lead the robot to a different state not included in the sequence retrieved by the navigation procedure. When this occurs, the topological localization algorithm identifies the new current state and the topological navigation computes a new sequence of states to reach the goal.

4.3.3 Combining Behaviors

The repulsive behavior emerges when the robot is in vicinity of an obstacle. For a non-holonomic robot, the distance to an obstacle is more important when the robot's orientation and the angle θ_o are close to each other, as explained by [96]. For instance, for the robot of the Rescue Project (the ATRV-Jr is a robot with differential wheels), it is more important to avoid an obstacle in front or aside it, since it has a differential drive kinematics.

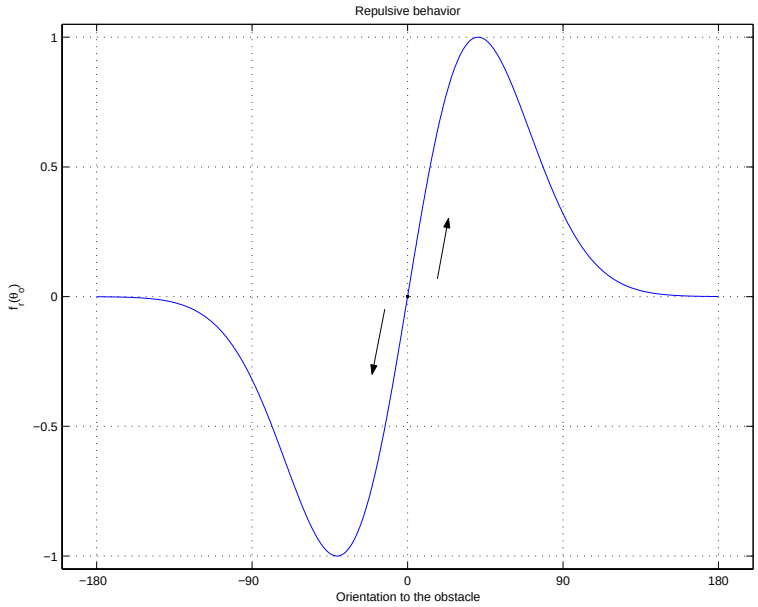


Figure 4.6: Repulsive behavior

To combine both behaviors it is necessary to measure the distance to the nearest obstacle. Since θ_o is defined as the orientation of the nearest point of the obstacle to the robot, it is also possible to measure that distance, d_o, as explained in [27]. If d_o is high meaning that the obstacle is far, the attractive behavior should decide the angular velocity, otherwise, the repulsive behavior becomes more important.

The two behaviors are mixed based on a simple strategy, similar to the Fuzzy Logic approach described in [33]. The function $g(d_o)$, dependent on the distance to the obstacle,

$$g(d_o) = \frac{1}{2} + \frac{1}{\pi}\arctan[k_3(d_o - k_4)], \tag{4.32}$$

combines both behaviors according to the value of d_o. The resulted behavior, a weighted combination of both behaviors, $f_w(d_o, \theta_o, \theta_e)$, is given by,

$$f_w(d_o, \theta_o, \theta_e) = f_a(\theta_e) \cdot g(d_o) + f_r(\theta_o) \cdot (1 - g(d_o)). \tag{4.33}$$

When the distance to the obstacle is equal to k_4 the attractive and repulsive behaviors have the same importance (both equal to 0.5). The value of k_3 influences not only the slope of the curve, but also the safe distance to the obstacle. In the example illustrated in Figure 4.8, with $k_4 = 4$ and $k_3 = 2$, when the distance to the obstacle is above 1m the weight of the attractive behavior is less then 0.05, while the weight of the repulsive behavior is approximately 0.95. The attractive behavior can become exactly 1 with a different function $g(d_o)$.

The angular velocity is now defined by combining both behaviors,

$$\Omega = f_w(d_o, \theta_o, \theta_e) \cdot \Omega_{max}. \tag{4.34}$$

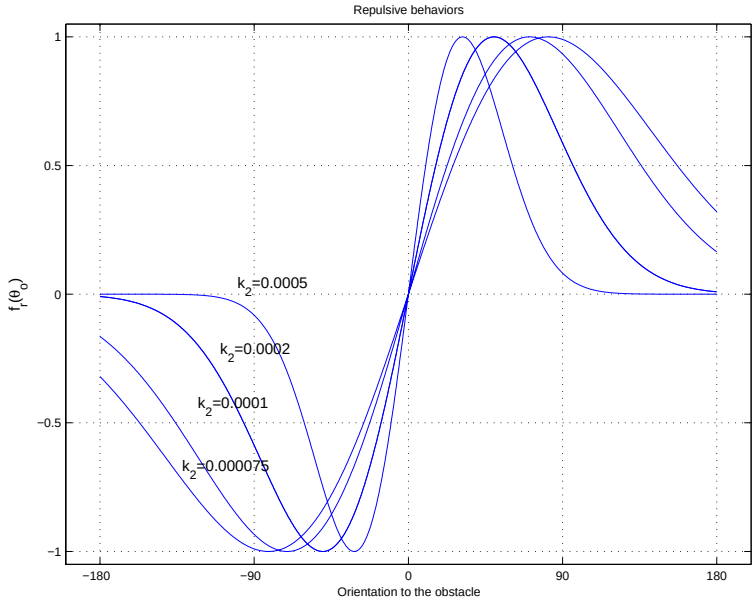


Figure 4.7: Repulsive behavior for different values of k_2

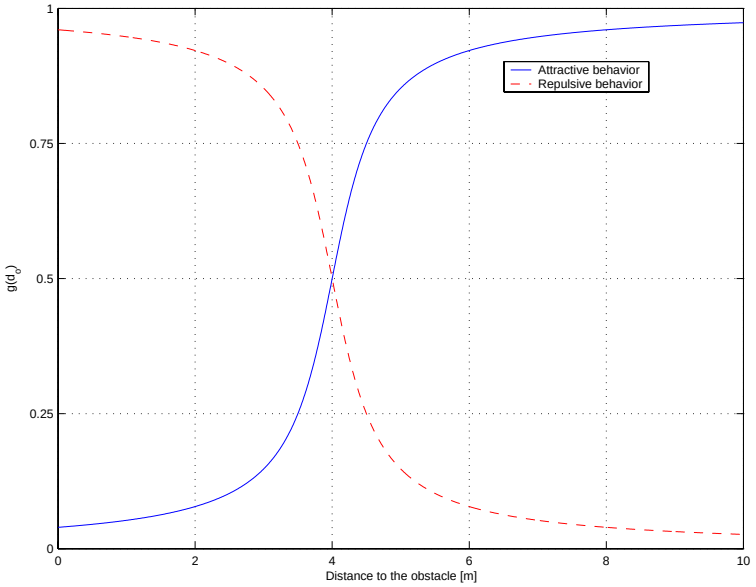


Figure 4.8: Combining the attractive and repulsive behaviors

In summary, the resulted behavior, f_w, depends on three variables and results from gathering the two components: $f_a(\theta_e) \cdot g(d_o)$ and $f_r(\theta_o) \cdot (1 - g(d_o))$. To simplify the visualization, these two components are illustrated individually in Figure 4.9 and Figure 4.10. Figure 4.9 illustrates the attractive component, or $f_a(\theta_e) \cdot g(d_o)$, with a small weight when the robot is close to an obstacle.

The repulsive behavior becomes more important when the robot is near to the obstacle.

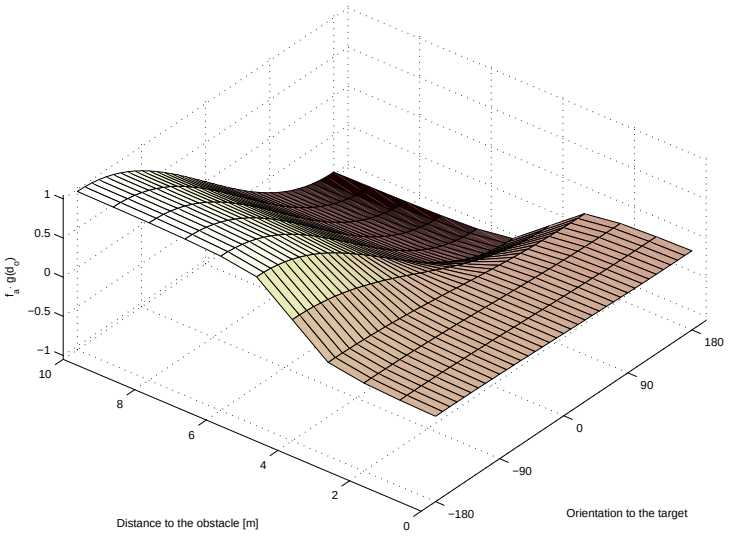


Figure 4.9: The attractive component of the driving behavior

4.4 Simulation Results

The navigation algorithm was tested including the topological localization described in Chapter 3 and the low level motion control methodology described in previous section. The test-bed map is the same displayed in Figure 3.8, defined by a set of states, but including a set of obstacles, as illustrated in Figure 4.11. The topological map, or the set of states, were computed without the obstacles, which were introduced a posteriori.

The first simulation depicted in Figure 4.12 is retrieved by imposing a set of state goals that the robot has to reach, starting at state s_1. The first goal is s_5. The navigation algorithm returns the sequence of states $s_1 \rightarrow s_2 \rightarrow s_6 \rightarrow s_5$. Even with the obstacles, the sequence of states remains unchanged during the path. When the state s_5 is reached a new goal is defined, s_2. At this stage, the navigation algorithm has to update the sequence a few times, since the robot is among obstacles. The experience is repeated to reach new goals, s_6 and s_4, starting always from the last goal. It is assumed that the robot is always in the area covered by the map. If the robot tries to leave the map, which is measured by the localization as an equal distribution at all states (ambiguity), the map should be updated with new states.

The next results are retrieved to test the length of the sequence states computed by the navigation algorithm. If the length of a sequence, defined as Δ (the number of states) introduced in Section 4.3, is constant, the navigation algorithm returns a sequence with the same length, each time it changes the previous sequence. As an example, for $\Delta = 4$ the best sequence of states from s_1 to s_5 is $s_1 \rightarrow s_2 \rightarrow s_6 \rightarrow s_5$. If the topological localization estimates the robot location as being state s_2 (or one of the other states of the sequence), the sequence is reduced starting in the current state. For instance, it the current state is s_2, the sequence becomes $s_2 \rightarrow s_6 \rightarrow s_5$. If the localization returns other state not covered

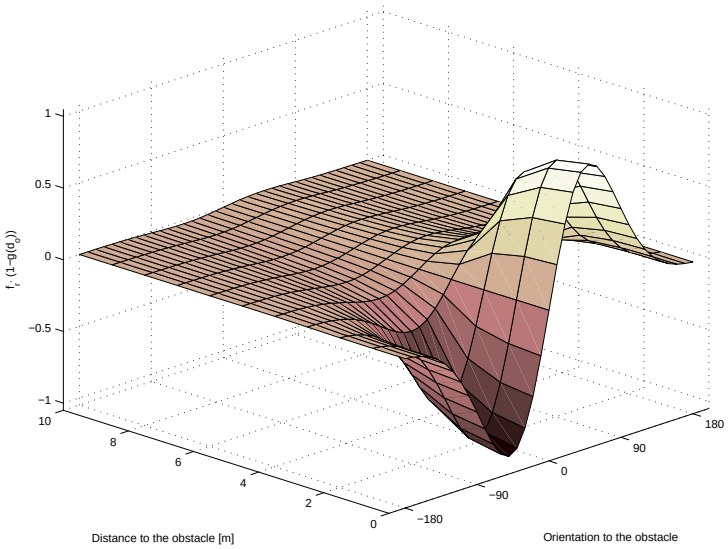


Figure 4.10: The repulsive component of the driving behavior

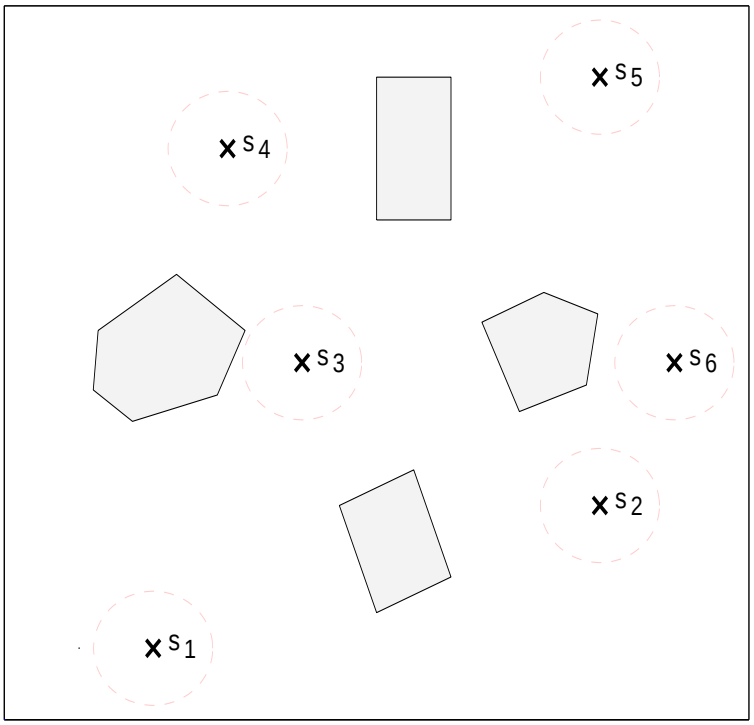


Figure 4.11: The map presented in Figure 3.8 compounded by obstacles

by the sequence, the navigation has to update to a new sequence, always starting with the same length Δ. This strategies seems absurd, since the robot may never reach the goal, but is useful for moving the robot always linked to several states. It is similar to a blind person moving in a corridor, who tries to search the walls and moving along these walls. When Δ is variable, for each time that the navigation algorithm returns a sequence, the

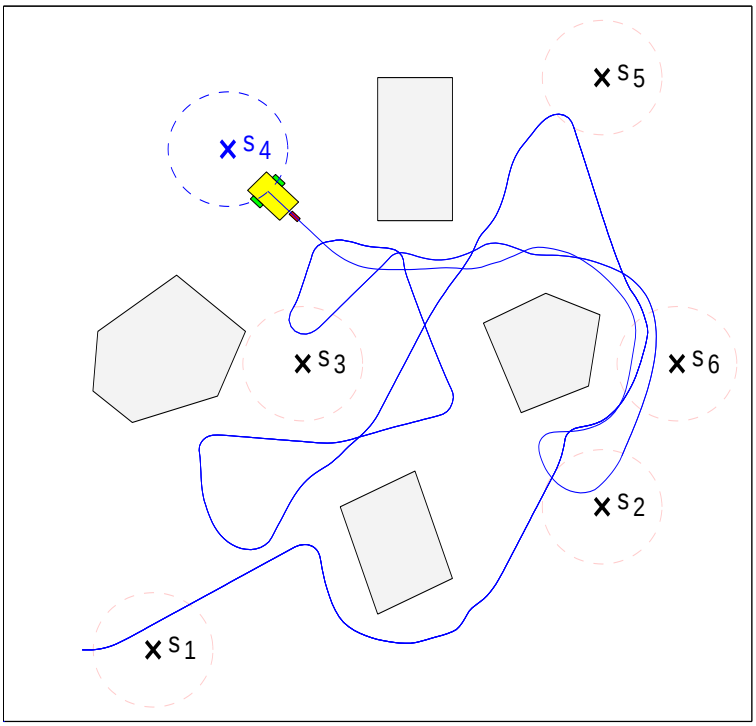


Figure 4.12: Example of a navigation algorithm, starting from state s_1 to reach s_5, s_2, s_6 and s_4 at this order

length may assume a different value, as explained in the Section 4.1.1.

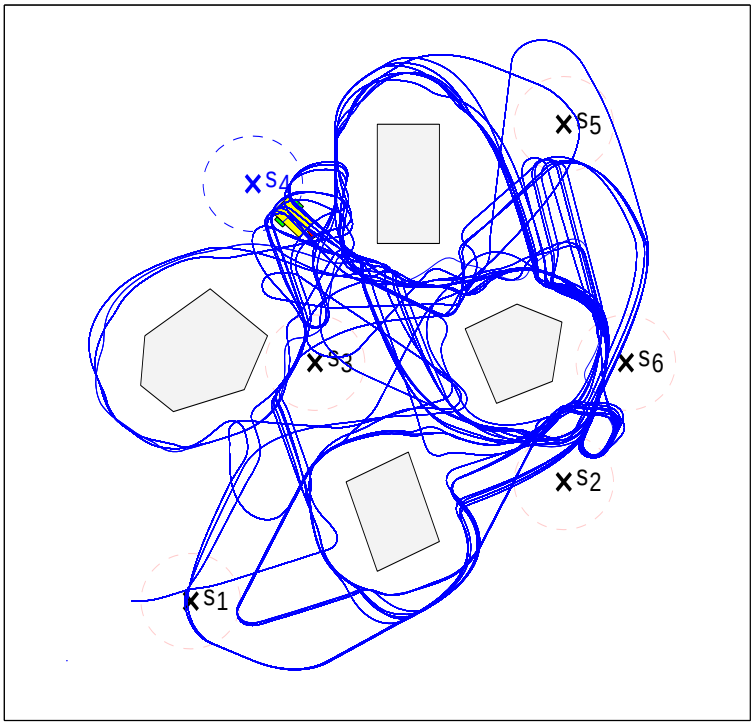


Figure 4.13: Resulted trajectories with Δ constant

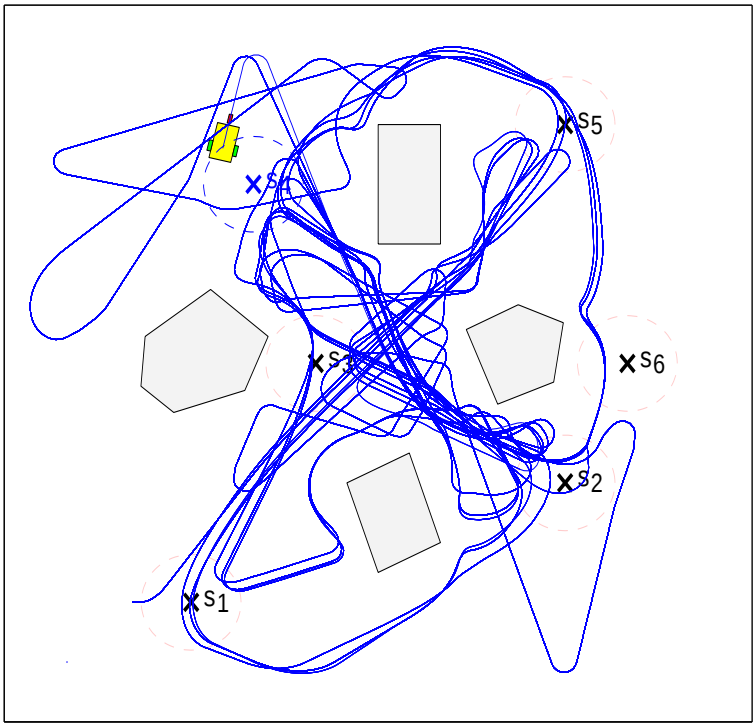


Figure 4.14: Resulted trajectories with Δ variable

To repeatedly test the navigation algorithm, we define a set of goals (around 50) that the robot has to follow at a specified order. The navigation was updating the sequence based on the localization results. Figure 4.13 depicts an example of the robot motion with $\Delta = 4$. The resulted trajectories describe a complex mesh, which is similar to a Voronoi diagram. In Figure 4.14, the parameter Δ is not constant. At this experiment, the robot reached the goals faster, with direct paths. When Δ is variable the navigation algorithm assumes a conquer profile, since the robot tries to leave the known area often.

When Δ is constant, as exemplified in Figure 4.13, the navigation starts returning always a sequence of 4 states. Even if the current state and the target state are closer (in the simulation the metric distance is available for experimental purposes), the navigation returns a sequence of 4 states. This situation, leads the robot following a long trajectory but with high certainty to be localized in each state. If the Δ is variable, the navigation commonly returns sequences of 2 states, the current and the target, as depicted in the Figure 4.14. However, the robot leaves the area covered by the current map more often, which, in a real situation with the localization, navigation and mapping implemented, requires a map update with a high frequency.

Chapter 5

Map Building

The previous chapters, the Topological Localization and Navigation, were addressed assuming that the topological map is known. This chapter discusses how to build and update a topological map, a representation of an environment with no metric information available. The map shows physical features that characterize particular locations or places that constitute the nodes of the map denoted as states. The features that characterize each state are represented by Gaussian pdfs. The main goal of this chapter is to determine the number of states and the parameters of the Gaussian pdf that define each state. The states can be represented by different type of features, where the extraction and selection procedures are also topics discussed in this chapter.

5.1 Problem Statement

In the present work, a topological map is built to support the navigation of a mobile robot. To perform a symbolic representation of the environment the robot perceives it with its on-board sensors and the acquired data is processed aiming at extracting the most relevant features of the environment. The built topological map provides the essential information for the navigation process.

The robot perception is condensed in observations, o_t, that represent the information obtained from the processing of the raw data acquired at each time instant t. For a time interval, T, the result is a sequence of observations, O_T. An observation is a vector where each component is related with a different feature, v_j. For instance, a feature defined as "Color" might have the values "red", "green" or "blue". This characterization using colors is translated in numerical values, similarly as the colors can be written in RGB format.

The notation used to characterize the observations is the following:

- $o_t = [o_t(v_1)o_t(v_2)\ldots o_t(v_M)]^T$ is a M-dimensional observation vector referred to time instant t, extracted from the rawdata r_t,
- $o_t(v_j)$ is the value of the attribute v_j extracted from the rawdata r_t,

- $o_t(v_j) = \emptyset$ states that the observation of the attribute v_j, at time instant t, is not achieved,
- $O_t = \{o_{t_0}, o_{t_1}, \ldots, o_t\}$ is an observation sequence from t_0 to t.

The different components of the observations reflect that the robot is able to perceive a diversity of attributes of the environment. These different types of perception have to be recorded in each state of the map. According to the uncertainty of the sensor measurements, each state s_j is represented by a Gaussian pdf, characterized by a mean vector μ_j and a covariance matrix, R_j, of dimension $[M \times M]$,

$$s_j \sim \mathcal{N}(o_i - \mu_j, R_j).$$

A map is composed by a set of states s_i and, consequently, is represented by a set of Gaussian pdfs, each one represented by its mean and covariance matrix, as shown in (5.1), with N being the number of states,

$$S = \{s_1, \ldots, s_N\} \sim \{\mathcal{N}(\mu_1, R_1), \ldots, \mathcal{N}(\mu_N, R_N)\}. \tag{5.1}$$

The Figure 5.1 shows an example of a map, where each state s_i is symbolically represented by the plot of the values of its mean vector, μ_i, whose components express the different types of features. In this example, each state is characterized by 5 features. The representation in Figure 5.1 does not provide any information on the state spatial distribution. The bindings represented by the grey arrowed lines express the probability transitions between states. In the proposed framework, these transitions result from a Hidden Markov Model approach described in [118].

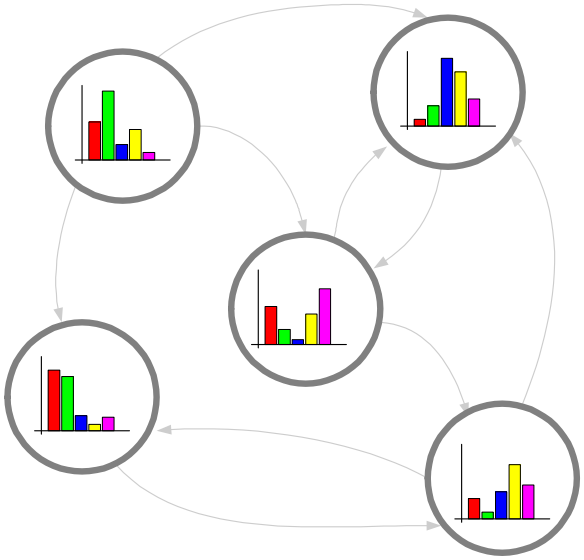


Figure 5.1: An example of a topological representation, where each state is characterized by 5 features

With this map characterization, the mapping procedure estimates the states, the mean vectors and the covariance matrices in (5.1) that maximize the probability of all observations given the environment model, i.e., that maximize the likelihood function $p(O_t|S)$,

$$\hat{S} = \arg\max_{S} p(O_t \mid S). \tag{5.2}$$

The $p(O_t \mid S)$ can be decomposed in a product of individual $p(o_i \mid S)$, based on the Markov Models, yielding

$$\begin{aligned} p(O_t \mid S) &= p(o_{t_0}, o_{t_0+1}, \ldots o_t \mid S) \\ &= p(o_t \mid S) \cdot p(o_{t_0}, \ldots, o_{t-1} \mid S) \\ &= p(o_t \mid S) \cdot p(o_{t-1} \mid S) \cdot \ldots \cdot p(o_{t_0} \mid S) \\ &= \prod_{i=1}^{t} p(o_i \mid S) \end{aligned} \tag{5.3}$$

or, equivalently, its logarithmic representation,

$$L(S) = \log p(O_t \mid S) = \sum_{i=1}^{t} \log p(o_i \mid S). \tag{5.4}$$

Given that S is a set of states, any observation o_i extracted from rawdata is a measurement of the state s_k with a probability c_k, for $k = 1, \ldots, N$. Therefore, the pdf $p(o_i \mid S)$ in (5.4) can be written as a combination of all the pdfs $p(o_i \mid s_k)$, $k = 1, \ldots, N$ weighted by the probability c_k, i.e.,

$$L(S) = \sum_{i=1}^{t} \log \left(\sum_{k=1}^{N} c_k \cdot p(o_i \mid s_k) \right). \tag{5.5}$$

The map estimate the set of states, $\hat{S}$, the argument that maximizes the likelihood function $L(S)$,

$$\hat{S} = \arg\max_{S} L(S). \tag{5.6}$$

The main issue at this point is to determine the states (defined by Gaussians pdfs) and the probabilities $c_k, k = 1, \ldots, N$, introduced in (5.5).

5.2 Maximization Criteria

The maximization of the likelihood function $L(S)$ in (5.5) is a hard problem to solve. A way to overcome the associated computational burden is by replacing the function $L(S)$

by the expectation of the likelihood given a previous estimation of the model, S^{old}, i.e.,

$$\begin{aligned} F(S) &= E\left\{L(S) \mid S^{old}\right\} \\ &= E\left\{\log p(O_t \mid S) \mid S^{old}\right\} \\ &= E\left\{\sum_{i=1}^{t} \log p(o_i \mid S) \mid S^{old}\right\} \\ &= E\left\{(\log p(o_1 \mid S) + \ldots + \log p(o_t \mid S)) \mid S^{old}\right\} \end{aligned} \tag{5.7}$$

this corresponding to the use of the Estimation and Maximization algorithm [93, 111, 113].

Given a previous estimate of the model, S^{old}, it is assumed that it is possible to evaluate the probability that the observation o_i belongs to the state s_j, herein denoted by w_{ij}. Accordingly, $\log p(o_i \mid S)$ in (5.4) can be written as $\log(c_j \cdot p(o_i \mid s_j))$ with probability w_{ij}. Therefore, the likelihood function (5.7) becomes

$$\begin{aligned} F(S) &= \sum_{j=1}^{N}\sum_{i=1}^{t} w_{ij} \log(c_j \cdot p(o_i \mid s_j)) \\ &= \sum_{j=1}^{N}\sum_{i=1}^{t} w_{ij} \log(c_j \cdot \mathcal{N}(o_i - \mu_j, R_j)) \\ &= \sum_{j=1}^{N}\sum_{i=1}^{t} w_{ij} \left[\log c_j - \log\left((2\pi)^{\frac{M}{2}}\sqrt{|R_j|}\right) - \frac{1}{2}(o_i - \mu_j)^T R_j^{-1}(o_i - \mu_j)\right]. \end{aligned} \tag{5.8}$$

The likelihood function $L(S)$ is replaced by $F(S)$ and the map, or the set of states, S, is the argument that maximizes $F(S)$,

$$\hat{S} = \arg\max_{S} F(S). \tag{5.9}$$

This maximization may be accomplished using the Expectation-Maximization Algorithm described in the following section.

5.2.1 Expectation-Maximization (EM) Algorithm

The Expectation-Maximization Algorithm described in [98] is implemented in three steps, the initialization, the expectation and the maximization. The expectation updates the values of w_{ij} given a previous estimation of the map, S^{old}. The maximization determines a new map based on the updated w_{ij}.

Expectation Step

The probability w_{ij} in (5.8) is evaluated using the values of the previous map estimation, S^{old},

$$w_{ij} = \eta \cdot c_j^{old} \cdot \mathcal{N}(o_i - \mu_j^{old}, R_j^{old}), \tag{5.10}$$

where η is a normalization factor.

This corresponds to the Expectation step of EM algorithm, as described in [98].

Maximization Step

The next step of the EM algorithm is the maximization of (5.8). The selected approach for the maximization is based on the Lagrangean of the likelihood function $F(S)$,

$$Q(S) = F(S) + \lambda(\sum_{j=1}^{N} c_j - 1),$$

where λ, the Lagrange Multiplier, is introduced to estimate the c_i.

The equations:

$$\frac{\partial}{\partial \mu_j} Q = 0, \quad \frac{\partial}{\partial R_j} Q = 0, \quad \frac{\partial}{\partial c_j} Q = 0$$

provide the values for μ_j, R_j and c_j parameters leading to

$$\mu_j = \frac{1}{\sum_{i=1}^{t} w_{ij}} \sum_{i=1}^{t} w_{ij} o_i, \tag{5.11}$$

$$R_j = \frac{1}{\sum_{i=1}^{t} w_{ij}} \sum_{i=1}^{t} w_{ij} (o_i - \mu_j)(o_i - \mu_j)^T, \tag{5.12}$$

$$c_j = \frac{1}{\lambda} \sum_{i=1}^{t} w_{ij}. \tag{5.13}$$

The constraint $\sum_{j=1}^{N} c_j = 1$ (total probability for all possible states of the model) leads to

$$\sum_{j=1}^{N} \left(\frac{1}{\lambda} \sum_{i=1}^{t} w_{ij} \right) = 1,$$

which corresponds to $\lambda = t$, a normalization in time and, consequently, (5.13) becomes

$$c_j = \frac{1}{t} \sum_{i=1}^{t} w_{ij}.$$

After the maximization process in one step, the algorithm returns to the expectation step with the values of (5.11) - (5.13). This procedure repeats until all the w_{ij} parameters have stabilized.

Initialization and Stopping Criteria

The EM algorithm is initialized with

$$c_j = \frac{1}{N}, \quad \mu_j, R_j = random \quad \forall j = 1, \ldots, N$$

that are used in the first expectation step. The parameters μ and R are generated by a uniform pdf, assuming values in an interval bounded by the physical constraints of the sensors.

The EM algorithm, as an iterative procedure, also requires a stopping criteria. This algorithm belongs to the class of unsupervised learning algorithms and, consequently, the only variable that expresses the representation accuracy is the parameter w_{ij}. As mentioned above, w_{ij} is the probability that the observation o_i belongs to the state s_j and, according to the maximization step of EM, w_{ij} stabilizes after some iterations. The parameter w_{ij} is considered stabilized when the difference between a couple of successive iterations is less or equal to a given threshold, for all the observations o_i. It is important to retain that the iterations of EM occur between successive observations, as illustrated in Figure 5.2.

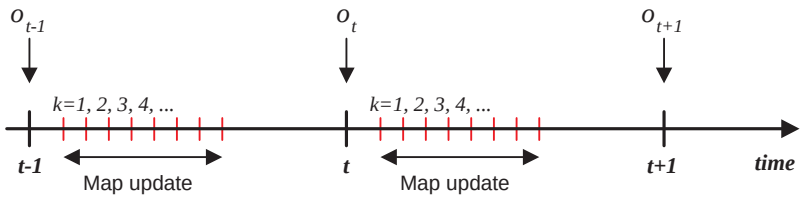


Figure 5.2: Expectation-Maximization iterations

Moreover, to evaluate the stabilization of w_{ij}, it is necessary to record the value of w_{ij} in each iteration k, $w_{ij}(k)$, as depicted in Figure 5.2. The differences between a sequence of L iterations is evaluated for a single state s_j, as:

$$\delta_j = \sum_{i=1}^{t} \sum_{l=1}^{L} |w_{ij}(k) - w_{ij}(k-l)|. \tag{5.14}$$

To evaluate the overall stabilization of the algorithm it is necessary to define the convergence for all the states,

$$\delta = \sum_{j=1}^{N} \delta_j. \tag{5.15}$$

When δ in (5.15) is lower than a given threshold Δ_δ, it is considered that the EM algorithm has stabilized. The value of Δ_δ is defined according the oscillations accepted during the stabilization of the algorithm, i.e., when an observation becomes part of a different state.

After stabilization, it is necessary to analyze the quality of the representation obtained by the algorithm. A good representation, s_j, for the observation o_i, corresponds to a high value of w_{ij}. Moreover, a good representation for all the observations $o_i, i = 1, \ldots, t$, reducing the number of outliers, requires high values w_{ij} for all the states s_j. However, a good representation may not occur, this resulting from the existence of spurious states or even from a small number of states for the representation.

5.2.2 Dynamic EM

According to the previous subsection the initial number of states, N, which is constant during the EM algorithm, does not necessarily guarantees a good representation of the environment. Even if a good representation is achieved at a given time instant, a possible update of the number of states might be required as the robot is always acquiring new measurements. Consequently, it is strictly necessary to re-evaluate the number of states after the stabilization of the EM algorithm. The loop procedure for this re-evaluation is represented in Figure 5.3.

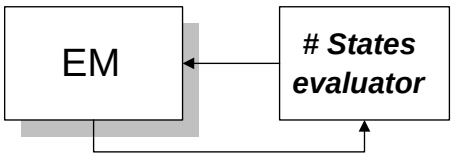


Figure 5.3: Evaluation of the number of states

The evaluation of the number of states is explained in the sequel and represented in Figure 5.4. Starting with an initial estimate of the number of states, the EM algorithm is applied iteratively. As represented in Figure 5.4, when $\delta < \Delta_\delta$ the EM algorithm converges to an environment representation. The next step assesses the quality of each state to find possible superfluous states or the requirements of each states. To implement this analysis, it is necessary to evaluate the amount of observations represented by each state, s_j. A natural way to formalize this analysis is consider the entropy of state s_j, H_j, as referred in [74],

$$H_j\left(w_{ij}\right) = \sum_{i=1}^{t} w_{ij} \log(w_{ij}). \tag{5.16}$$

The parameter H_j quantifies the accuracy of the representation for state s_j. For high values of the entropy (the entropy, H, assumes negative values) the state represent several observations, while for low values of H some observations are represented by more than one state. If H_j is less or equal than a given threshold, H_{min}, the j-th state is removed and the number of states is decreased by one. If $H_j > H_{min}$ for all $j = 1, \ldots, N$, the set of observations requires a new state to improve the representation. Accordingly, the number of states is increased by one and the new state is initialized by a mean vector μ

and a covariance matrix R as described in Section 5.2.3. The EM algorithm runs again and when the stabilization is reached for this new number of states, a new evaluation takes place, along the steps in Figure 5.4, to check if the new state improved (or not) the representation.

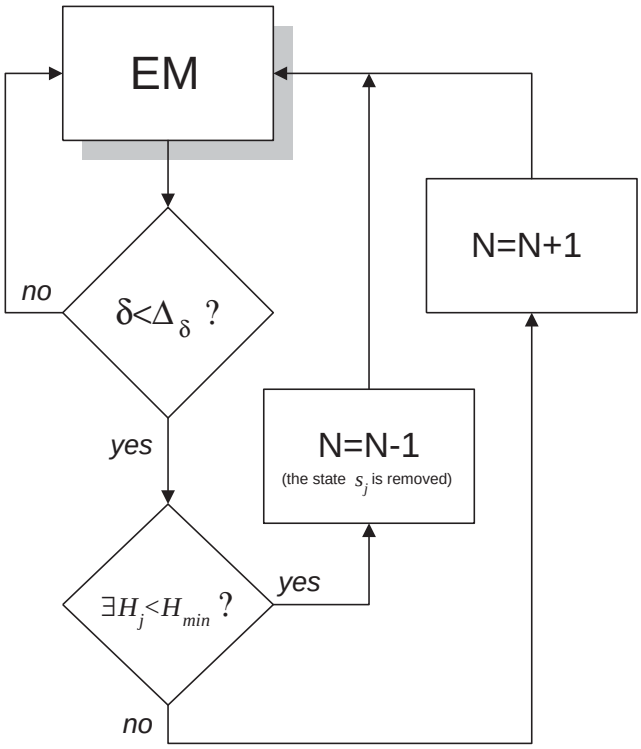


Figure 5.4: Evaluation of the number of states

Low values for H_{min} lead to a large number of states, that could yield states with similar parameters. Additionally, the parameter Δ_δ defines the level of oscillations of the algorithm state estimates for a constant value of i.

As represented in Figure 5.4, the algorithm is always trying to adjust the number of states to update the model. Therefore it is prepared for changes in the environment, adding new states and/or removing useless states. Nevertheless, without any changes in the environment, the algorithm converges and the number of states oscillates around a particular value. Whenever these oscillations are above a threshold during a given time interval the algorithm stops.

5.2.3 Initializations in the Dynamic EM

In the previous subsection, when the number of states increases, the new state has to be initialized, i.e., the values of the mean vector and the covariance matrix of the associated Gaussian pdf have to be established. According to the amount of the observations acquired since the time instant t_0 and the iterative nature of the EM algorithm, it is strictly necessary to optimize the initialization procedure to reduce time consumption of the con-

vergence. The initialization step establishes the values μ and R for the new state. There are two possible ways to accomplish this:

- **Random combinations of the current states:** when more than one state is identified, it is possible to generate the mean of the new state as a combination of the means of two or more of the current states. The selected states $s_{sel_1}, s_{sel_2}, \ldots, s_{sel_{Nsel}}$ $(N_{sel} = 2, 3, \ldots N-1)$ are randomly selected with uniform distribution and the initialization follows,

$$\mu_{N+1} = \frac{1}{N_{sel}} \sum_{j=1}^{N_{sel}} \mu_{sel_j} \quad , \quad R_{N+1} = \frac{1}{N_{sel}} \sum_{j=1}^{N_{sel}} R_{sel_j} \tag{5.17}$$

- **Random values:** the mean value of the new state is randomly generated with different types of probability density functions (the covariance matrix is generated in the same way as explained on the initialization of the EM):
  - Uniform: each observation is represented by a vector of features, where each component (representing a different type of feature) assumes values of an interval (for instance, the feature "free-area" assumes values between 0 and a maximum range and the initialization generates a value inside this interval with an uniform pdf).
  - With the $1 - P(O \mid S)$ with which the initialization generates news states to cover observations disregarded by the current representation.

The mapping algorithm builds a representation of the environment based on a topological approach, where each state is represented by a Gaussian. The algorithm is permanently improving the environment representation, updating the Gaussians and the number of states according to the entropy. The overall algorithm is illustrated in Figure 5.5, where the observation's buffer increases along time and the mapping algorithm, with several iterations, updates the number of states and their parameters between successive observations.

5.3 Simulation Results

The mapping algorithm was tested supported on simulated observed features. The feature selection and feature extraction procedures are covered in Section 5.4 and, consequently, the observed features have to be simulated to test only the performance of the mapping algorithm. To generate the observations, we created 6 Gaussians pdfs to generate 6 different sets of observed features, as depicted in Figure 5.6. Each observation is defined as a vector of two elements, representing a Feature A and a Feature B. The generated observations are mixed and introduced into the mapping algorithm. It is possible to understand the correspondence between the observations in Figure 5.7-a) (intentionally colored to simplify the perception) and the Gaussian pdfs, i.e., the samples generated by Gaussians pdfs of

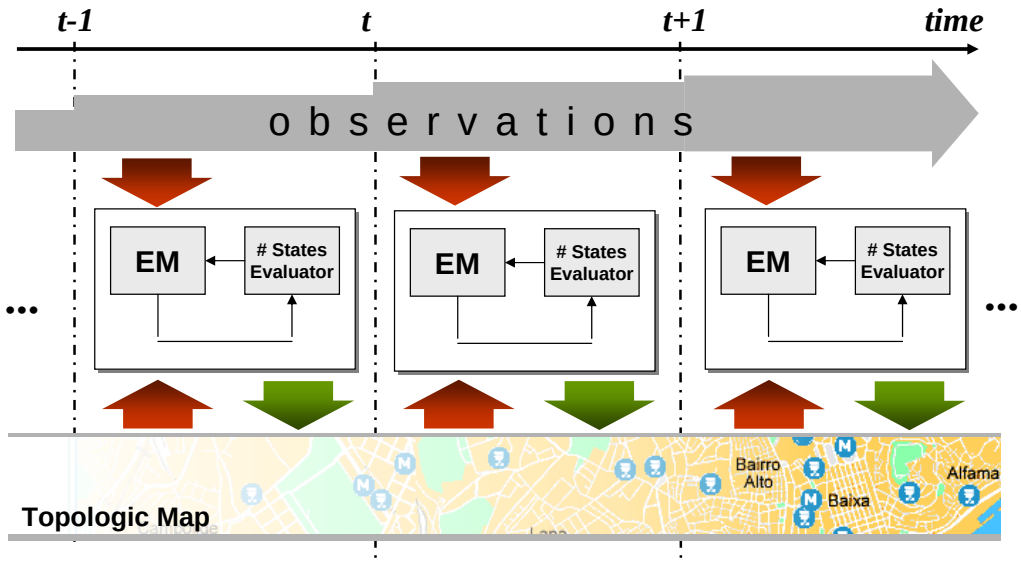


Figure 5.5: Brief illustration of the mapping algorithm

the Figure 5.6. However, in the perspective of the algorithm, there is no distinction, as shown in Figure 5.7-b) where no color distinction was introduced.

It is assumed that each state is represented by a single Gaussian. The mapping algorithm is initialized with 2 states, which means 2 Gaussian pdfs, where the means and covariance matrices are initialized by random values (uniform distribution).

After few iterations, the mapping algorithm retrieves the best topological representation with two states. When the convergence is achieved, the number of states is increased by one, as illustrated in Figure 5.9-a) and the algorithm stabilizes at a new representation, with three states at this iteration. When new states are created, some of them could not satisfy the minimum entropy, H_{min} and, consequently, they are removed and initialized with a new state, as illustrated by some peaks (decreasing the number of states, which requires a re-stabilization between 10-15 iterations) in Figure 5.9-b). This justify the importance of the initialization procedure, which defines the rate of convergence. If a new state appears in a region (in the referential where the axis are the types of features) well represented by the current states, it is natural that the new state never reaches a high level of entropy.

The simulation experiment leads to a topological map composed by 6 states (there are 7 states, but the State 5 is removed given the level of entropy), as shown in Figure 5.8, whose parameters are described in Table 5.1. The Gaussians are represented with different colors to illustrate the features covered by each state. The simulation does not contain metric information and therefore the plots are only along the types of features. Since the observed features are 2D dimension it is possible to illustrate the topological map. However, when the dimension is larger, it is only possible to make projection on 2D or 3D axis, as exemplified in Figure 5.8-b), or by symbolic representations as in Figure 5.10, with

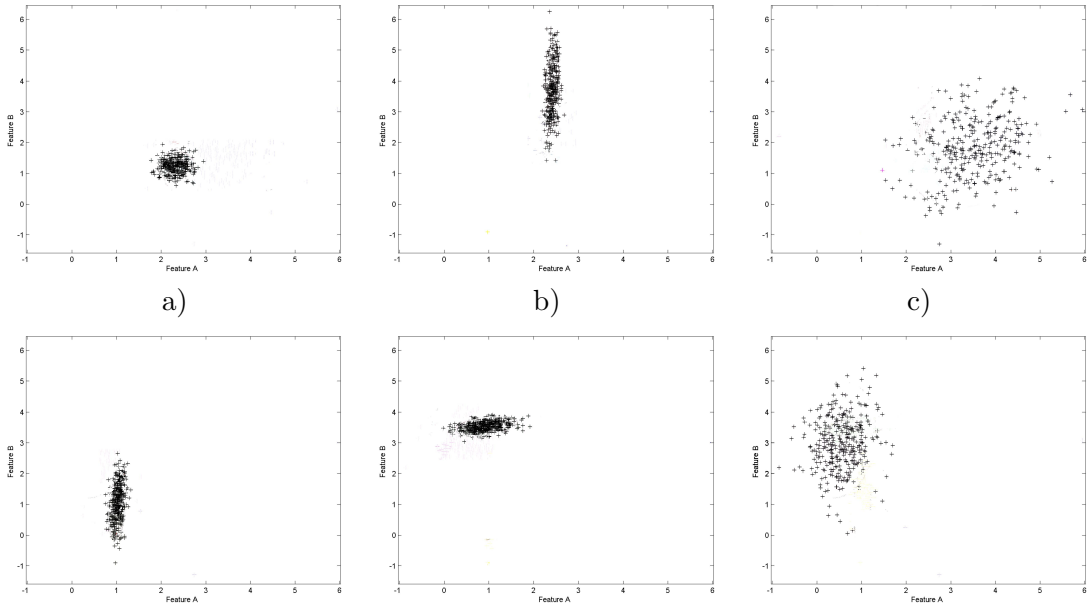


Figure 5.6: The 6 sets of observations generated by the 6 different Gaussians pdf

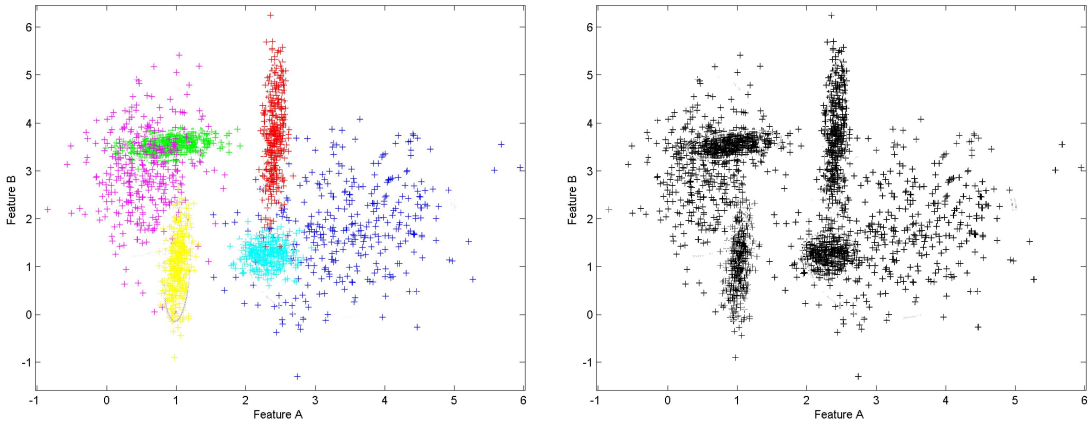
the Gaussian parameters described in Table 5.1. The Figure 5.10 also includes the lines that represent the connection or the direction between states. As explained in Chapter 4, the directions between states are acquired during a learning phase, where the robot's orientation is recorded when it changes between two states.

	state 1	state 2	state 3	state 4	state 5	state 6
μ_j	0.9517 3.5260	3.4447 1.7646	2.3192 1.2264	0.5385 3.0846	1.0182 1.0147	2.4205 3.6809
R_j	0.1174 0.0157 0.0157 0.0218	0.5837 0.0962 0.0962 0.8647	0.0455 0.0023 0.0023 0.0531	0.1564 0.0515 0.0515 0.8831	0.0097 0.0141 0.0141 0.3549	0.0062 0.0129 0.0129 0.6779

Table 5.1: The estimated Gaussians

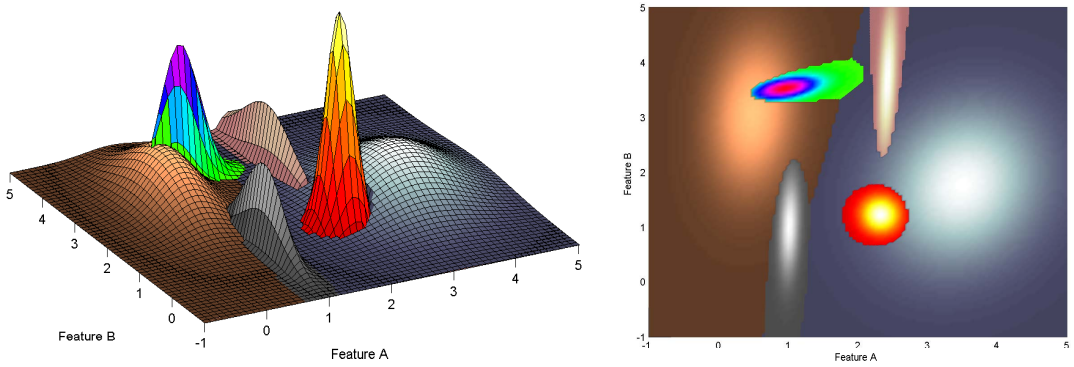
In this experiment, the topological mapping algorithm was tested using a single Gaussian for each state. As described previously, in the topological map the probability that the observed feature o_i belongs to the state s_j is expressed by w_{ij}. Applying a threshold criteria, the observed feature o_i is assumed to belong to the state s_j if w_{ij} has the maximum value for that state. However, it is possible to describe the same state with more than one Gaussian. To achieve this goal, we should apply the same map algorithm only to the observations associated to this state. This can be assumed as a topological map where each state contains a sub-topological map.

In this simulation experience, it is assumed that no initial map is available at the beginning and, consequently, the mapping algorithm was initialized with two Gaussians. The algorithm would be initialized with an a priori map if available. In both cases, the acquisition of new features leads to a map improvement through the mapping algorithm, which could result in adjustments on the current states (tuning of the Gaussians parameters) or



a) b)

Figure 5.7: All the observations are mixed: a) colored for visualization and b) as introduced into the mapping algorithm



a) b)

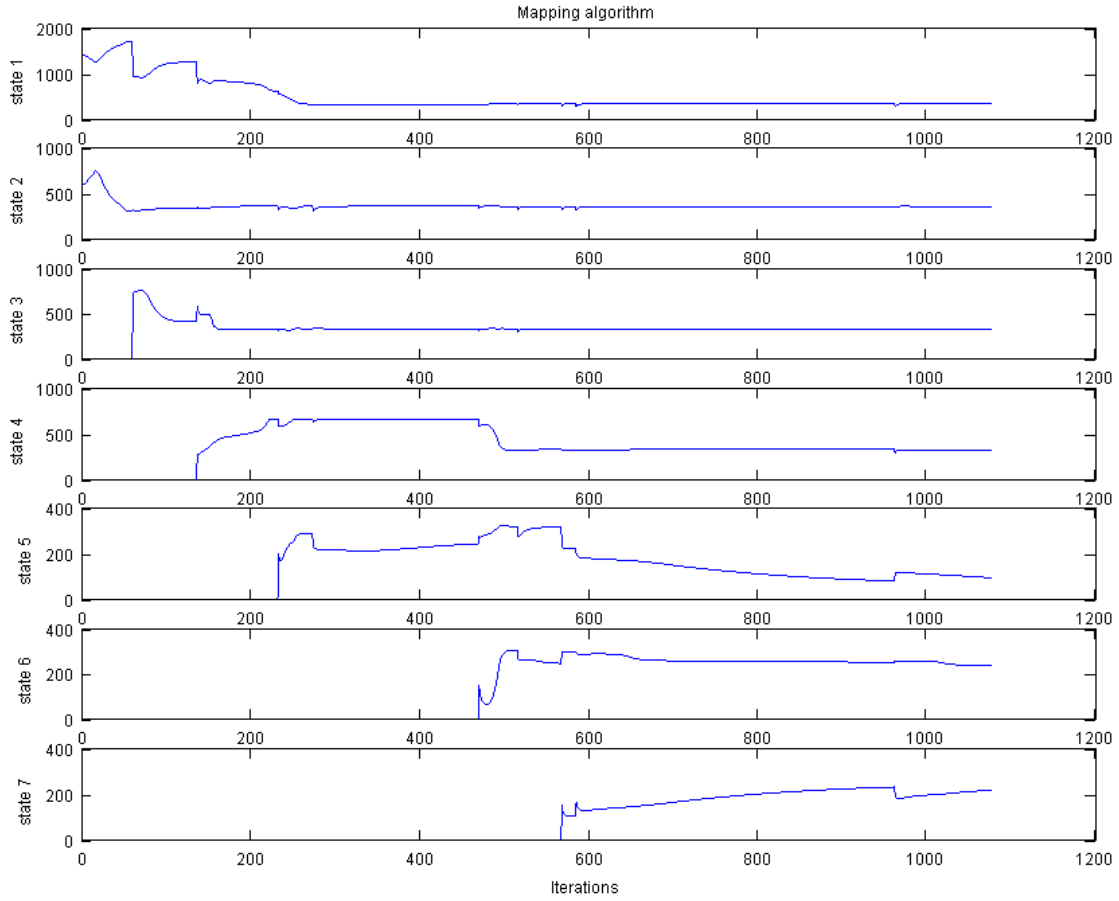
Figure 5.8: Simulation result, generating 6 states (6 Gaussians) with 2 different types of features, from different point of view a) 3D representation b) top view

even in adding/removing one or more states.

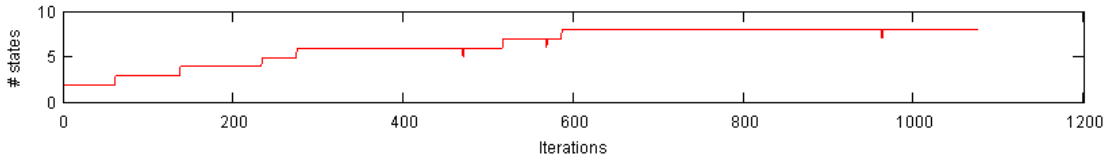
In summary, the procedure to build a topological map is based on a clustering procedure applied on the observed features that classify them as states, modeled by Gaussian pdfs. But a question arises: how does the mapping algorithm depend on the types of features? Different scenarios require different type features. This is the topic of the next section.

5.4 Features

One application of mobile robots is to carry out tasks in unstructured environments often without or with a reduced a priori knowledge of the scene map. To accomplish this type of mission, mobile robots have to adapt, recognize, localize and navigate simultaneously, while moving towards the desired target goal.



a)



b)

Figure 5.9: Mapping algorithm evolution, a) for each state and b) the number of states of the topological map

If no environment representation is available, mobile robots have to build it based on the observations acquired by on-board sensors. Large scenarios, as those in outdoor environments, rides to a large amount of information to store, including that required to build the map, but also the localization and navigation algorithms. Topological representations, as the ones considered in the thesis, are based on landmarks characterized by features. Therefore, a feature extraction procedure, reducing the data acquired by the sensors but retaining the crucial information, is required. Features have to support the topological navigation of mobile robots in different scenarios but not every type of feature is essential to represent a particular scenario, this requiring a feature selection criteria.

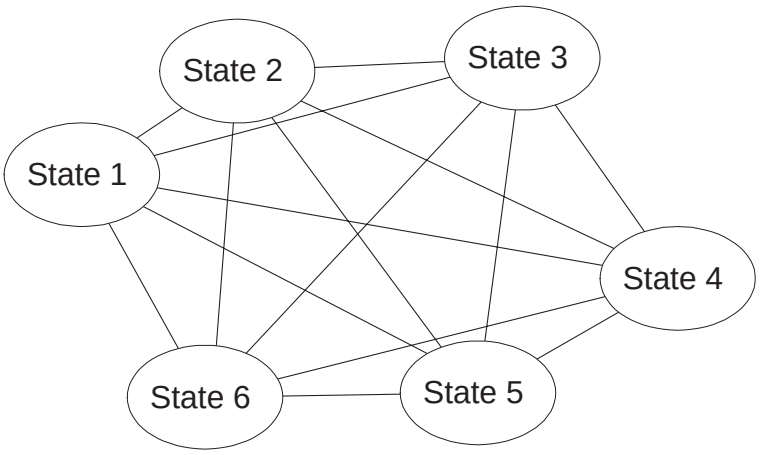


Figure 5.10: The symbolic representation of the topological map

In natural scenes, there are several features: corners, distinctive features such as buildings, streetlamps, placard, vertex and lines junctions, colors, textures, vertical edges [13, 39, 58, 59, 70, 109]. The key question is how to determine the best set of features in an outdoor scene, aiming at producing a topological map that supports the navigation of a mobile robot.

In this section we address the problem of feature extraction and selection to build a topological map. Given a set of rawdata acquired by the on-board sensors, the thesis describes how to obtain the features used in the mapping algorithm that supports the localization and navigation of a mobile robot in outdoor environments [119]. The main goal of this section is the choice of the best features, according to a statistical criteria, that fits on the scenario representation. Our approach uses the data acquired by different types of sensors, namely range sensors, inertial sensors, and a standard camera installed on top of the mobile robot. The main features are extracted from the rawdata acquired by the camera, since this sensor acquires more relevant information.

5.4.1 Related Work

To obtain a topological representation from a real environment, it is necessary to perform a feature extraction procedure from on-board sensor data (e.g., vision camera, laser range finder). Some works deal with this issue.

Santos-Victor *et al.* [104], proposed a vision-based navigation which takes into account special spatial representations and visual geometries. The navigation problem is based on the decomposition of sub-goals, identified by recognizable landmarks.

Ulrich *et al.* [115], presented an appearance-based place recognition system for topological localization. This work focuses on color images to distinguish the places. Hähnel *et al.* [52], discusses the problem of creating maps in dynamic environments, using a technique to identify dynamic objects.

Zhou *et al.* [130], proposes structural features for content-based image retrieval (CBIR),

especially edge/structure features extracted from edge maps. They describe a new algorithm to extract edges. Experiments show that the features can catch salient edge/structure information and improve the retrieval performance.

The Principal Component Analysis (PCA) obtains a feature set and provides a mathematical model that quantifies the loss of information contained in the images [56]. It supplies a linear representation of the original data using the least number of components with the minimum mean-square error. PCA has been successfully used in several robotic applications to find linear features from intensity data. It can also be applied on laser range data [124, 125]. Thrun [110], presents a method to learn what features/landmarks are best suited for localization, using neural networks. Vlassis *et al.* [122], proposes a method for an appearance based modeling of the environment, using linear image features extracted using PCA.

This section addresses some of the issues described in the reviewed literature, resumed in [117]. In particular, we take the best characteristics of different types of features (e.g., vertical edges, color histograms, PCA) and propose the choice of the best features for mapping based on a statistical criteria.

5.4.2 Feature Extraction

A feature extraction procedure corresponds to the projection of a high-dimensional data space onto a low dimensional subspace leading, in most cases, to a loss of information. Any feature extraction method must satisfy the following properties [122]:

- Robustness to small displacements or small scenario changes,
- Invariant to lighting conditions,
- Robustness to partial occlusion,
- Fast computation,
- Capacity to compress the images as much as possible while retaining pertinent information.

Different feature types can be used to solve the mobile robot localization problem, in particular geometrical features (lines, corners, edges, shapes), color, textures and whatever can be identified as a landmark. Given the huge amount of information acquired by the sensors as rawdata but also the requirement to support localization it is necessary to choose the best features to represent a landmark and to carry out a feature selection procedure.

An important goal to support robot navigation is to achieve a good and optimized representation of features to improve the performance of the matching required in the mapping procedure, as described in [119].

The notation used for feature extraction and selection within the mapping procedure is:

- r_t is the rawdata acquired by the sensors at time instant t, considered as a vector of dimension D,
- o_t is a M-dimensional feature vector extracted at time instant t. It results from a data processing on r_t,
- $FE(r_t)$ is a nonlinear function to extract features from the rawdata r_t,
- o_t^a is the type of feature a, extracted from r_t at time instant t,
- O_t is the observation sequence up to the time instant t, i.e. $\{o_1, o_2, \ldots, o_t\}$.

According to the previous notation we herein state, for the sake of clarification, that whenever we refer observation, o_t, we are considering the features extracted from rawdata r_t. The rawdata r_t integrates all the sensor information available. The feature vector, o_t, is extracted at each time instant t from the rawdata r_t by a nonlinear function FE,

$$o_t = FE(r_t),$$

where

$$FE : \mathbb{R}^D \longrightarrow \mathbb{R}^M.$$

Feature extraction has an underlined sensor fusion procedure, since the input variable, r_t, may contain information from different sensors (e.g., intensity and range data). The extraction function FE reduces the amount of data, retaining only the essential information of sensor data. For that reason, $FE^{-1}(o_t) \supset r_t$, which means that different rawdata vectors could lead to the same feature. When this happens, it is important to identify if the features were extracted in the same place to avoid ambiguities, or in places where the distinction among them is not important. The performance of FE is addressed in Section 5.4.3, where the best features, o^*, which are time independent, are chosen.

The features can be extracted from the rawdata acquired by any type of sensors, including range sensors or inertial sensors. However, the most important features are retrieved by intensity sensors as a vision camera, given the complexity of the environment and the limitations of the other sensors namely the range sensors. The following subsections describe some important algorithms to extract the essential information from images. The selected algorithms are relevant to test the feature extraction procedure in different scenarios to build a topological map that supports the localization and navigation.

Edges and Hough-Transform

Most of the relevant features in outdoor environments are the vertical edges, which identify important structures as buildings or trees. This subsection describes the procedure to select the most important vertical edges in the image.

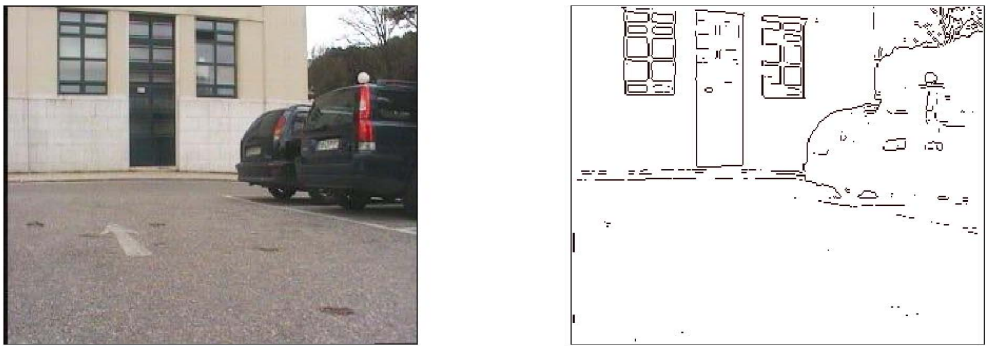
As described in [40, 65] the image dependencies due to lighting source and illuminance, mainly in outdoor environments, require a color image normalization procedure.

This drawback points towards the use of edge-based features to support environment representation and robot navigation [71, 83, 122, 130].

To extract edges from an image a specific filter (e.g., Sobel, Prewitt, Roberts, Gaussian, or other) is applied, where the original image and the identified edges are illustrated in Figure 5.11. In outdoor environments, where the scenario is unstructured, it is important to detect the vertical ones. Moreover, the edges present noisy information and therefore it is necessary to remove or, at least, reduce the superfluous data, applying the Hough Transform (HT) to the edges [49, 129]. This technique yields an histogram of straight lines for different directions, as shown in Figure 5.12-a), where the darkness corresponds to the amount of pixels that belong to a specific line. The Figure 5.12-b), shows the most relevant edges on the image, corresponding to the points represented in Figure 5.12-a).

A straight line is defined by (φ, θ), $x\cos\theta + y\sin\theta = \varphi$, where (x, y) are the coordinates of an image pixel and (φ, θ) the distance and the inclination of each edge in the image. To select only the vertical or near to vertical edges, the directions are chosen around 0 (between 0 and 5 degrees) and 180 degrees (between 175 and 180 degrees), as exemplified in Figure 5.13-a). The k_{edges} straight lines with the larger number of pixels (high level on the histogram) are selected and considered as the edges' features extracted from the image. The image in Figure 5.13-b) shows the result. To reduce the dependence of φ_i from the sensor's orientation (the image is acquired by a non-omni-directional camera), rather than considering φ_i to characterize the edge, we consider the distance between two consecutive straight lines, i.e., $d_i = \varphi_{i+1} - \varphi_i$, yielding

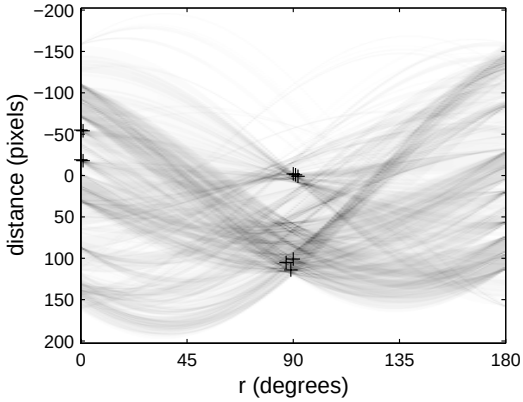
$$o_t^{Edges} = FE(r_t) = \begin{bmatrix} d_1 & d_2 & \cdots & d_{k_{edges}} \\ \theta_1 & \theta_2 & \cdots & \theta_{k_{edges}} \end{bmatrix}. \tag{5.18}$$



a) b)

Figure 5.11: Example of edge detection: a) original image and b) the most important edges

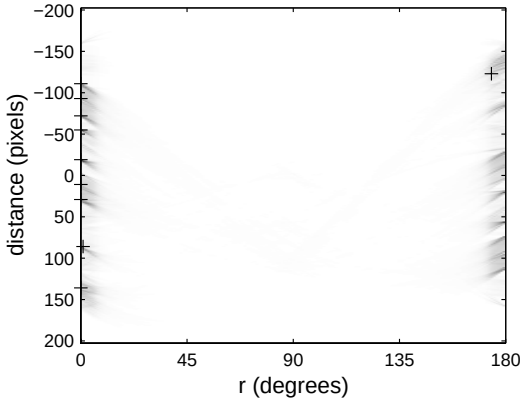
In [49] it is focused the relevance of the geometric forms of the images (square or rectangular), which underlines the importance of vertical and/or horizontal edges in the image.



a)

b)

Figure 5.12: Example of straight-lines detection: a) the Hough Transform of all edges and b) the most important straight-lines edges



a)

b)

Figure 5.13: Example of vertical edges detection: a) the Hough Transform of vertical edges and b) the most important vertical edges

Histogram Parameterizations

Even with the inherent light and geometric bindings, the color is an important source of information. Applying a normalization procedure, as suggested by [40], or simply, using the HSV colormap in spite of RGB, color histogram are important features. However, histograms provide large amount of information that could be parameterized as exemplified in Figure 5.14. We tested the parameterization of Hue and Saturation histograms using polynomials and a sum of Gaussian pdf functions. A parameterization using a polynomial function of order n requires $n+1$ parameters $(a_0, a_1, \ldots, a_n)$, while a parameterization by a sum of n Gaussians, $\mathcal{N}(\mu, \sigma)$, requires $3n$ parameters (weights, means and variances). The parameterization error is evaluated by the square error of the original and the parameterized histograms. We carried out experimental tests with a large amount of images acquired in different places of outdoor environments. The corresponding Gaussian parameterization errors are lower when the number of parameters are equal or larger than 6, as

shown in Table 5.2.

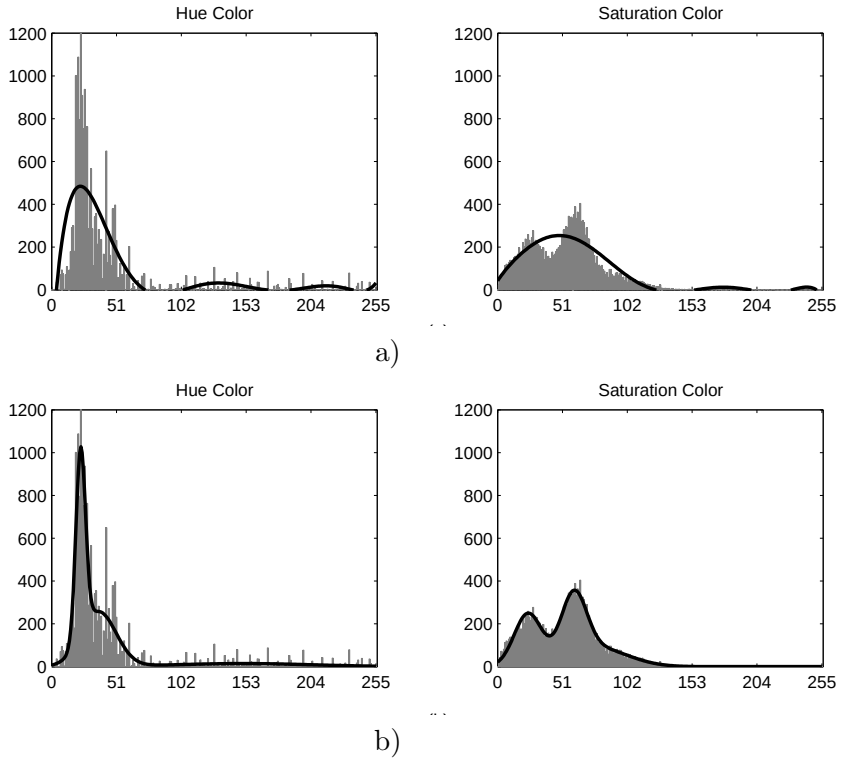


Figure 5.14: Example of histograms parameterization using 9 parameters: a) by a polynomial of order 8 and b) by 3 Gaussians

♯ of parameters	Gaussians		Polynomials	
	H color	S color	H color	S color
3	3214	1087	2401	1284
6	2010	830	2200	960
9	1537	534	2035	681
12	714	478	1871	539

Table 5.2: Parameterization error using Gaussian and polynomial functions

According to the presented experimental results, the Gaussian parameterization yields better histogram representations for the considered outdoor scenarios, when the number of parameter, k_{Hist}, is higher than 3. Consequently, in our work, the features extracted are the Gaussian parameterization, i.e., the weights c_i, means μ_i and variances σ_i, represented by,

$$o_t^{Hist} = FE(r_t) = \begin{bmatrix} \begin{bmatrix} c_1 & c_2 & \cdots & c_{k_{Hist}} \\ \mu_1 & \mu_2 & \cdots & \mu_{k_{Hist}} \\ \sigma_1^2 & \sigma_2^2 & \cdots & \sigma_{k_{Hist}}^2 \end{bmatrix}_{Hue} \\ \begin{bmatrix} c_1 & c_2 & \cdots & c_{k_{Hist}} \\ \mu_1 & \mu_2 & \cdots & \mu_{k_{Hist}} \\ \sigma_1^2 & \sigma_2^2 & \cdots & \sigma_{k_{Hist}}^2 \end{bmatrix}_{Sat.} \end{bmatrix} \tag{5.19}$$

where Hue and Sat. correspond to the Hue and Saturation components. The features extracted from the Parzen windows [94], return similar results and are computationally faster, which addresses a future work.

2D Histogram and Image Segmentation

Based on histograms it is possible to identify regions on the image with similar colors. We performed the bi-directional histogram along Hue-Saturation colors, as illustrated in Figure 5.15 and selected the k_{2Dhist} most significant colors $(Hue, Saturation)$. For each significant color, the smallest boundary-box that fits all the pixels with the same color define a region. The features extracted from each boundary-box are the width and height, the amount of pixels and the color, i.e.,

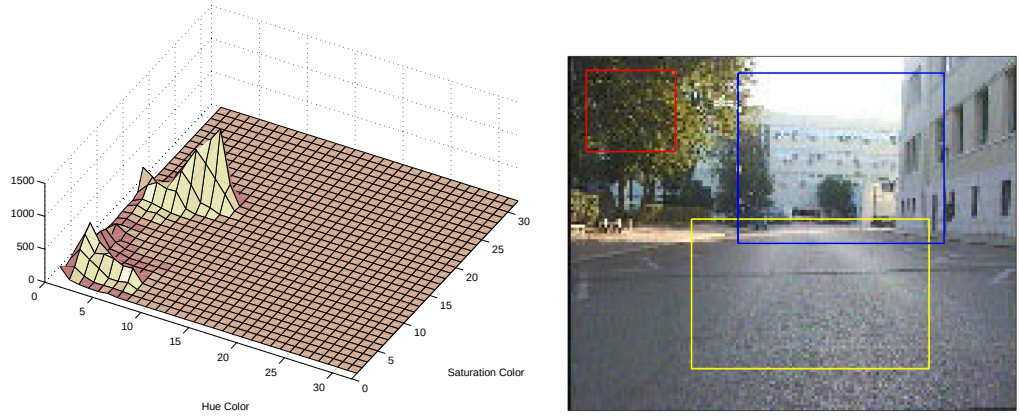
$$\begin{aligned} o_t^{2Dhist} &= FE(r_t) = [box_1\, box_2 \cdots box_{k_{2Dhist}}] \\ &= \begin{bmatrix} width_1 & width_2 & \cdots & width_{k_{2Dhist}} \\ height_1 & height_2 & \cdots & height_{k_{2Dhist}} \\ pixels_1 & pixels_2 & \cdots & pixels_{k_{2Dhist}} \\ color_1 & color_2 & \cdots & color_{k_{2Dhist}} \end{bmatrix}. \end{aligned} \tag{5.20}$$

The position of the boundary-box on the image is not recorded, since it is much dependent on the point of view [59].

PCA and ICA

A common approach to extract the essential information from images is the Principal Component Analysis (PCA) [56]. A similar technique where the components are orthogonal is the Independent Component Analysis (ICA) [123]. Both techniques extract a base, $B = \{B_1, B_2, \ldots, B_{k_{comp}}\}$, from a training set of images, where the features correspond to the projection of the images in that base. We will refer each B_i as a component of the base. The number of images that define the base, k_{comp}, is chosen according to the quality of reconstruction of each image and must be related with the diversity of the scenario. In this subsection, the selection of k_{comp} is disregarded.

Given the size of the images, Wachtler [123] proposes an implementation optimization



a) b)

Figure 5.15: Example of regions extraction: a) the bi-directional histogram (Hue and Saturation Colors) and b) the selected regions

dividing the images into sub-images. This is useful, since the original images present common areas (e.g., the ground, the sky), as illustrated in Figures 5.16 and 5.19. Therefore, each sub-image may represent a specific type of feature, i.e., a sub-image representing only the sky or only the ground or only trees. As a result, the performance of the representation given a new base of images increases significantly, when compared with the dimensions of the training set (e.g., 4 times if each image is sub-divided in 4 sub-images). This fact is shown in Table 5.3 and explained later in this subsection. The Figure 5.18 exemplifies a PCA-base of a set of images, where each image is subdivided into 16, as illustrated in Figure 5.19-c).

♯ of components	PCA			ICA		
	1 (no division)	4 (sub-images)	16 (sub-images)	1 (no division)	4 (sub-images)	16 (sub-images)
5	6.6	7.3	6.1	11.5	12.2	8.9
10	1.7	5.7	5.1	2.3	11.0	8.7
15	0	4.4	4.5	0	10.4	8.5
20	0	3.2	4.1	0	8.9	8.3
25	0	2.0	3.8	0	8.0	8.1

Table 5.3: Error of the image reconstruction using PCA and ICA

The projection of the training set into each base B provides different energy distribution. The PCA results condense the energy into the first components (usually the first 2 retain more than 90%), as exemplified in Figure 5.17, using the first 25 principal components of the base evaluated from the training set with the 12×16 images obtained by sub-dividing in 16 each of the 12 images of Figure 5.16.

The features used for the mapping procedure are the projection of the acquired images, r_t, on the base, B, or equivalently:

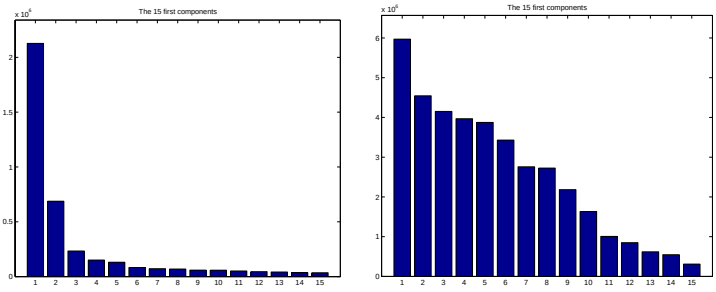
$$o_t^{PCA} = FE(r_t) = \left[< r_t, B_1 > \cdots < r_t, B_{k_{comp}} > \right]. \quad (5.21)$$

The features o_t^{ICA} are similarly extracted if the basis results from the ICA procedure. Both techniques, PCA and ICA, can be applied to the images in RGB or HSV format. However, Hue and Saturation are the most important as explained in Section 5.4.2.

Figure 5.16: A training set of images

The relationship between the basis, the number of components and the number of sub-images is non linear, as exemplified by the results in Table 5.3. This table presents the image reconstruction error using PCA and ICA, with 5, 10, 15, 20, 25 components. The columns correspond to the sub-divisions of the images (1-no division, 4,16-divides the image into 4 and 16 sub-images respectively, as illustrated in Figure 5.19) with a training set of 12 images, which is used to reconstruct each test image. For example, using this base (12 images) with no division, the reconstruction error is low when the number of components is close to 12. Using the same base with images divided into 4 sub-images (equivalent to 12 multiplied by 4), the reconstruction error is low when the number of components is close to 48. The error is an average for all pixels (each pixel changes between 0 and 255).

When the number of components in the basis increases, the error decreases. For instance, the reconstruction error is zero when the number of components is larger than 12 since the training set has 12 images. However, the reconstruction using 5 or 10 components and images divided into 4 sub-images provides an error larger than the one obtained with images divided into 16 or not divided. For a basis with more than 10 components the error decreases. This situation occurs because when the original images are divided into 16, the sub-images coincide with the ground, the sky or the buildings.



a) b)

Figure 5.17: L_2 norms of the basis functions using a) PCA and b) ICA

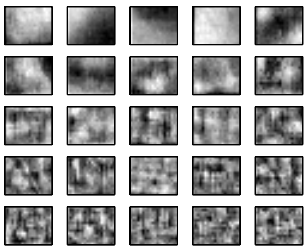
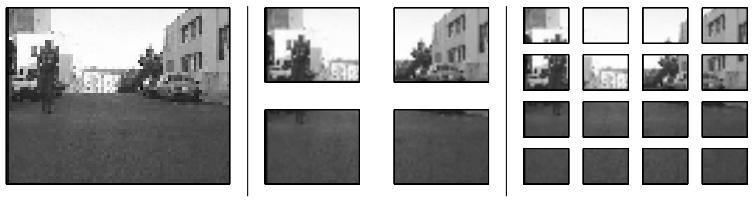


Figure 5.18: The first 25 principal components of the base shown in Figure 5.16, when each original image is divided in 16 sub-images

5.4.3 Feature Selection

As soon as features are extracted it is necessary to select the ones that will be used for mapping, this requiring a selection criteria. The quality of a feature for mapping purposes has to be analyzed along two different perspectives: time/space and correlation with other features as illustrated in the following example (see Figure 5.20). Consider that there are two types of features: "colors" and "geometric forms", and that the mobile robot navigates along three distinct places. If all the places are identified by the same color, the feature "color" is useless, regardless of the "geometric" information. If the two first places are "red" (the same value for the feature "color") and the third place is "blue", the feature "color" can identify some places, but the ambiguity still remains. In this case, if the geometric form is the same for the "red" places, but different for the "blue" place, the two features are redundant, i.e., the correlation between features is too high.

The feature quality explained before is based on a concept of features correlation that



a) b) c)

Figure 5.19: a) The entire image, b) image divided into 4 sub-images and c) divided into 16

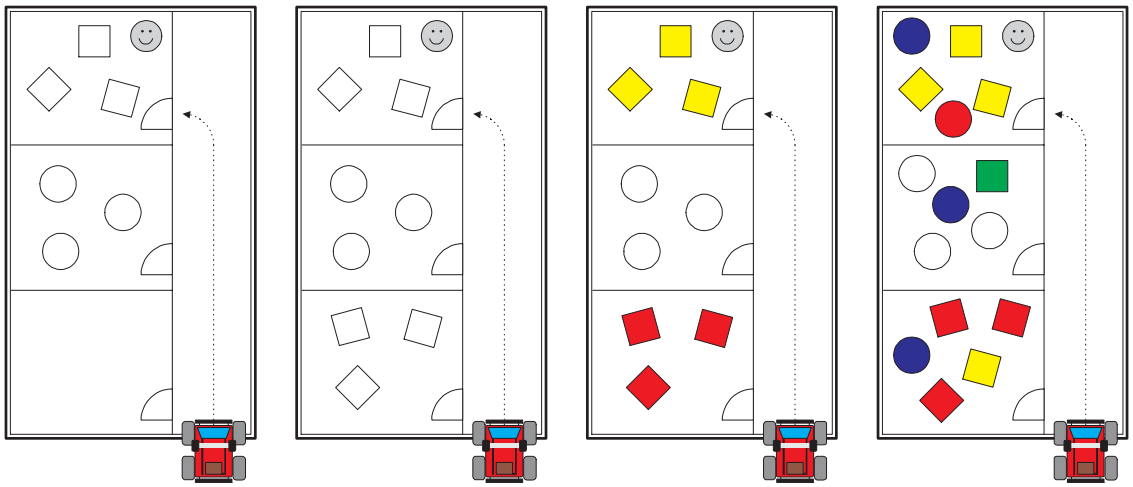


Figure 5.20: Four different situations in a scenario with different combination of features: a) two different geometric forms b) two rooms are represented by squares and c) geometric forms and colors and d) mixed

is formalized in the sequel.

Let the set of extracted features be given by

$$O = \{O^{Edges}, O^{Hist}, O^{2Dhist}, O^{PCA}, O^{ICA}\}, \tag{5.22}$$

where O^a corresponds to the features of type a extracted from $n+1$ vectors of rawdata,

$$O^a = \{o^a_{t_0}, o^a_{t_1}, \ldots, o^a_{t_n}\} \tag{5.23}$$

with $a \in \{Edges, Hist, 2Dhist, PCA, ICA\}$.
Define μ_{o^a}, μ_{o^b} as

$$\mu_{o^i} = \frac{1}{n+1} \sum_{t=t_0}^{t_n} o^i_t, \qquad i = a, b,$$

where a and b are different type of features.

Let M be the $D_a \times D_b$ matrix,

$$M = \frac{1}{n+1} \sum_{t=t_0}^{t_n} \left[o^a_t - \mu_{o^a}\right] \left[o^b_t - \mu_{o^b}\right]^T \tag{5.24}$$

with D_a and D_b such that $o^a_t \in I\!R^{D_a}$ and $o^b_t \in I\!R^{D_b}$.

The correlation between two different types of features, o^a and o^b, with $a \neq b$, is evaluated as

$$corr(o^a, o^b) \propto \sum_{i=1}^{D_a} \sum_{j=1}^{D_b} |M_{ij}|\,, \tag{5.25}$$

where M corresponds to the matrix defined in (5.24), with generic element M_{ij} given by,

$$M_{ij} = \frac{1}{n+1} \sum_{t=t_0}^{t_{lc}} \left[o_t^a(i) - \mu_{o^a}(i) \right] \left[o_t^b(j) - \mu_{o^b}(j) \right]^T . \tag{5.26}$$

For the map construction, the algorithm uses the lc lowest correlated features as they convey the highest degree of environment information achieved with the lowest number of features. The dimension of lc is selected according to the desired accuracy of the mapping.

The feature selection is important to avoid ambiguities between states, as illustrated in Chapter 6. The number of states is not only defined by the type of features but also by the length of the feature's vector. As described previously, the feature representation evolves several variables. For instance, the histograms can be represented by k_{Hist} Gaussian pdfs. The value of k_{Hist} defines not only the histogram parameterization but also the topological representation. Large values of k_{Hist} (or k_{2Dhist}, k_{Edges}, k_{PCA}, k_{ICA} for other types of features) leads to an increase in the number of states. However, the correlation between features remains unchanged.

In outdoor environments, people walking or moving objects are quite often present. These dynamic obstacles provide some observations, which results in noise filtered by the mapping algorithm and avoided in the navigation using the obstacle avoidance behavior. However, more important are the changes in the scenario, which require new feature selection. The feature selection procedure must be performed periodically to adjust the best features. When a specific feature is considered by the feature selection as useless, it has to be removed from the topological map. Thus, the mapping algorithm adjust the states according to the available features. If a new type of feature is added, the mapping algorithm adjust the states with a new element in the features vector.

The experimental results including the extracted features and the feature selection performance are presented and described in the Chapter 6.

5.5 Mapping Initialization

The contents of this chapter is divided in two main parts: the mapping algorithm (using the Dynamic EM) and the feature extraction and selection. The Dynamic EM is described before the discussion of the type of features. As in previous chapters, the topics are purposely sorted to simplify the presentation. Therefore, this sequence requires some assumptions. For instance, to explain the localization it was assumed that the map is already known. The same assumption was taken in the navigation algorithm. In both situations, the topological map is defined by states and by specific information to link all the states. This information is not only the orientation to travel between two states s_i and s_j, θ_{ij}, but also the transition probabilities, a_{ij}. These two different variables have to be estimated also in the mapping procedure, after knowing the states.

At this point of the thesis presentation, all the main procedures of topological navigation are known, i.e., the localization, the navigation and the mapping. Therefore, it

is important to understand how the entire process is implemented, in particular how it starts when the robot is switch on in an experience.

If no map is available, then it is not expected to assume an initial map (the assumption in the Chapters 3 and 4 was only for presentation purposes). Hence, the localization and navigation can not be implemented at a topological level. In this situation when there is no a priori knowledge, known as a cold initialization (technical definition commonly used in GPS, when a device is switched on without an initial position), it is important to structure the initialization. As illustrated in Figure 2.3, the first block to be implemented is the map building. However, the map algorithm requires the identification of the best features to represent the environment. This identification has to be done during a learning phase where the robot moves around. A motion random algorithm is implemented in the robot, since the environment is unknown. For instance, the robot is driven randomly during a period of time, while a large amount of rawdata is acquired by the sensors and converted into all possible features. Using a considerable amount of features, it is possible to evaluate the best type of features to build a topological representation. Using only the best features for the current scenario, a map is built. After this learning phase, the set of best features is updated if necessary. To start the localization and topological navigation, a topological map with a minimum contents is required. The threshold that corresponds to a map that has enough resolution/definition to start supporting the localization and the navigation procedures may be defined using the entropy, or defined by a user. This particular topic is not covered in the thesis.

With a minimum topological map but without the linkage information, i.e., the orientation to travel between states, θ_{ij}, and the transition probabilities, a_{ij}, only the localization can be addressed. Since the localization requires the knowledge of the parameters a_{ij}, they are initially considered as having a uniform distribution. During the localization the parameters a_{ij} and the orientations θ_{ij} are updated according to the methodologies described in Chapter 4. The topological map can be updated by the mapping algorithm simultaneously with the localization algorithm. The orientation between two states, θ_{ij}, is very important, to follow with the navigation procedure. During the map update procedure, the feature selection can also be accomplished. Even, if using only some types of features, all the rawdata acquired by the different sensors installed on the robot are saved to periodically evaluate the best features.

If, at least, a minimum map is already available, covering part of the environment (it is assumed that the best features and the linkage information are known), the initialization starts with some information. At this stage, all the three main blocks, localization, navigation and mapping, can be addressed simultaneously at different rates, as illustrated in Figure 2.3.

In summary, each time that the robot is switched on, it starts from zero (without a map) or with a previous map, which will be updated again and again during robot motion. The topological map is refined along several trajectories followed by the robot in the

environment. Simultaneously with the map building, the localization and the navigation can be addressed with different priorities. If the robot has traveled several trajectories, covering most of the possible positions in the environment (including different points of view of the scenario), it is suitable to reduce the number of times that the mapping algorithm is accomplished. The map building algorithm is a process with the highest time consumption and, with a good representation, the localization and the navigation become the main goals to accomplish the mission.

Chapter 6

Experimental Results

The thesis addresses the problem of mobile robot navigation in outdoor environments based on a topological approach, dealing with three main issues: environment representation, localization and navigation. Along the thesis, simulation results were presented, mainly for illustration and for understanding the issues involved in the problem, in particular for the localization in Section 3.4 and for the navigation in Section 4.4. In both cases, it was assumed that the map was already known. In Section 5.3, and using simulated features, the mapping algorithm results and its evolution is illustrated. The simulated experiments were carried out individually, to illustrate each problem at a time.

This chapter presents a set of experiments in real environments, combining simultaneously the problems of mapping, localization and navigation addressed along the thesis. The experimental results were obtained using the mobile robot ATRV-Jr displayed in Figure B.1 and described with more detail in Appendix B. The world robot's perception is grounded on features extracted from the sensing rawdata, acquired by different types of sensors (laser scanner, a ring of 17 ultrasound sensors, GPS, gyroscope, pan&tilt vision camera). The mapping algorithm retrieves the topological map based on the world robot's perception, one of the most important tasks along the thesis. Given the high level of abstraction using the topological representation, the map building algorithm is strictly dependent on the type of features, as previously referred in Section 5.4. The previous chapters presented simulation results. Therefore, the experimental results presented in this chapter are carried with real data, from witch the features are extracted to build a topological map.

The experiments presented in this chapter were carried out indoor and outdoor environments. At the indoor environments the experiments deal with features extracted from range sensors. The mapping algorithm was tested based on these features and, consequently, the localization and navigation algorithms were tested using the resulted map. The influence on the topological representation of having trajectories followed in reverse direction, i.e., similar trajectories yielding different observations, is discussed. At the outdoor environments, obtained in different scenarios (IST campus of Alameda, Lisbon and Palácio de Cristal, Oporto) mapping was supported in image information. It is

also depicted the potentiality of the information extracted from a gyroscope to retrieve a topological representation of the environment.

6.1 Indoor Results

This section presents a topological map of a traditional indoor environment, which is of office type. The same environment could be represented by different topological maps, if different features were selected to represent it. Some of these maps, efficient for robot navigation, are also understandable for humans. Hence, the type of features used in the first experimental results are the same as humans use to distinguish the same scenario.

To avoid complex features at this stage, the experiments are conducted using only range sensors: the laser range scanner and the ring of ultrasound sensors installed on the mobile robot. The type of features were based on free-area measured by the laser and the ultrasound sensors. The free-area could be characterized by the mean, the variance or other combinations of the free-area along different directions measured by the sensors. As an example, the laser range scanner installed at the ATRV-Jr (Sick LMS 200) acquire the distances to an object between 0^o and 180^o with 0.5^o of increment, while the ring of ultrasound sensors measure free-space along 17 different directions.

The vector of chosen features is described in (6.1), where the first two elements correspond to the sample mean of the values measured by the laser and ultrasound sensors and the third component is the variance of the measurements acquired in a 180^o laser scan. In (6.1) the parameters N and M, corresponding to the samples acquired by the laser and to the number of ultrasound sensors, take the values $N = 181$ and $M = 17$.

$$o_t \equiv f_t(r_t) = \begin{bmatrix} E[freearea\_laser] \\ E[freearea\_sonars] \\ Var[freearea\_laser] \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} r_t^{laser}(i) \\ \frac{1}{M}\sum_{i=1}^{M} r_t^{sonar}(i) \\ \frac{1}{N}\sum_{j=1}^{N}(r_t^{laser}(i) - \frac{1}{N}\sum_{i=1}^{N} r_t^{laser}(i))^2 \end{bmatrix} \tag{6.1}$$

The selected indoor scenario to test the mapping algorithm, is a traditional office type indoor environment, with a corridor and rooms. The robot moved from a large room (a laboratory) to a corridor. The laboratory is a common place containing chairs, tables and people walking and is larger than the corridor, as illustrated in Figure 6.1.

The mapping algorithm, supported on the Dynamic EM described in Section 5.2.2, was tested using the observed features in (6.1) acquired during the travel from the middle of the room to the corridor. The selected features do not contain metric information of the robot's position. The range rawdata recorded during the trajectory is displayed in Figure 6.2-a) using the odometry only for illustration and to simplify the data visualization. With no map at the beginning, the navigation algorithm was applied using only the obstacle behavior, following a free direction.

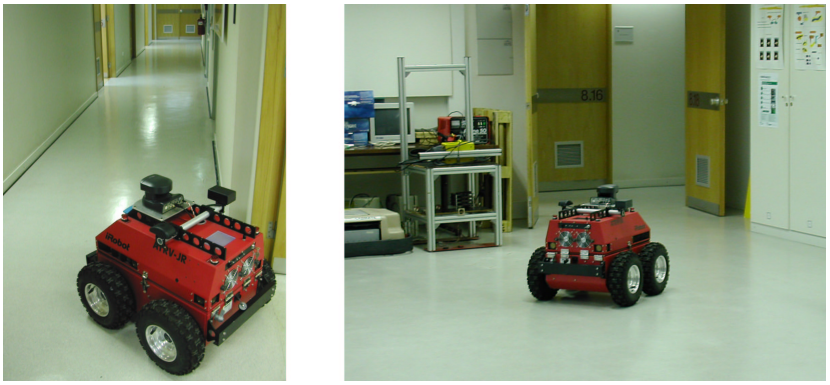
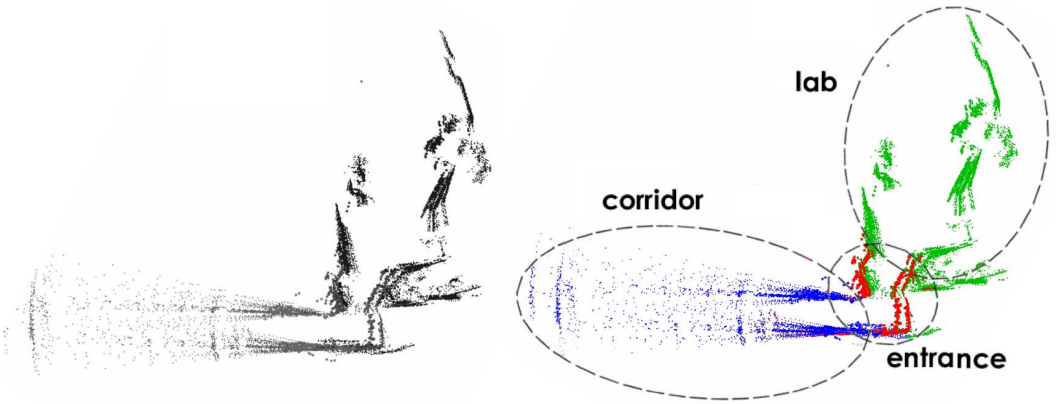


Figure 6.1: An indoor scenario, defined by a corridor and a large room



a) b)

Figure 6.2: a) the raw range laser and ultrasound sensor measurements acquired in the environment and b) the states that compose the topological map

Setting low accuracy to the mapping algorithm, the result is a topological map with three states, corresponding to the room and the corridor, as expected, and an additional state that corresponds to the transition between them, defined as an entrance. Figure 6.2-b) represents the measurements of the laser and ultrasound sensors with three different colors, corresponding to each state. The ellipses plotted in Figure 6.2-b) are only for understanding the different states that compose the topological map and have no statistical meaning. The odometry was only used to record the location where each measurement was acquired and to display the places corresponding to each state. A place corresponds to a physical location where the robot acquires measurements.

The states were distinguished based on the differences detected on the free-area, which is represented by Gaussian pdfs, i.e., by its mean vectors and covariance matrices, as shown in Table 6.1.

The state corresponding to the room, and herein denoted as lab, results with a large free-area measured by the laser (0.7472) and ultrasound sensors (0.9353) and with high

variance (0.9590), which corresponds to the noise created by thinner objects as tables, chairs or even people. The corridor has a free-area not so large, but with a low variance (0.7968), when compared with the other states. The entrance, is still defined by high variance (0.9713), but the values of the free-area are different from the laser (0.6732) to the ultrasound sensors (0.2880). This is explained by the position of the ultrasound sensors. The laser scanner measures only the free-area in front of the robot (180º in front), while the ring of 17 ultrasound sensors measures the free-area values around the robot covering 360º around the robot.

While free-area translates the amount of free space measured by range sensors, the variance of free-area gives an idea of the noise. However, the ultrasound sensors may retrieve wrong measurements give the electronics limitations. The ultrasound sensors return acceptable values when the target object is in a small incidence angle. Even under this condition, the measurements may be wrong given the type of the object. Therefore, places as the entrance (with different walls) may be identified with large variance, since two near ultrasound sensors may retrieve completely different measurements.

Another important issue of ultrasound sensors lies on their position, since they cover 360º while the laser only covers 180º. The used features are the mean of the free-area, which are dependent to the point of view. When the robot is in the entrance, the free-area around the robot is even smaller then in the corridor.

corridor				entrance				lab			
μ	R			μ	R			μ	R		
0.4228	0.0029	0.0072	0.0012	0.6732	0.0158	0.0080	0.0022	0.7472	0.0224	0.0003	0.0062
0.3362	0.0072	0.0188	0.0036	0.2880	0.0080	0.0065	0.0006	0.9353	0.0003	0.0005	0.0000
0.7968	0.0012	0.0036	0.0032	0.9713	0.0022	0.0006	0.0011	0.9590	0.0062	0.0000	0.0027

Table 6.1: The Gaussian parameters of the topological map presented in Figure 6.1

At this point, topological maps based on multi-dimensional Gaussians (or, on a sum of multi-dimensional Gaussians) can only be represented by plots if the vector's size is less or equal to 3, as exemplified in Figure 6.3 (mean and variance of free-area measured by laser and mean of free-area measured by ultrasound sensors). The different states, corridor, entrance and laboratory are painted using different colormap, "HSV", "Pink" and "Hot", respectively.

When the feature vector's size is greater than 3, the states of topological maps can be represented by tables containing the means and the covariance matrices. In the present case, since only three types of features are used, it is possible to plot the topological map as a set of ellipsoids, where the referential axis correspond to the three type of features, as illustrated in Figure 6.4. The ellipsoid is a symbolic representation of the probability that an observation belongs to a particular state. Each ellipsoid, centered in the mean, is plotted with a value of probability 95%, illustrating the principal combinations of the features. It is important to stress that some states present some intersections (the example in Figure 6.4-b), between state 2, the entrance and state 3, the lab), i.e., using only 2 types of features it is quiet difficult to distinguish all the states.

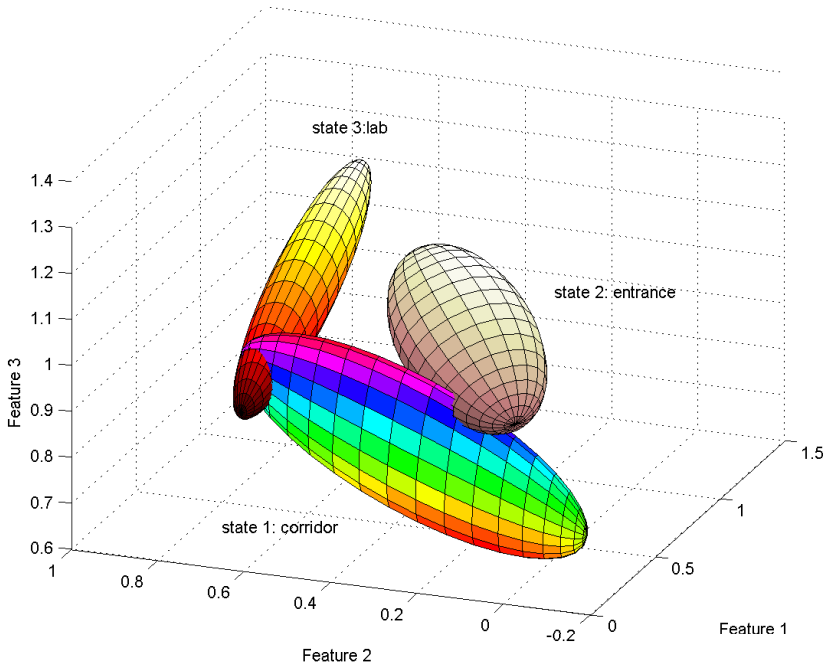
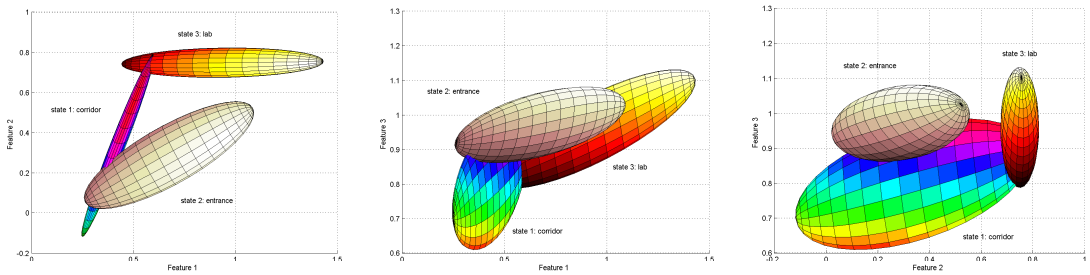


Figure 6.3: The states of the topological map are represented in a 3D space, where the axis correspond to the free-area measured by the laser and ultrasound sensors and the variance

It is also possible to plot the topological map along only two type of features, projecting along the axis of the third feature.



a) b) c)

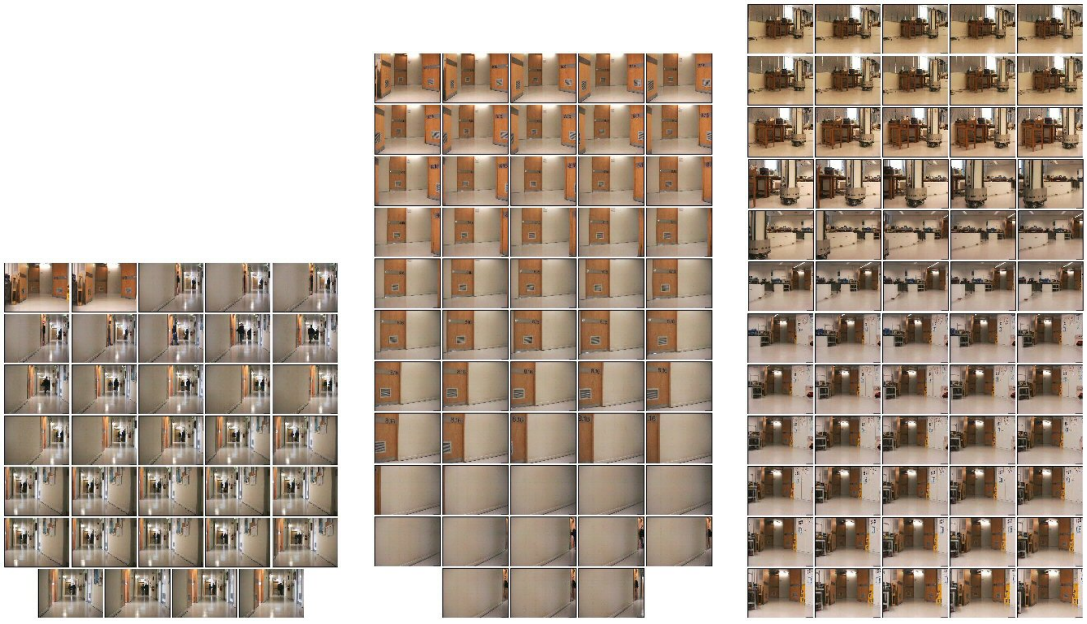
Figure 6.4: The projection of the topological map into the axis, combining 2 different type of features

In Figure 6.4-a), it is visible the importance of Feature 2, the free-area measured by the ultrasound sensors, to identify the lab and the entrance. The corridor is not understandable with only the Features 1 and 2, i.e., the free-area is not enough to identify that state. In Figure 6.4-b), it is difficult to distinguish all the states, with the ambiguity being large between the lab and the free-area entrance if using only the free-area and the free-area variance measured by the laser. The corridor is well identified by the Features 2 and 3, as shown in Figure 6.4-c).

The topological map presented in Figure 6.2-b), resulted from an experience carried in

an indoor environment. It does not contain metric information neither features extracted from a vision camera. The metric information is only used for plotting the rawdata to illustrate the scenario. During the experience, the images were grabbed simultaneously with the laser and ultrasound sensors data. The camera is mounted on the top of the robot pointing in the front direction (see Figure B.1). After the topological map construction, the images were classified according to each state (obtained exclusively based on range data). This classification is shown in Figure 6.5.

The image acquired at time instant t is put in correspondence with the state where the information extracted from the range-sensors belongs. The images are displayed, for increasing values of t, along rows from left to right and from top to bottom. By inspecting Figure 6.5, it is reasonable to accept that the images are properly clustered, except two images in State 1 which, according to human reasoning, should belong to State 1. This occurs when the robot acquires some free-area that corresponds to another state.



a) b) c)

Figure 6.5: The images associated to the a) State 1 (corresponding to the corridor), b) State 2 (corresponding to the entrance) and c) State 3 (corresponding to the lab)

The images were clustered according to each state, based only on range sensor data and no vision information. The resulting clustered is acceptable and understandable by a human to classify and distinguish the three states. By inspection, the third state, the lab, may be also decomposed in two possible new states. For instance, one state when the robot is observing other robots (the first half of images in Figure 6.5-a)) and another state when it observes the door (the second half). However, for the topological map to exhibit this more refined environment characterization, more information than the rawdata extracted from solely the range-sensors should have been available, as discussed in Section 5.4.

At this point, it is important to evaluate how the localization algorithm deals with the

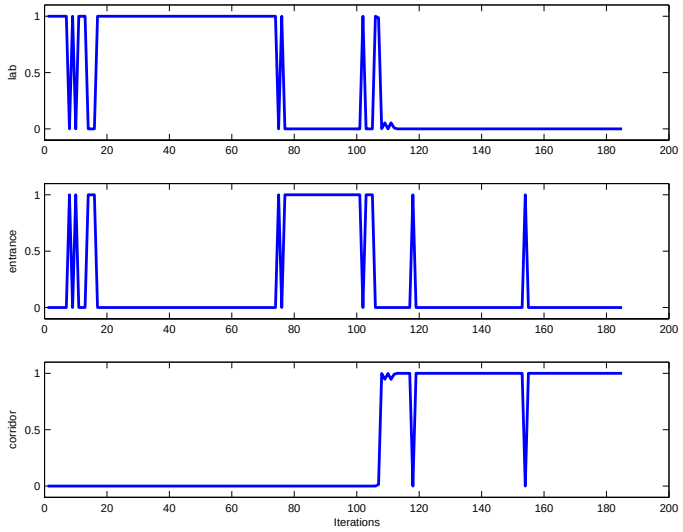


Figure 6.6: Localization results based on the topological map shown in Figure 6.2

retrieved topological map. For this indoor environment, the first experiment was obtained just activating the navigation procedure, to follow the trajectory lab→entrance→corridor (State 3→State 2→State 1). The resulted trajectory is similar to the one the robot followed to compute the topological map. As shown in Figure 6.6, the localization estimates the robot position in terms of states, dealing with the uncertainty. The localization results present some uncertainty at the beginning between the lab and the entrance. This situation occurs again in the transition between the entrance and the corridor, mainly caused by the noise on the sonars. Now a question arises: what happens if the robot follows the same trajectory but in opposite direction and using the previously obtained topological map?

It is imposed a trajectory, illustrated in Figure 6.7, where the robot had to follow the sequence lab→entrance→corridor→entrance→lab (State 3→State 2→ State 1→State 2→ →State 3). Note that the first part of this trajectory (defined in terms of states), coincides with the previous one. The localization results are similar to the previous experiment, during the first 3 states, as shown in Figure 6.8, but when the robot comes back to the lab (the robot has to reverse direction at iteration 190) the uncertainty in the localization is large between the lab and the entrance. This is caused by the laser, which covers only 180º in front. The robot may be placed at the same position but with a different orientation retrieving different rawdata and consequently, different features. In this case, when the robot rotates 180º it automatically observes the State 3, the lab.

6.2 Outdoor Results

The next experience was targeted to test the topological representation of outdoor scenarios using features extracted from intensity color images, acquired by the image camera mounted on the mobile robot. The topological approach is oriented towards outdoors, large and unstructured environments. The proposed scenario is the Campus of IST (Instituto

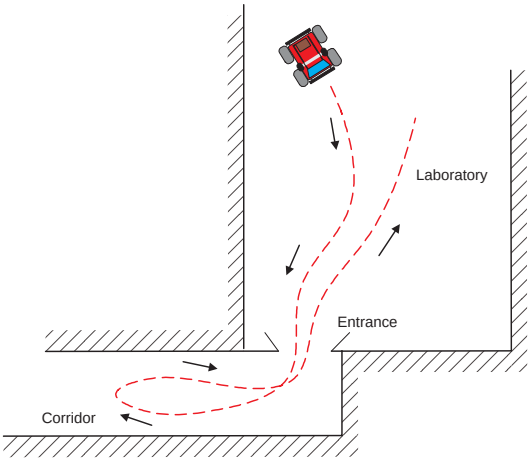


Figure 6.7: The trajectory followed by the robot

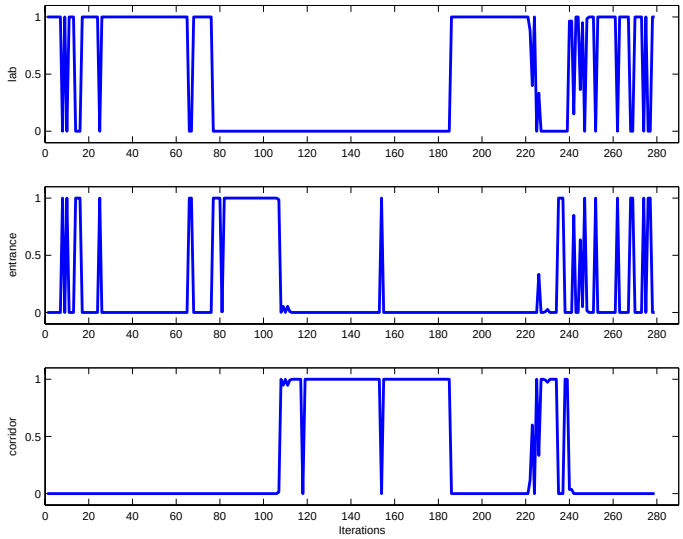


Figure 6.8: Localization results based on the topological map shown in Figure 6.2, with a different trajectory

Superior Técnico) at Lisbon - Alameda illustrated in Figure 6.9. The robot acquired the vision rawdata, while it was navigating around the Central Building and simultaneously computing the topological map.

Since no map was available at the beginning, the navigation algorithm during the initial mapping acquisition phase was proposed to guide the robot to follow some via-points (defined by latitude and longitude coordinates), using the behaviors defined in Section 4.3 and an orientation estimation obtained through a procedure described in Appendix B.2. The algorithm to estimate the current robot's metric location, is based on Extended Kalman Filter, knowing the kinematics, the sensors' model and the respective measurements (GPS and compass). The main goal of this behavior guidance procedure is to have no tele-operation and to let the robot moving with no human intervention, to test

mainly the mapping algorithm.

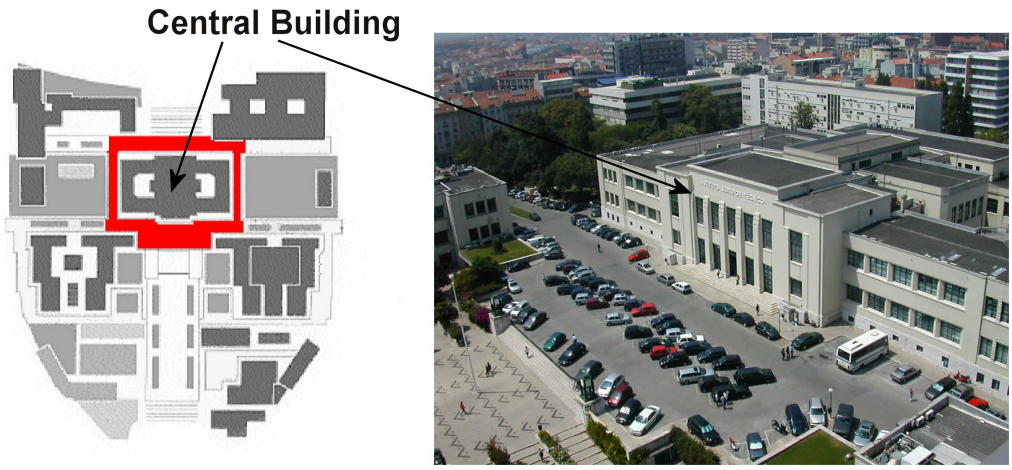


Figure 6.9: Outdoor scenario at the IST Lisbon campus (Alameda)

The robot moved along a distance of approximately 400 meters and acquired vision images during 1586 iterations. The images were converted in features using the procedures described in Section 5.4.2. The trajectory is illustrated in Figure 6.10-a). It is important to underline that the features reduce the size allocated to record all the rawdata but retain the essential information. As illustrated in Table 6.2 that represents the compression from rawdata to features, the edges and histograms (with colors or 2D), reduce significantly the space required from an hard-disk, while PCA and ICA, still using images (even with reduced dimensions), retain more information.

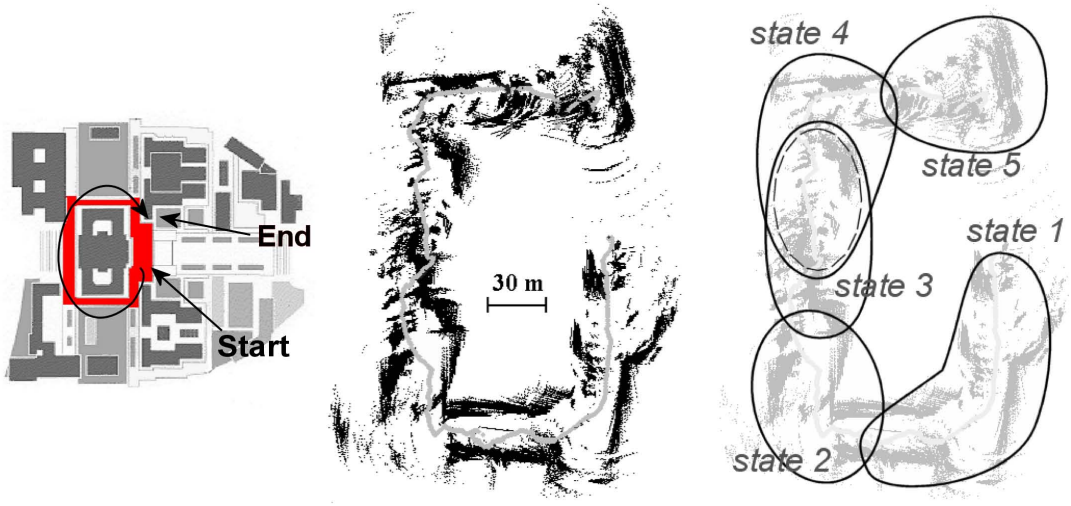
Feature extraction yields data compression as represented in Table 6.2 for a set of 1586 images (JPG) for each different experiment. The storage space required for the images is approximately 20 Mbytes while only 1 Mbytes are required for all features mentioned above. The edges, histograms and 2Dhist yield higher compression than PCA or ICA, since the two last features require a base of images.

edges	histograms	2Dhist	PCA	ICA
98.83%	98.94%	99.11%	85.80%	85.80%

Table 6.2: Compression from observations to features

At this point, it is necessary to evaluate the resulted topological map, the quality of the information retained by the feature extraction procedure and the selection of the best features to represent the scenario. In outdoor environments the best features are extracted from images, given the other sensors limitations in outdoor environments, as the range sensors. However, the data acquired by the other sensors installed on the robot (range sensors, GPS or inertial sensors) is also stored. The data acquired by the range sensors, mainly from the laser range scanner (since ultrasound sensors have a small range for large outdoor environments), is presented in Figure 6.10-b), using pose estimation by GPS and

Kalman Filtering, described in Appendix B.3, only to illustrate the robot's position in the scenario. The Central Building of IST is located in the center of the image in Figure 6.10-b), surrounded by trees, cars, people walking and other buildings. We point out that in this experience we only used features extracted from the vision camera. However, it would have been possible to extract features from laser range data or even from inertial sensors. The resulted topological map using histograms (of colors or hist2D) and PCA is presented in Figure 6.10-c). These features were selected according to the human perception: a scenario commonly identified by colors and specific images (as buildings, trees, grass). At this point, the experience could be accomplished with other type of features, just to justify that is necessary a feature selection procedure.



a) b) c)

Figure 6.10: The resulted topological map: a) the followed trajectory, b) the laser measurements acquired in the environment and c) the 6 states that compose the topological map using histograms an PCA

The states 1, 2 and 5 are well defined, while states 3 and 4 present a high intersection in the metric space, i.e., there is an uncertainty between two states that can represent the same physical area. The quality of the information included on features is not enough to create a state between states 3 and 4, leading to an ambiguity between these two states. As referred in Section 5.4.3, increasing the number of features does not necessarily imply an information improvement to represent what is observed. Consequently, it is necessary to evaluate the quality of features and to choose the low correlated.

As described in Section 5.4.3, one possible way to find the best features is to measure the correlation between features and to choose the lowest correlated ones. To access the performance of this procedure, experiments were carried out in three different scenarios:

- Scenario 1 - a combination of buildings, trees, cars, walking people and diversity of

light conditions,

- Scenario 2 - a garden surrounded by trees and grass, and
- Scenario 3 - a parking place (a relatively structured area) mainly with cars and buildings.

The different scenarios are illustrated in Figure 6.11. All these scenarios have similar spots, like trees, cars, buildings and people walking. These spots are placed differently in each scenario, which leads to different correlation between the features.

a) b) c)

Figure 6.11: A preview of the three scenarios a) Scenario 1 - the IST central area (around the Central Building), b) Scenario 2 - a garden and c) Scenario 3 - a parking place (scenario 3)

The features used for the correlation analysis (described in Section 5.4) are vertical edges, the Hue/Saturation-colors histograms parameterization (using 3 Gaussians pdf), 2D histograms (the first 4 boundary-boxes), the PCA (images subdivided in 14 sub-images and building a base with 15 components) and ICA.

As expected, the correlation differs between experiences and features, as shown in Table 6.3. The edges have low correlation with other features in Scenarios 1 and 3, since the buildings contain mainly vertical edges (independent of the colors), while in Scenario 2, the trees and the waved terrain provides rough edges. The correlation between PCA and ICA is high, mainly in Scenario 1 and Scenario 3, since the basis of PCA and the basis of ICA present similar images. In Scenario 2, the correlation between PCA and ICA is not so high given the irregularity of the ground and given the fact that the sky is never observed. These facts are some of the most important principal components of the basis in Scenario 1 and Scenario 3. In Scenario 2, the features 2Dhist and histograms are still correlated, which could be caused by the correlation between PCA and ICA. This fact occurs since the boxes of similar colors coincides with some image components. According to these results, histograms and PCA or edges and ICA are the less correlated features in Scenario 2 or Scenario 3. Histograms and ICA is also a good combination for the last scenario.

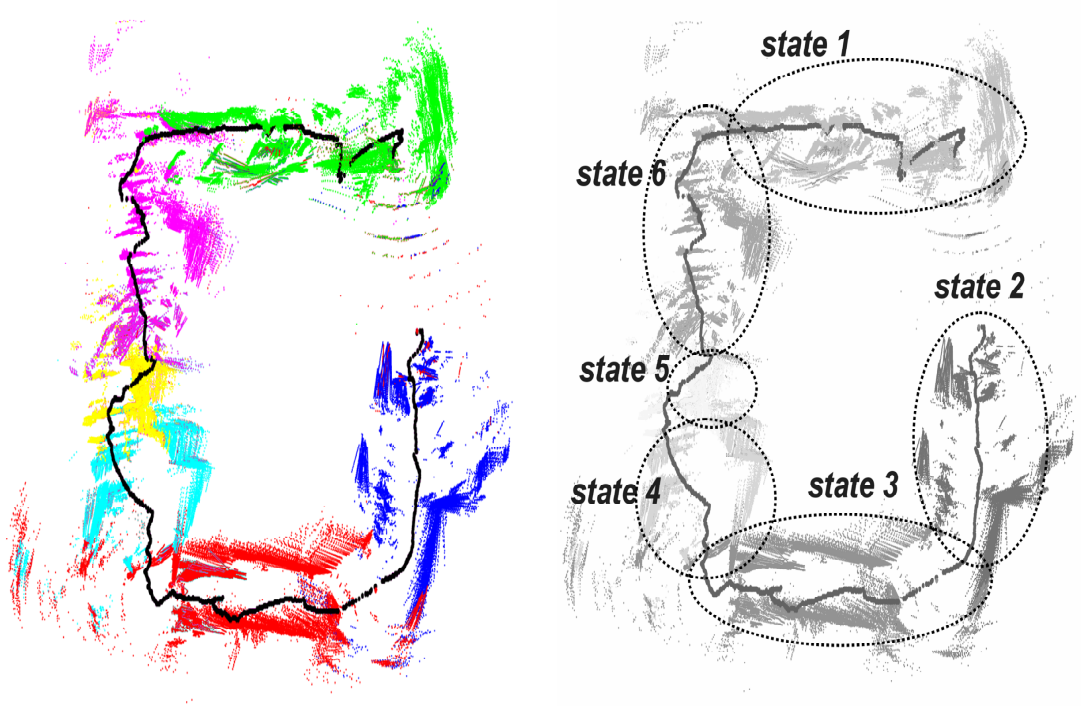
From the results presented in Table 6.3, the edges and ICA, histograms and ICA or edges and histograms are the lowest correlated features to build a map in Scenario 1. The

	Scenario 1	Scenario 2	Scenario 3
edges & histograms	0.049	0.513	0.265
edges & 2Dhist	0.194	0.442	0.226
edges & PCA	0.189	0.474	0.409
edges & ICA	0.013	0.288	0.074
histograms & 2Dhist	0.138	0.381	0.569
histograms & PCA	0.237	0.236	0.110
histograms & ICA	0.033	0.298	0.108
2Dhist & PCA	0.213	0.472	0.296
2Dhist & ICA	0.058	0.315	0.093
PCA & ICA	0.832	0.452	0.861

Table 6.3: Features correlation in the three different scenarios

histograms and ICA or histograms and PCA, but not the three simultaneously (since PCA and ICA are high correlated) should be used to build a map in Scenario 2.

Based on the results obtained from the feature selection, the histograms and edges were adopted as one of the best features combination for Scenario 1. As suggested in Chapter 4, one of the procedures to estimate the orientation between two states is given by the states, when the orientation is also a property. So, the orientation was also included on the features: histograms, edges and orientation.



a) b)

Figure 6.12: The Topological map compiled by the algorithm using the selected features (histograms and edges): a) the Laser and ultrasound sensors measurements acquired in the environment and b) the states identified by ellipses

The same rawdata, acquired in the previous experience around the central building of IST campus, was used to extract the selected features (histograms and edges). The mapping algorithm computed a new topological map, as illustrated in Figure 6.12, containing six states. The ambiguity of the previous experiment (using the features histograms and PCA) was removed, since the topological map identified a new state (State 5) according to the selected features (histograms and edges). The states resulted from both experiments are not exactly related, i.e. the State 1 of the first experiment is not related with the State 1 of the second experiment with different types of features. However, there is a slighter relation between some states. For instance, the State 1 in Figure 6.10-c) and the State 2 in Figure 6.12 are not the same, but these states result from the observations acquired in similar places.

The topological map improvement is illustrated in Figure 6.13 in 9 different places along the trajectory. The interval between two successive images is around 170 acquisitions. In the first image, there are still poor information for a topological representation, resulting on two states. As soon as more information is acquired, the mapping algorithm updates the representation, as illustrated in images of Figure 6.13. The first state includes the two old states and a new one is created. After the second image, it is visible that the state represented by the blue color is completely defined. The following image shows that a third state is imminent. After 850 acquisitions (two more images in Figure 6.13), the topological map has 4 different states. In the 6th and 7th image, an uncertainty between the two last states (illustrated by the cyan and yellow colors) are visible. Additional 170 acquisitions are required to clarify the existence of two other states, removing the previous ambiguity. The last image corresponds to the topological map after 1500 acquisitions, also shown in Figure 6.12.

The Figure 6.14 displays vision images representing each state of the topological map. The first three states are mainly defined by vertical edges (these edges are defined by buildings). The last three states, also have vertical edges, but are mainly characterized by the color segmentation. It is notable, the difference between State 5 and its neighbor states (4 and 6), by a higher temperature color, which explains how the histogram colors remove the ambiguity in Figure 6.10.

Using the topological map constructed from the last experience represented in Figure 6.12, the localization algorithm described in Chapter 3 was tested. It is assumed that the map remains the same and that there is no significant changes requiring a map update. To localize the robot in the topological map it is also necessary to acquire and extract the same type of features used in the mapping algorithm. The robot followed a similar path around the central building while acquiring the rawdata and extracting the features.

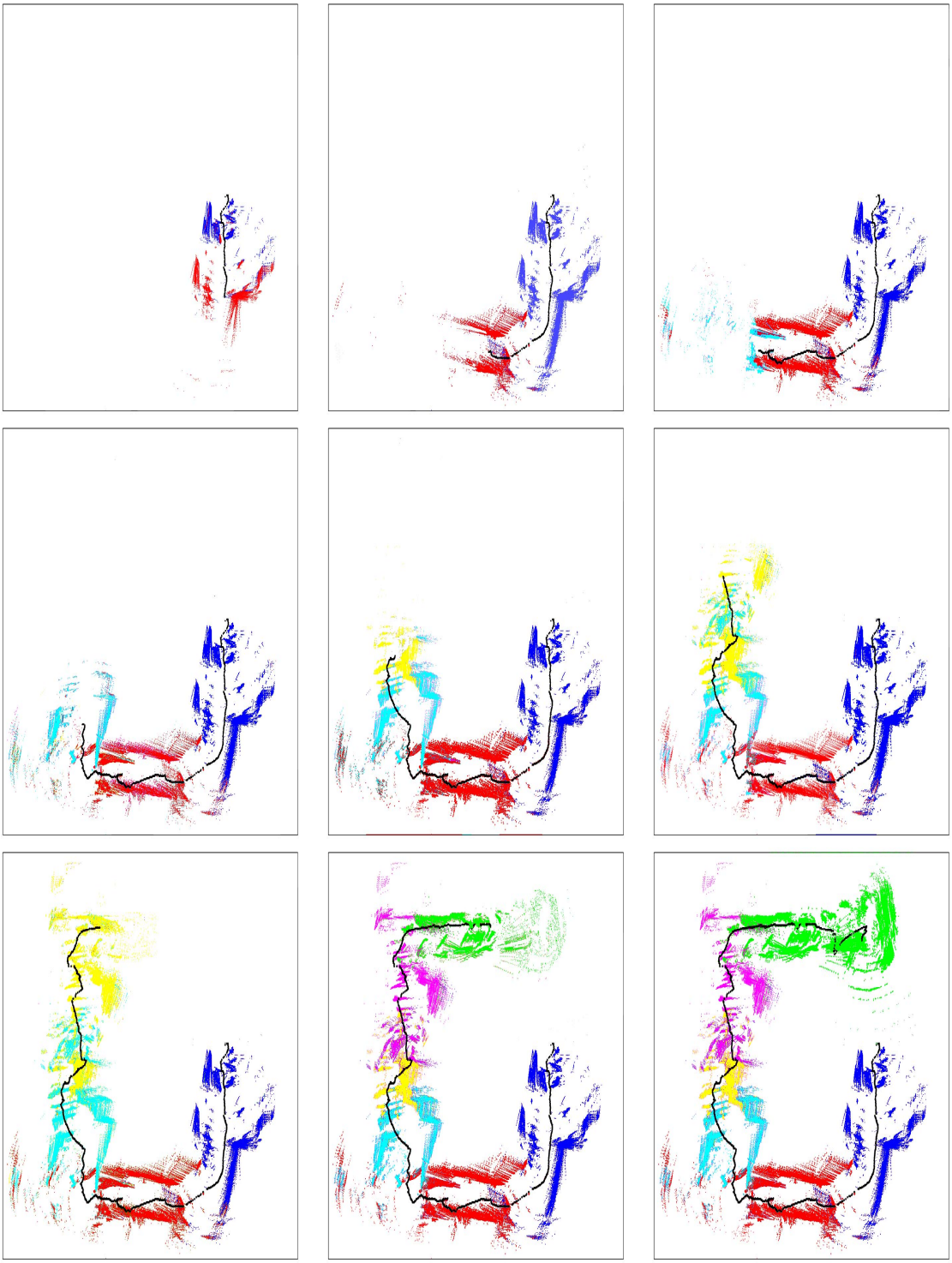
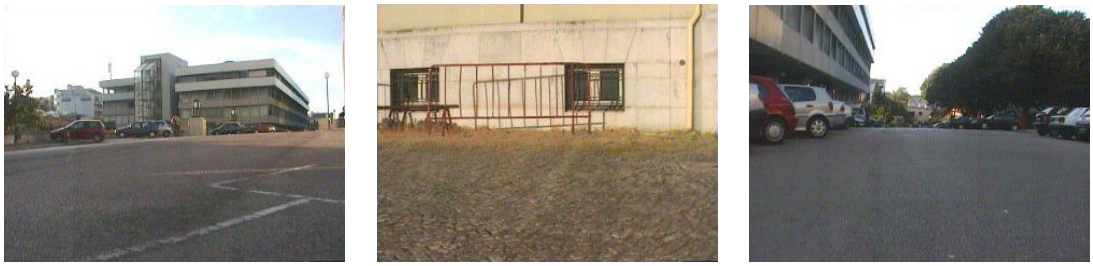


Figure 6.13: The topological map improvement illustrated in 9 different places along the trajectory, with an interval of 170 acquisitions

State 1 State 2 State 3



State 4 State 5 State 6

Figure 6.14: Each state of the topological map is illustrated by a randomly selected image

The experimental results, following the principles explained in Section 3.4, are presented in Figure 6.15, with the probabilities of the robot to be localized in each state, given the observed features along time. The localization results present some uncertainty between states 2 and 3 in the beginning of the trajectory, quickly solved after few iterations remaining in State 2 during more than 200 iterations. The following State is 3 and there is a period of uncertainty between states 3 and 4, lately solved to the last one. This is also verified in Figure 6.15, when the mapping algorithm shows a small oscillation between states 3 and 4, visible by the blue and red colors. This situation reflects not only the feature selection issue, but also the consequence of moving objects (for instance, cars and people walking). The presence of obstacles forces the robot to change its orientation, observing the features of State 3.

The localization algorithm results present less uncertainty in states 5, 6 and 1. The peaks in the localization probabilities occur when the robot is deviated by the navigation behavior to avoid a detected obstacle. An example occurs close to the iteration 800 between states 5 and 6, as shown in Figure 6.15. Therefore, the robot changes its orientation, observing some features of the previous and/or the next state. When this occurs, the localization updates the robot position in the topological map and the navigation also updates a new sequence of states and commands. The target goal remains the same, so the navigation algorithm returns a new sequence ending in the same goal state. The difference lies on the first states of the sequence, since the robot had been repulsed by temporary obstacles.

Along the path and based on the localization results on the topological map, it is also possible to estimate the transition probability between states used in the Markov Model,

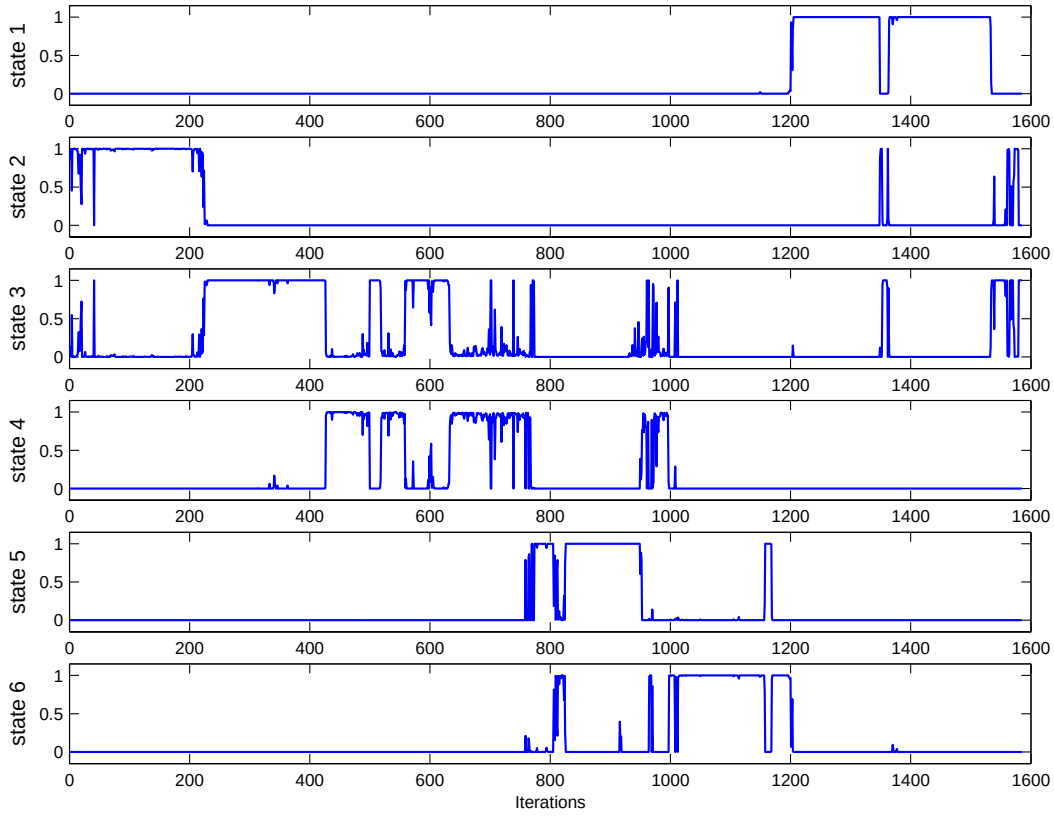


Figure 6.15: The localization evolution on the topological map, i.e., the probability of the robot be placed in each state

as described in Subsection 3.2. The results of the transition estimated probabilities are shown in Table 6.4. This is a rough estimation, given the few number of tests. For a better estimation it is necessary several experiments in the same scenario. The robot has a high probability to remain in the current state, for every state of the topological map. If the robot moves to another state, it is more probable to jump to the next or to the previous state. However in State 1 it is also probable to jump to the State 3, or from the State 6 to the State 3, given their proximity in the features space, i.e., the states present similar Gaussians. The noise present in the State 3 is similar to the last experience, caused by the moving obstacles.

a_{ij}	1	2	3	4	5	6
1	0.9906	0.0031	0.0032	0	0	0.0031
2	0.0042	0.9541	0.0417	0	0	0
3	0.0028	0.0249	0.9282	0.0276	0.0055	0.0110
4	0	0	0.0397	0.9495	0.0072	0.0036
5	0	0	0.0056	0.0167	0.9556	0.0221
6	0.0057	0	0.0142	0.0047	0.0189	0.9575

Table 6.4: Transition estimated probabilities between states

It is also possible to estimate the orientation between states, θ_{ij}, as explained in Sec-

tion 4.2. The estimated values are shown in Table 6.5. The predominance of signal minus results from the selected orientations computed by the navigation behaviors, given the obstacles and the target orientations being clockwise. For instance, when the robot is localized in the state s_i and the next goal is the state s_j, its current orientation must be increased (or decrease, according to the signal) by θ_{ij}. The robot should rotate clockwise for negative values of θ_{ij}. There are two situations, where the directions are not correct, from state 5 to state 3 and from the state 4 to the state 3, also a consequence of the same issue explained above. However, the orientations are a good approximation when looking to the topological map illustrated in Figure 6.12-b).

θ_{ij}	1	2	3	4	5	6
1		-37°	-45	-	-	50°
2	-120°		-20°	-	-	-
3	-129°	-78°		-64°	-86°	-88°
4	-	-	-9°		-24°	-28
5	-	-	-15°	-103°		-5°
6	-105°	-	-40°	170°	150°	

Table 6.5: The orientation angles between states

Since it is possible to determine the orientation as a feature, it is important to test if it is possible to compute a topological map using only orientation information returned by a gyroscope. The first experiment was obtained at the garden of Palácio de Cristal in Porto, where the robot followed a predefined trajectory (no navigation at this level), climbing a ramp (near 100 meters at a positive inclination) and after rotating 150° and driving along a horizontal surface during 40 meters. The information recorded by the gyroscope: the angles Roll, Pitch and Yaw, as illustrated in Figure 6.16, define the vector of features. The information acquired during 1000 iterations is plotted in Figure 6.17.

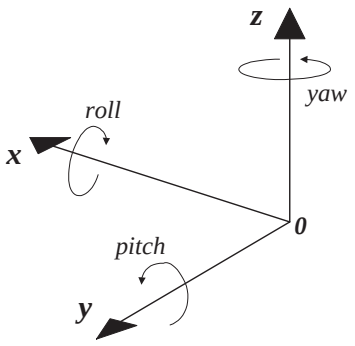


Figure 6.16: The three angles along the referential axes

The resulted topological map is defined by two states, where the values are shown in Table 6.6. The Pitch and Yaw are the features that best identify each state, as expected given the results shown in Figure 6.18. The State 1, which is characterized by the observations acquired until the iteration 500, presents an inclination of 4° while State 2 is close to

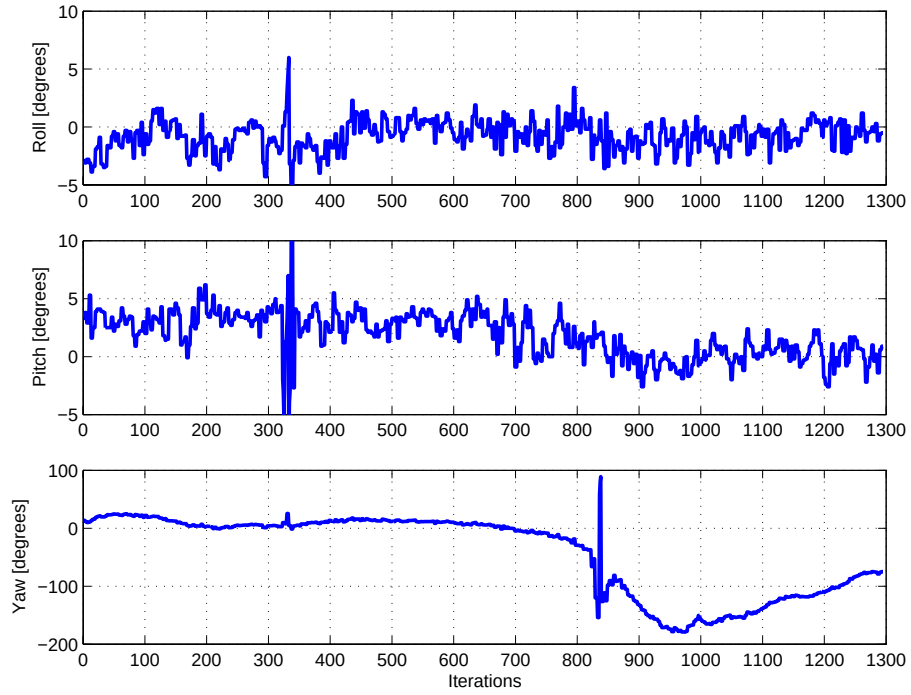


Figure 6.17: Experiment 1 - The gyroscope information recorded during the trajectory

horizontal. The states are illustrated in Figure 6.20. It is visible few peaks around iteration 350 given the type of the terrain. The presence of small rocks on the ground implies some noise on the roll and pitch measurements. At iteration 830, a temporary obstacle forced the robot changing its orientation, which is reflected on the yaw measurements. The peak is increased only in the plot of Figure 6.17 given the boundaries of the orientation, i.e., when the orientation assumes an absolute value higher than 180º, it is subtracted 360º inverting the signal. For instance, an orientation equals to 193º is converted to -167º; an orientation equals to -212º is converted to 148º.

State 1 State 2

Figure 6.18: The state values of the resulted topological map using gyroscope features in the first experience carried out at Palácio de Cristal

The second experience carried out at Palácio de Cristal, was performed in the same type

	State 1				State 2			
	μ	R			μ	R		
Roll	-0.7353	0.0001	0.0000	0.0010	0.6516	0.0003	0.0001	0.0003
Pitch	0.4396	0.0000	0.0002	0.0033	3.5980	0.0001	0.0001	0.0000
Yaw	258.0	0.0010	0.0033	3.0710	-11.016	0.0003	0.0000	0.0039

Table 6.6: The state values of the resulted topological map using gyroscope features

of environment. The results show that is still possible to identify different states at different terrains using only, and again, the gyroscope data. The robot followed a trajectory with two types of terrain, one covered by sand and another with concrete pavement. This fact is understandable by the measurements present in Figure 6.19, even with some noise. There is a ramp with a significant slope (around -16°), as shown in Figure 6.19-Pitch, between iterations 450 and 750. The noise is higher in the last iterations (over 750).

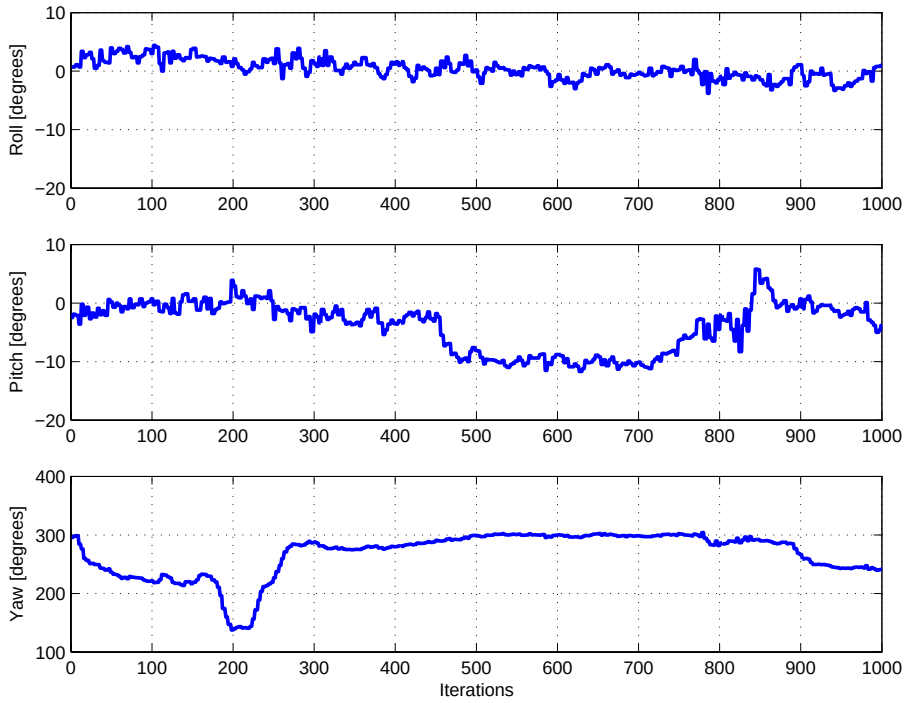


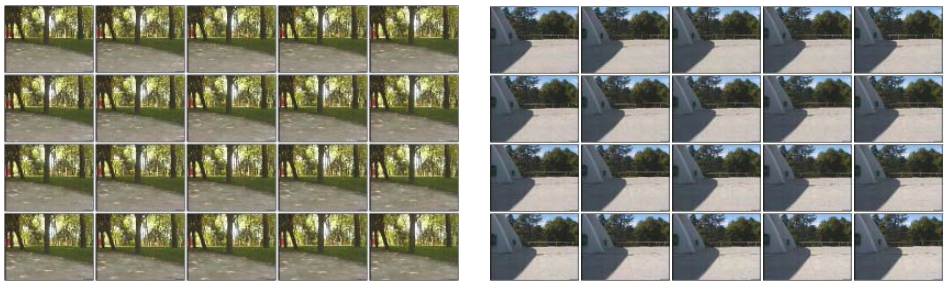
Figure 6.19: Experiment 2 - The gyroscope information recorded during the trajectory

Therefore, the resulted topological map is defined by 4 states. The State 1 is defined mainly by its slope (around -16°) and the State 4 as an horizontal terrain. There are two other states that define when the robot enters and leaves the ramp. They are identified by the Yaw and the Pitch, which translates the characteristics of the terrain, since State 3 includes sand, which provides the noise as previously described.

6.3 Conclusions

This chapter illustrated how it is possible to implement a topological navigation on mobile robots in different environments, indoors or outdoors scenarios, using different types of

State 1 State 2



State 3 State 4

Figure 6.20: Images illustrating the 4 states of the resulted topological map using features extracted from inertial sensors in the second experience carried out at Palácio de Cristal

	State 1				State 2			
	μ	R			μ	R		
Roll	0	0.9	0.3	1.1	0	1.8	-2.5	-2.0
Pitch	-16	0.3	3.0	1.8	-6	-2.5	3.9	-14
Yaw	300	1.1	1.8	0	286	-2.0	-14	65
	State 3				State 4			
	μ	R			μ	R		
Roll	2	2.2	1.1	-0.7	1	1.3	-0.7	23
Pitch	-3	1.1	7.4	9.8	0	-0.7	6.5	-52
Yaw	247	-0.7	9.7	28	216	23	-52	125

Table 6.7: The state values of the resulted topological map using gyroscope features

features and including the components of mapping, localization and navigation.

The experimental results obtained in an indoors scenario illustrated that is possible to build a topological map using only range sensors, identifying three states, room, corridor and entrance, in the human perception. However, as the localization experiments demonstrated, the resulted topological representation depends on the orientation assumed by the robot when following a path. This issue is caused by the type of sensors that limits the angle of view, even with the most appropriated features. The localization is able to determine the robot's position in the topological map even when the robot follows a different path from the one(s) used for mapping. Nevertheless, if the robot follows a similar trajectory, but in a reverse order, the localization presents a large level of uncertainty, as illustrated in the experiment carried out indoor environment, illustrated in Figure 6.8.

The same issue occurs in outdoor environment, at the IST Lisbon campus (Alameda). When the mobile robot moves from State 3 to State 4, illustrated in Figure 6.15, it has to change temporarily its orientation given moving obstacles. Thus, the robot observes

features corresponding to another state, producing some noise in the localization results. This handicap may not be interpreted as an algorithm limitation, since it is caused by the properties and positioning of the hardware that supports the robot's perception. Given the orientation dependence caused by the type of sensors, the topological representation may result in two or more different maps according to the followed direction. For instance, following the same road in different directions, may result in different observations, given the sensors limitation. Even improving the types of features (as the histograms colors or vertical edges as described in the previous chapter) this fact of orientation dependence persists. It is important to remember that the field of view of the camera or range sensors (except the ultrasound sensors) is reduced to less than 120º.

The precision of the topological representation depends not only on the type of features, but also on the amount of information retrieved by features. A large number of features may increase the number of states if the environment is characterized by a diversity of information. However, each feature could be represent by different parameters, that define the precision of the selected feature. The amount of parameters, in the thesis identified as k (for instance, k_{hist} defines the number of parameters to represent the histograms), also determines the computation cost of the mapping algorithm. For a large value of k, the application in real time is bouncing, even if the mapping algorithm is running at a low rate when compared to the localization or navigation, as represented in Figure 2.3.

The transition probabilities between states, a_{ij}, completely dependent of the localization efficiency, are refined when the algorithm runs several times in the same scenario. The same occurs with the orientation between states, θ_{ij}. However, the navigation is able to overcome the problem, when the orientation is not correct moving the robot to an undesired state not covered by the current sequence of states to reach the goal. In this situation, the navigation calculates a new sequence of states to overcome the situation, adjusting θ_{ij} to a new value.

Therefore, there are still open issues as the algorithm optimizations (select the appropriate values for several threshold values), improve the initializations, or even increase the types of features. These main open issues are discussed in the next chapter.

Chapter 7

Conclusions and Future Directions

7.1 Summary

The thesis addresses the problem of mobile robot navigation in unstructured outdoor scenarios and proposes methodologies based on a topological approach that accomplishes a scenario representation at a high level of abstraction. Based on a topological representation, the localization and the navigation problems are solved at the same level of abstraction.

The thesis starts from a motivation based on some cutting edges of the nature. There are millions of species with fantastic navigation capabilities, that retrieve from the surrounding environment the essential for motion, like ants, honeybees, migratory birds, etc. This motivation aims the three main problems of robot navigation: Localization, Navigation and Map Building. The thesis explains the importance of addressing these three problems simultaneously, all of them at a topological level.

Since these three main problems are based on an environment representation, the other possibilities besides the topological representation are discussed, namely geometric and hybrid maps.

The topological representation relies on a set of nodes that represent the states, characterized by a set of relevant features modeled by mathematical functions (e.g., n-dimensional Gaussian pdfs, where the dimension corresponds to the number of different features). This type of representation corresponds to a high level of abstraction, covering large physical areas, which is implemented as a hierarchical level of representation, according to the required resolution. The associated to the high level of abstraction, not yet covered by researchers on robotics, justify the topological representation as the selected approach for the first steps of the target application that motivated the thesis: search and rescue like operations. These type of operations usually take place in completely unstructured environments, with a high level of complexity yielding a difficult problem of robot navigation. The thesis developed important steps towards mobile robot navigation in outdoors, highly unstructured environments and implemented indoor and outdoor structured scenarios, motivated by a long-term goal of autonomous operation in situations of real catastrophes.

To build the topological map, it is necessary to estimate the mean vectors and the covariance matrices that define the Gaussian pdfs that characterize each state of the map. The adopted algorithm is a modified and revised version of EM, Dynamic Expectation and Maximization algorithm, which adjusts the number of states and the parameters that define the mathematical functions (in particular the sum of n-dimensional Gaussian pdfs). It is possible to establish the precision of the mapping algorithm by adjusting some parameters of the Dynamic EM. However, the parameters represent features, (for instance, k_{hist} defines the number of parameters that represent the histograms) and also determine the computation cost of the mapping algorithm. For instance, in one of the outdoor experiences, at IST campus of Alameda, the PCA features becomes difficult to implement in real time, since PCA uses image processing. Therefore, features with large dimension, increases not only the amount of information but also the computational cost.

The approach to accomplish the topological representation uses different types of features, represented by vectors. The quality of the topological representation depends on the type of features used in each scenario. The feature selection is based on a covariance matrix retrieved during a testing phase to evaluate the best features for a particular scenario. In the thesis, the used features are extracted from the rawdata acquired by range, inertial and vision sensors. The type of features and the corresponding procedure to extract that information is defined at the initialization of the mapping algorithm.

The connection between states is also necessary in the topological representation. The linkage between two different states s_i and s_j, is represented by the transition probability between these states, a_{ij} and by the orientation to switch from one state to the other, θ_{ij}. While a_{ij} has a probability information with a major influence in the localization, the angle θ_{ij} provides physical information to drive the robot between states, i.e., is useful for the navigation procedure.

The transition probabilities between states, a_{ij}, and the corresponding orientations, θ_{ij}, are estimated based on the localization results and, consequently, are dependent on the localization efficiency. The values of a_{ij} and θ_{ij} are refined when the algorithm runs several times in the same scenario. However, the navigation is able to overcome the problem of wrong values of θ_{ij} (given the localization uncertainty, the robot can move to an undesired state not covered by the current sequence of states to reach the goal). When this occurs, the navigation calculates a new sequence of states, adjusting θ_{ij} to a new value.

The localization and the navigation problems are addressed based on the topological map, a set of states, S, represented by Gaussian pdfs and the corresponding connections, a_{ij} and θ_{ij}. Given the type of representation, the localization and navigation problems are also addressed with a topological approach.

The mathematical support for localization and for navigation has to deal with the uncertainty included in the world, the perception of the world (observations and/or consequently on features) and also on the motion. This justifies a probabilistic approach, a fundamental issue in the thesis.

The localization problem is addressed by a changed version of the Forward-Backward (FB) algorithm and Markov Models to estimate the current robot's state in the topological map. The localization algorithm minimizes the uncertainty given the observed features. The changes introduced in the FB algorithm, consists on sampling the time in intervals and recording the observations acquired during each interval in a buffer of observations. These intervals or buffer of observations retains the most relevant information for the localization algorithm at each time instant. The FB algorithm estimates the robot's location in each time instant using the past and the future information acquired during a time interval, where the current time instant is integrated.

The navigation algorithm retrieves the best path to reach a goal in the topological map with the best certainty. The linkage between the high level of abstraction, the topological navigation, and a motion control to follow the resulted sequence of states, is produced at this stage. The motion control is based on a sum of behaviors, an attractive behavior to the state goal and a repulsive behavior to avoid obstacles or not desired directions.

The experimental results are divided in two categories, indoors and outdoors. The experiments accomplished indoors are based in simple features extracted from range sensors to test mainly the map building algorithm. The resulted states have some meaning for the human perception: the room, the corridor and the entrance of the experience in a laboratory. These results are illustrated using the images acquired during the experiments.

The approach explanation, the algorithms description and the experimental results obtained by the implementation illustrate a different type of SLAM, the topological SLAM, where the three main problems are addressed simultaneously, at different rates and all based on a topological framework. Furthermore, it is necessary to underline the novelties and the limitations that are present. The main contributions and novelties of the thesis are,

- The topological map, a type of representation with a high level of abstraction,
- A different type of SLAM, a topological SLAM,
- The Dynamic Expectation and Maximization algorithm for map building,
- The probabilistic approach for localization and navigation with a revisited Forward-Backward algorithm and
- The feature extraction and selection methodologies.

7.2 Evaluation of the Approach

As described and illustrated in the thesis, it is possible to develop a topological approach for the navigation at a high level of abstraction and implemented in simple outdoor environments. The experiments were accomplished outdoors, even though in not so unstructured environments as in disaster scenarios. For more complex environments, as the scenarios

caused by disasters, different features may be necessary, retaining more heterogeneous information. Vertical edges provided by the buildings or the PCA components illustrating the sky, trees, ground, different characteristics may not be so common in disaster areas and may not be present. The features as colors and histograms are more acceptable for these types of scenarios, but also temperature, radio-frequencies, images acquired by different types of cameras (for example, infra-red cameras or UV cameras), are essential information for these situations in unstructured scenarios.

As seen, the feature extraction is an important topic of the topological approach, since it reduces drastically the amount of storage space required by the rawdata. The features retain the essential information, parameterized by some Gaussian pdfs. It is possible to navigate in large environments in real time, but without a sharp representation of the scenario.

It is important the advantage of this topological approach, since it represents the possibility of, hierarchically, represent the environment. It means that the scenario may be represented by a topological map with a low resolution and therefore, with each state incorporating another topological map or even a geometric map, becoming an hybrid map.

With a simple topological map or with a hierarchical representation, it is important to underline the necessity of the linkage between the high level of navigation based on the topological representation and a motion control or hardware control. This linkage is implemented in the navigation algorithm by a behavior approach that converts the sequence of states in velocities. To accomplish this goal, the orientation between states, θ_{ij}, that is recorded in the topological map, is necessary. Not only the angles θ_{ij} but also the other variables (as a_{ij}) require an initialization procedure.

The experiments illustrate a limitation of the selected approach of the thesis. When a similar path is followed by the robot but in a reverse order, the topological localization usually presents different results, in a situation when the mapping algorithm was intentionally disabled to test the localization in both directions. When the mapping was disabled and the robot followed the same path but in a different direction, the localization presented some uncertainty in the results, which underlines the necessity in this situation of running the mapping algorithm to update the current map. This illustrates the dependence of the robot on the type of features or mainly the different points of view of the scenario. It is expected that a topological map using the features extracted from the rawdata acquired by a sensor with a view angle equals to 45^{o} has more states than if using a sensor with a large angle of view, at least equal to 180^{o}. The experiments accomplished outdoors test the feature selection efficiency, disregarding the most correlated features. In the same experiments it was possible to estimate the connections between states, the transition probabilities between states, a_{ij}, and the corresponding orientations, θ_{ij}. However, these values should be refined with several rounds in the same scenario. Other outdoor experiments illustrate how it is possible to build a topological representation in an outdoor scenario, but with a sensor that retrieves a poor amount of information, as a gyroscope.

The initialization is a very important topic in the topological approach. When the mobile robot is switched on and it starts from zero, i.e., no a priori map is available and the best features are still unknown, it is not expected to accomplish a mission at once, as a Search and Rescue (SaR) like operation. It requires a learning phase, to perceive the best types of features to represent the scenario, while building the map. If the mapping algorithm starts from zero, there is no initial map (number of states, best or less correlated features unknown) and the procedure requires more computational time. Otherwise, starting from an initial topological map, even a reduced one, the performance increases when the algorithm has only to update the current map and to adjust the previous parameters.

Starting from zero, with the best features still unknown, is similar to a "cold initialization" (commonly used in GPS frameworks: when the receiver is switched on and the position of satellites is unknown), without a map for the localization and navigation procedures. Consequently, during an initial phase, defined as the learning phase, the mobile robot moves in random directions. After a period of time, a map is available based on the best features selected. This period of time can be long and considered as a drawback of the topological approach. Therefore, different types of initializations may be addressed as a possible field of future research, as described in the following subsection.

If an initial map is available, even if incomplete, the localization estimates the robot's position in that map and, consequently, the navigation has to compute the best sequence of states to reach the main goal. However, during the robot motion, the mapping algorithm continues updating the representation (number of states, the respective Gaussian pdfs and the connections between states). The efficiency of an initial map depends on the type of used features. As an example, some states obtained in the experience made in the Campus of IST at Alameda strongly depend on the features characterized by the position of the sun. The State 5 in Figure 6.14 is defined by the yellow color, based on the position of the Sun. If the experience had been accomplished at a different time of the day, for instance, during the morning, the edges would be the same but the histograms would be different. This issue imposes a feature optimization. But even with some features strongly robust to these conditions, the ambiguities still remain, i.e., some features are not completely invariant to the light conditions. It would be difficult to use the same topological representation to navigate in a scenario at day or at night, or even during the sunrise/sunset. This situation bears to a possibility where the time information is also gathered as a particular feature to choose the appropriate representation. For example, a specific topological map during the day and another topological map during the night.

The algorithms that implement the topological approach for mobile robot navigation are tested using a real robot of the Rescue Project (described with more detail in Appendix A), the terrestrial vehicle ATRV-Jr (see Appendix B). However, the navigation in outdoor environments is not the only long-term goal of the referred project. The capability of cooperation is also a target, where a map is shared by all the robots integrated in the mission.

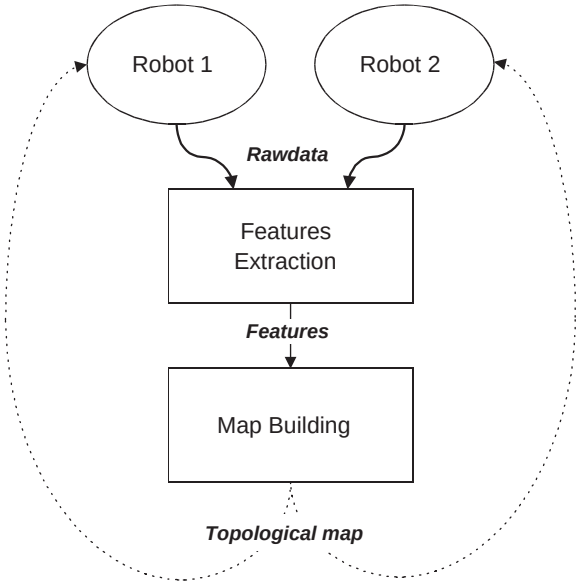


Figure 7.1: The topological approach with two robots

The topological approach described in the thesis aims the capability to provide a map useful for the joint operation of several robots, as illustrated in Figure 7.1. Each robot can be equipped with different types of sensors, extracting from the environment only the information that can be perceived by that robot. From the point of view of the team, large amount of rawdata is acquired in different places. All this information is converted in features (by each computer installed in the robots or in a central unit) and, consequently, the mapping algorithm provides a topological map, shared as a broadcast. The number of robots may change along time. Each robot, using the same map (or the part of the map that it can understand given the installed sensors) and integrating the acquired observations, accomplishes the localization and the navigation procedure. A map built based on the features acquired by robots in different places is similar to a map computed by a single robot moving through these states and retrieving similar features. However, the connection between states may differ for different types of robots. For instance, two different robots, one terrestrial and other aerial (as in the Rescue Project) leads to different transition probabilities a_{ij}. A physical obstacle for the terrestrial robot, which results in a low a_{ij}, could not apply for the aerial robot, but the set of states have common information. The process to estimate the connections between states can be the same for all robots.

Hence, the topological approach can be implemented in a team of robots, if each robot is equipped with a traditional computer (without particular specifications). In spite of a computer, robots may be equipped with a motion controller, sensors and transmitter/receiver that communicate with a central unit that supports the computational process.

Besides the advantages of the topological approach it presets limitations, opening future directions of research as described in the next subsection.

7.3 Perspective of Further Research

The topological approach on mobile robot navigation addresses several areas of research. This challenging application is supported by mathematical models to solve the three main issues of environment representation, localization and navigation. Most of the topics include open issues, in particular

- The precision of a topological map, including hybrid maps,
- The linkage between states (not necessarily by the orientations θ_{ij}),
- The initialization and the adjustment of some threshold values,
- Different types of features including new types of sensors (providing features independent to the light conditions and to the robot's orientation),
- New feature selection methodologies and,
- Develop the topological approach using a team of robots on a cooperative navigation.

A topological map, when defined as a set of states, is represented by mathematical functions. In the thesis, the mathematical functions are multi-dimensional Gaussian pdfs. The performance of the representation and the efficiency of the mapping algorithm should be tested if the states are represented by different types of mathematical functions, not Gaussians. The algorithms used in the localization and the navigation problems can also be different. The problem of localization consists on finding the current robot's state given the observed features. The selected algorithm may follow a probabilistic approach, but different from the Forward-Backward algorithm. This also applies to the navigation, where a different algorithm may be used to select the best sequence of states.

The connection between the states of the topological map may not include the orientations, but be represented by commands. For instance, to switch between states, the connection information may assume values in a set of motion commands, as "move forward", "move left" or "maintain the current direction". The motion is achieved by a set of predefined instructions to control the robot's actuators. As in the adopted approach explained in the thesis, the connections between states have to be learned as the state transitions probabilities.

A future direction of work has to evaluate the localization and navigation performance with topological maps at different resolutions/precisions, mainly when configured with other maps available in each state, i.e., an hybrid representation. The resolution/precision defines mainly the amount of states, which is important in the initialization and, consequently, in the implementation performance.

Certainly, the most relevant and open issue is related with features. The resolution and the precision of a topological map is intrinsically related with the type of features. To select the appropriate features for representing a specific environment, it is necessary to

know the available features to test. The approach adopted in the thesis does not provide a procedure to discover types of features directly from the acquired rawdata.

In the thesis, the types of features are selected according to the target scenario. The features database (to know which features to explore) remains as an open issue, since there are certainly more features then free-area, edges, histograms or PCA and ICA. One of the most prominent area is the feature extraction, which covers promising research topics, mainly on vision. It is possible to extract large amount of relevant information from image rawdata. There are other types of features not covered in the thesis, as textures. Most of the research developed on texture extraction relies on Gabor Filters and/or Nonlinear Operator [47, 67]. In spite of the types of filters or methodologies to extract the features, there are also open issues on the way to represent the features. As described through the thesis, the features are represented as vectors, where the dimension corresponds to the precision of each feature. It is important to develop a tuning criteria for the dimension of each feature (e.g., optimize the appropriate number of edges, k_{edges}, or the number of Gaussians to parameterize the histograms, $k_{histograms}$, or the number of boundary-boxes, k_{2Dhist}). The features' precision reflects that non correlated or weakly correlated features are not necessarily essential, if they carry poor information.

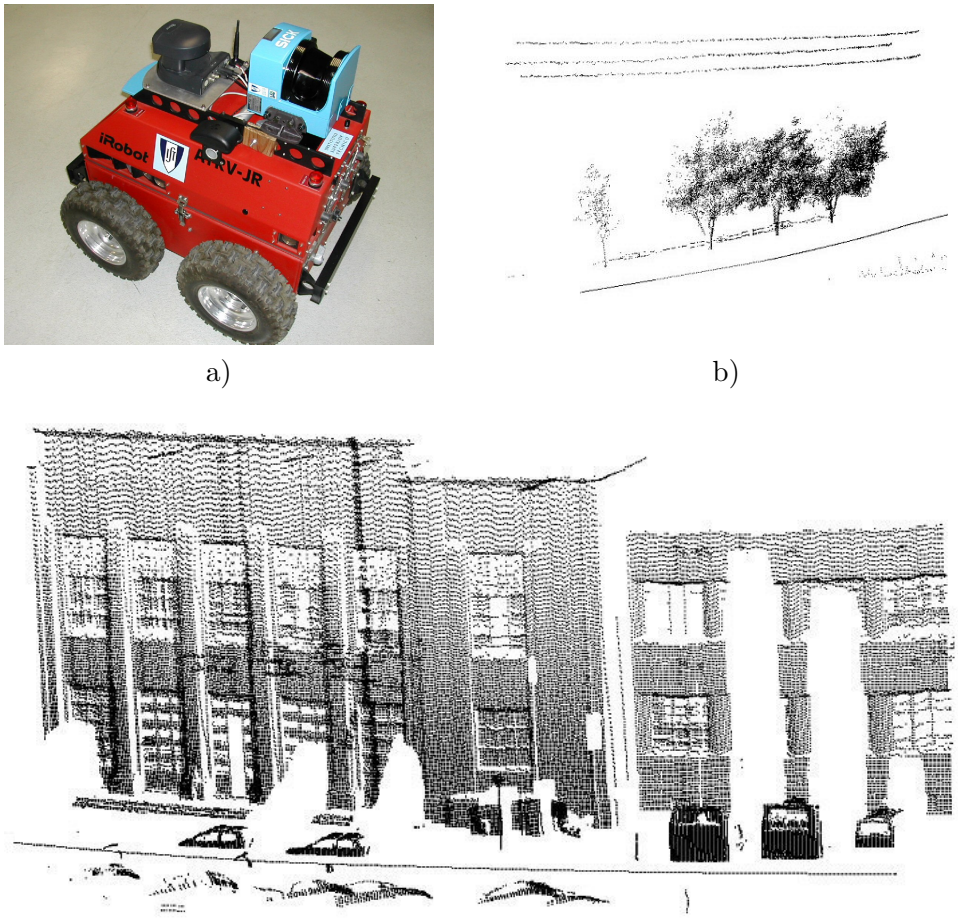
Extracting more information from the robot's perception requires a possible hardware improvement, installing new sensors on the mobile robot, as suggested by [24, 58, 69, 122]. Even the best types of features extracted from image data acquired by the camera installed on the ATRV-Jr are deeply dependent on the robot's orientation. Because of this situation the mapping procedure builds different states based on the observations acquired at the same physical place, but with different orientations. This situation should not be taken as a drawback as explained before, since the localization and the navigation algorithms also return possible solutions based on the current topological map. However, the resulted topological map may not suit the requirements, mainly when dealing with human perception/operation. For instance, a room is also a room independent on the observed direction.

One possible choice to avoid this situation is based on a panoramic camera, as the one suggested in 0-360.com [7] illustrated in Figure 7.2-a), with an image example in Figure 7.2-b). For several image acquisitions at different orientations, it is possible to apply a transformation (shifting the images along the horizontal axes), resulting in the same image. This assumption requires an horizontal/quite-horizontal robot's placement.

The features extracted from the range sensors installed on the robot provide poor information, mainly in outdoor environments, even with the Sick Laser. This situation occurs since the sensor is installed horizontally on the robot and the measurements acquired at successive iterations are highly correlated, since the robot motion reflects its non-holonomy. If installing a similar range sensor transversally to the robot's movement, it would deeply increase the amount of acquired information, as suggested by [51] on the Pioneer 2 AT platform equipped with two laser range scanners.

a) b)

Figure 7.2: Perspective of adding an omni-directional camera on the top of the ATRV-Jr, a) the commercial model presented in 0-360.com and b) the resulted panoramic view



a) b)

c)

Figure 7.3: Perspective of adding new sensors to enlarge the features information: a) an extra laser mounted on the top of the robot, perpendicular to the movement, acquiring data containing b) trees and cables or c) a building

The Figure 7.3-a) illustrates a possible implementation of a second Sick laser targeting to vertical scanning. The Figure 7.3-b) and -c) illustrate two different scenarios, where trees and cables (in the first image) and a building and cars (in the second image) are perceptible. The improvement on the type of sensors opens new directions of research about feature extraction.

The topological approach presented in the thesis performs a high level of abstraction, for any type of mobile robot equipped with any set of sensors. Even tested with a particular wheeled robot, ATRV-Jr, the resulted topological map could be shared by other robots since equipped with similar sensors, or different sensors capable to return the same features, to perceive and understand the essential information of the topological map.

In the frame of the Rescue Project, to achieve cooperative navigation in outdoor environments, it is necessary to test the topological approach with different types of features in real scenarios. The features have to be extracted by appropriate sensors (for example, panoramic cameras, infra-red sensors), that better perceive the scenarios in real situations of disasters. The rawdata does not necessary come from a single robot. Stressing the importance of cooperative navigation, it is possible to extract these relevant features from several and heterogeneous robots (for instance, terrestrial, aerial, legged, wheeled, etc). The information is collected and sent to a central unit that retrieves a topological representation and localization of each robot, returning the motion commands for each robot according to its target goal. The operation can be simultaneously supervised by a human operator, providing tele-operation if necessary. The retrieved map can be shared by these robots, where it is used only the appropriate information from the map for navigation, concerning to the target applications.

Appendix A

Related Research: The Rescue Project

The Rescue Project – Cooperative Navigation for Rescue Robots (POCTI/1999/SRI/33293), was a multi-disciplinary joint venture to face the challenging applications of mobile robot navigation, addressing different issues such as scenario mapping, multi-robot (cooperative) navigation, (multi-robot) task planning and coordination. Three research groups (Intelligent Systems, Computer Vision and Mobile Robotics) of the Instituto de Sistemas e Robótica at Instituto Superior Técnico (ISR/IST), Lisboa, joined their research efforts on outdoors search and rescue robots under this project.

The main goal of the project was to provide integrated solutions for the design of teams of cooperative robots operating in outdoor environments. The scientific challenge in the short and mid-terms periods were on perception and representation issues, as well as cooperative navigation, and, in the mid to long-term, on task modeling, planning and coordination of multi-robot systems. The reference scenario for the project referred to a long-term goal of developing robotic teams to help humans in search and rescue missions. At a first stage, the project was based on two robots, a land and an aerial (blimp) robot. Simplifying assumptions include daylight operation and good meteorological conditions (weak winds and no rain).

The following is a list of problems that were addressed and, at least, partially solved during the Rescue Project lifetime:

- Definition of a functional architecture suitable for the integration of the subsystems composing the robotic team, oriented towards Search and Rescue (SAR) applications.
- Choice of the appropriate sensors and sensor fusion methodologies.
- Issues concerning topological navigation, world-model-based and sensor-based navigation within unstructured environments, as well as their extension to cooperative navigation.

- Task coordination, including the appropriate integration of the functional subsystems and resources management during the execution of a given robotic task.

The scientific advances made on the enabling disciplines (Computer Vision, Robot Navigation, Hybrid/Discrete Event Systems and Artificial Intelligence) during the project lifetime should be extendable to other outdoors applications, such as environmental monitoring and surveillance, satellite formations or planetary exploration and also to a team with a larger number of robots.

A.1 The Reference Scenario for the Rescue Project

A reference scenario for the Rescue Project was set up with two main goals: it should refer to a reasonably realistic situation, and it should be rich enough to accommodate all the research topics of interest for the involved groups. The robots used in the projects are one aerial blimp/zeppelin and one land outdoors robot. The aerial robot will perform several tasks, such as making a vision-based topological map of the destroyed site. The map will include information on the relevance of each of the mapped locations concerning the degree of destruction, presence of victims, etc, as well as on the difficulty of traversing regions between them, due to the presence of debris or obstructed paths. The map will be stored as a graph and will be used to choose the best path for the land robot to reach a goal location (e.g., one with a larger number of victims). It can also be used to help the aerial robot navigation. Issues of representation arise here, as the views of the land and aerial robots are different, even when they refer to the region associated to the same node of the topological graph. The land robot will use several sensors (GPS, inertial, vision, laser scanner, ultrasound sensors) to navigate towards the goal, handling the details associated to the path (e.g., debris, trees, people on the way, etc). This will extend the scope of the research to metric navigation. Nevertheless, the topological map obtained by the aerial robot can also be used as the initial iteration of a topological mapping algorithm based on cooperative information from the aerial and land robots. While the land robot moves towards the goal location, the aerial robot should follow it using a formation control algorithm, so as to keep a reliable communication link and to serve as a relay for information that the land robot may need to send to distant stations. Considering a long-term goal as a completely autonomous robot or teams of heterogeneous robots in a real scenario of catastrophes, the project requires a software architecture capable of handling real-time distributed control and supervision systems, as well as a functional architecture to integrate all the required subsystems (and their corresponding functions) consistently, i.e., so as the whole system is designed to get the maximum performance. The software architecture, currently in its implementation stage (designed for two robots), is based on a multi-thread and distributed blackboard approach. The different threads handle sensor data acquisition and processing, communications, behavior execution and behavior switching. Raw and processed data go to the distributed blackboard, a distributed shared memory organized in classes of vari-

ables corresponding to the different sensor classes. Some variable values are kept locally at the robot that acquired and processed the data. Others are broadcasted (or sent individually) to the teammates. Some transducers have an associated set of virtual sensors, each of them implemented by a separate agent. A typical example is the vision transducer, with associated virtual sensors aiming at localizing the object, determining the object speed and classifying the objects. The functional architecture includes the following subsystems: task planning (to handle task/behavior allocation, intra-team communications and formation topology, when a formation is required), task coordination (to handle the decisions on which behaviors to select at every time step, so as to optimize the performance in terms of time taken, reliability, cost, utility or other factors - requiring performance feedback from each subsystem) and behaviors (referring to one single agent and involving relations among more than one teammate). Behaviors are built as finite state automata, where arcs are associated to conditions of detected events and most states correspond to dynamic control systems implementing navigation functions, parameterized according to the task at hand (e.g., move until a given posture is reached, track an object or a teammate). Note that behaviors are implemented using alternative approaches (e.g., based on Petri nets or production rules), with relative advantages concerning the representational power but a potentially reduced debugging power. However, the software architecture is open enough to host these alternative models.

The navigation functions are the key practical developments expected during this project. They will be based on a multi-sensor system (GPS, compass, odometry, laser scanner, ultrasound sensors, vision and inertial navigation system composed of rate-gyros and accelerometers) for the land robot, an ATRV-Jr. The navigation of the blimp will be mostly vision based. Both the blimp and the ATRV-Jr navigation systems will mix metric and topological navigation. The current work concerning metric navigation under the project builds on existing closed loop solutions which use the GPS as a reference and odometry plus accelerometers as feedback sensors for position estimates, the compass as a reference and the rate-gyros as feedback sensors for orientation estimates (roll, pitch and yaw). The novelty here is on handling particular problems of outdoor environments, such as very poor odometry and frequent loss of GPS information due to trees or buildings. The amount of useful information must be increased by more than one robot, where a network of communicating sensors assembled on the team robots can take advantage of their space diversity to provide information to each of the team robots which they might miss if operating alone (e.g., one of the robots does not have GPS available but one of its teammates does and can also estimate their relative postures). All this work will necessarily require sensor models, which cannot be obtained before outdoor runs, during which data from all navigation sensors is registered, are performed. The recorded data can subsequently be subject to statistical analysis in order to build the sensor models.

A.2 e-Links and Multimedia

A web page has been created for the project, which URL is

http://rescue.isr.ist.utl.pt/.

An animation illustrating the scenario above was developed to better explain the reference scenario to project, is available at MPEG format at:

http://rescue.isr.ist.utl.pt/videos/rescue_web.mpeg.

The members of the project are mentioned in

http://rescue.isr.ist.utl.pt/members.php.

and respective publications.

http://rescue.isr.ist.utl.pt/publications.php.

There is also a web page that addresses the development in topological navigation in outdoor environments, concerning the topic of the thesis and the relevant research carried on the land vehicle, the ATRV-Jr,

http://lrm.isr.ist.utl.pt/rescue/.

Appendix B

iRobot ATRV-Jr

B.1 Technical Specifications

Figure B.1: Mobile Robot ATRV-Jr with a Sick Laser LMS and a Pan and Tilt camera Sony EVI-D31

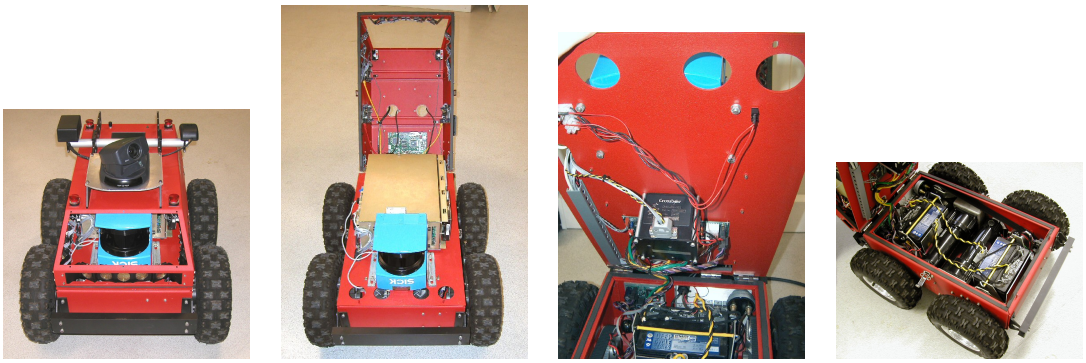
The ATRV-Jr is the land robot used in the Rescue Project, equipped with 4-wheel differential drive, with 2 high torque 24V DC servo motors, capable of achieving linear speeds up to 1 m/s and turning speeds up to 120 deg/s. With a 25 kg of payload, it weights 50 Kg plus 5 kg with external sensors and is capable of overcoming 45 degrees slopes. Its 2 lead acid, 672 Watt/hr batteries provide an autonomy of 3 to 5 hours, depending on the motion conditions. The on-board computer is a Pentium III running at 800 MHz. Communications with other robots and external computers are ensured by either a 100/10 Mbps Ethernet board or IEEE 802.11b Wireless radio Ethernet. The robot

includes an odometry system and Linux-based management software and libraries.

The ATRV-Jr is equipped with several navigation sensors:

- Computerized navigation compass,
- Global Position System (GPS) receiver (12-channel), Garmin GPS-35,
- Inertial Navigation System (DMU), including rate-gyros,
- 17 ultrasound sensors (5 forward facing, 10 side facing and 2 rear facing),
- 2D Laser Range scanner Sick LMS and
- Pan and Tilt camera Sony EVI-D31 12x optical zoom.

The compass and GPS are installed on the top of the vehicle to reduce electric interference. The video camera is also installed on the top of the vehicle to perform the best field of view. The range sensors are horizontally installed for information acquisition around the robot. The DMU is hidden, inside the vehicle, near the motor and batteries. All the sensors are connected to the robot on-board computer by RS232. The Figure B.2 illustrates the mobile robot and its equipment, through different side views.

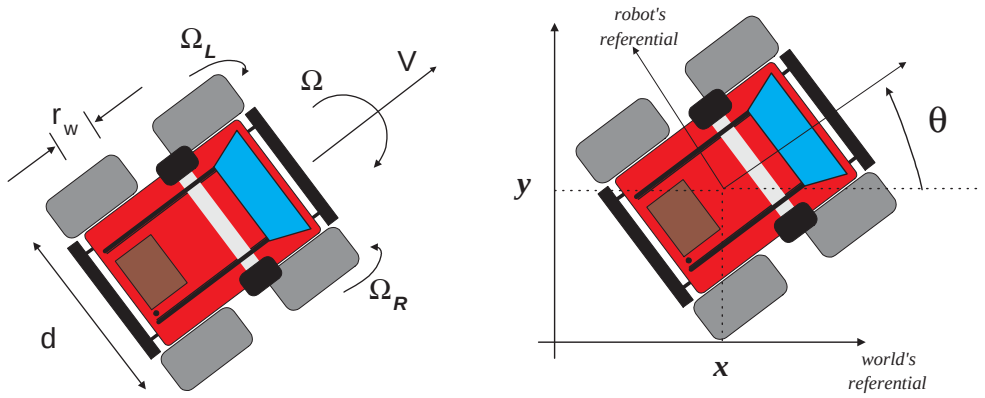


a) b) c) d)

Figure B.2: Different views of ATRV-Jr: a) from left to right, the compass, camera and GPS, the blue object is the laser sensor and a ring of ultrasound sensors, b) the computer and the laser sensor, c) the DMU and d) the motors and batteries

B.2 Robot and Sensors Model

The mobile robot is a non-holonomic vehicle, equivalent to a 4-wheel differential drive. The nominal radius of each wheel is $r_w = 10cm$ and the length of the axis between wheels is $d = 40cm$, as illustrated in Figure B.3. The radius of wheels varies with the payload installed on the robot, air pressure, temperature and soil, but r_w, the nominal value is assumed as constant.



a) b)

Figure B.3: ATRV-Jr model, a) the velocities and b) the position and orientation

The angular velocities of right and left wheels, given by Ω_R and Ω_L, are the only variables that can be controlled. The linear and angular velocities of the robot, $V(t)$ and $\Omega(t)$ respectively, relative to the central point of the robot body are given by

$$V(t) = \frac{r_w}{2}(\Omega_L(t) + \Omega_R(t)) \tag{B.1}$$

$$\Omega(t) = \frac{r_w}{d}(\Omega_L(t) - \Omega_R(t)). \tag{B.2}$$

The pose of the vehicle, $\mathbf{x} = [x \;\; y \;\; \theta]$, relative to the world referential (see Figure B.3-b) evolves in time according to

$$\dot{\mathbf{x}}(\mathbf{t}) = f(\mathbf{x}(t), V(t), \Omega(t)). \tag{B.3}$$

Based on (B.1) and (B.2), the robot kinematics (B.3) can be expressed as

$$\begin{cases} \dot{x}(t) &= V(t) \cdot \cos\theta(t) \\ \dot{y}(t) &= V(t) \cdot \sin\theta(t) \\ \dot{\theta}(t) &= \Omega(t) \end{cases} . \tag{B.4}$$

For the implementation of dead-reckoning, a discrete version of the kinematics (B.4) is required. Considering a uniform sampling interval T and assuming that in the interval $[KT, (k+1)T[$ the robot follows a line segment with translation velocity $V(k)$ followed by a rotation of $\Omega(k)$, an approximate discrete kinematics is given by

$$\begin{cases} x(k+1) &= x(k) + T \cdot V(k) \cdot \cos\theta(k) \\ y(k+1) &= y(k) + T \cdot V(k) \cdot \sin\theta(k) \\ \theta(k+1) &= \theta(k) + T \cdot \Omega(k) \end{cases} . \tag{B.5}$$

This approximation is equivalent to

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + \begin{bmatrix} V(k) \cdot \cos\theta(kT) \cdot T \\ V(k) \cdot \sin\theta(kT) \cdot T \\ \Omega(k) \cdot T \end{bmatrix}, \tag{B.6}$$

where the values of $V(k)$ and $\Omega(k)$ are given by the odometry and it was assumed that

$$x(k+1) = x(t)|_{t=(k+1)T}. \tag{B.7}$$

Given the errors introduced by the wheels slippage, irregularity of the floor, sampling, encoders precision, the radius of wheels, there is an uncertainty associated to the odometry, where the velocities estimation are $\hat{V}(k)$ and $\hat{\Omega}(k)$, respectively. The uncertainty is modeled as η_Q, herein considered as a white Gaussian noise, with a covariance matrix $E[\eta_Q \eta_Q^T] = Q$, i.e., the considered discrete kinematics model is

$$\mathbf{x(k+1)} = f(\mathbf{x(k)}, \hat{V}(k), \hat{\Omega}(k)) + \eta_Q. \tag{B.8}$$

The value of Q was estimated from several experiments, with the component relative to the orientation presenting a high variance.

The observations that will feed a pose estimation methodology are given by two sensors: the GPS sensor (based on UTM/UPS Universal Transverse Mercator/Universal Polar Stereographic 24 satellites [75], as illustrated in Figure B.4) and the compass.

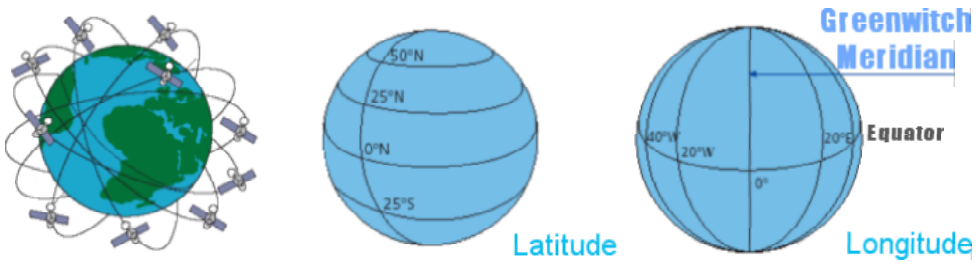


Figure B.4: GPS basics, Latitude, Longitude and Altitude

The distance followed by the robot between two GPS acquisitions is estimated by the differences between latitude and longitude values acquired. Since the Earth planet is nearly spherical, the distance is given by multiplying the difference in latitude and longitude by the radius of the planet, $r_E = 6378$ kilometers. Let $lat(k)$ and $long(k)$ be the latitude and longitude of the robot at time instant k provided by GPS. The distance between k and $k+1$ along the latitude and longitude is given by

$$r_E \cdot (lat(k+1) - lat(k))$$

and

$$r_E \cdot (long(k+1) - long(k)),$$

respectively. However, the vehicle position corresponds to the x and y values in a local referential. It is important to link this referential with longitude and latitude. Since the latitude is measured along the meridian line, as depicted in Figure B.5, and the orientation, which is gathered by the compass (having better precision than GPS), the x-axis is along latitude (to the north) and y-axis is along longitude, but with a reverse signal. However, the true north, retrieved by the compass, and the magnetic north, retrieved by the GPS, are not the same, which requires the correction given by

$$long(k) \leftarrow long(k) * \cos(lat(k)).$$

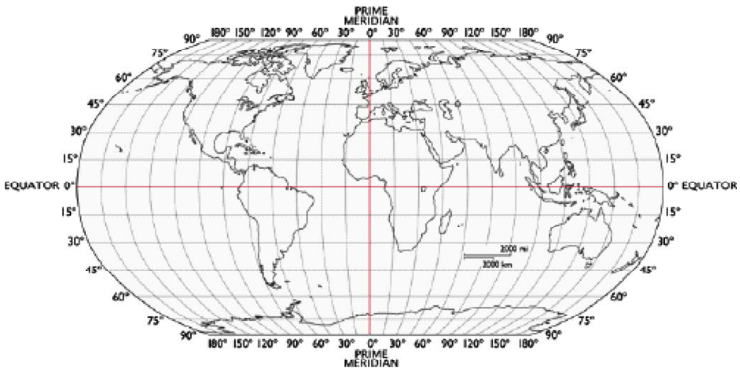


Figure B.5: World referential

The observation is equivalent to the robot's pose, i.e.,

$$\mathbf{z}(\mathbf{k}) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix}. \tag{B.9}$$

While the orientation is directly obtained by the compass measurements, the position along x-axis and y-axis requires a transformation of the GPS values, or equivalently

$$\begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} \leftarrow \begin{bmatrix} x(k-1) + r_E(lat(k) - lat(k-1)) \\ y(k-1) + r_E(long(k) - long(k-1)) \\ \theta_{compass}(k) \end{bmatrix} + \eta_R. \tag{B.10}$$

There is also an uncertainty associated to the sensors and, consequently, the (B.9) is rewritten as follows

$$\mathbf{z}(\mathbf{k}) = \mathbf{g}(\mathbf{x}(\mathbf{k})) + \eta_{\mathbf{R}}, \tag{B.11}$$

where $g(\mathbf{x}(\mathbf{k}))$ is equal to the identity and η_R is an uncertainty associated to the observa-

tion, herein considered as a white Gaussian noise. The covariance matrix $E[\eta_R \eta_R^T] = R$ is estimated based on the technical specifications presented in the manuals of the GPS and the compass. Using the observation vector, $\mathbf{z}$, the next step is how to estimate the robot's pose.

B.3 Pose Estimation using Extended-Kalman Filtering

The robot's pose gives the position and the orientation of the robot in a metric referential. The metric estimated position is exclusively to plot the rawdata acquired by the sensors for display of the topological results presented through the thesis. The information retrieved by the GPS is not enough, since it is necessary to travel distances larger than 5 meters to acquire different measurements. Moreover, the orientation is also necessary to estimate the angle transition between two states, θ_{ij}.

The pose estimation is accomplished using the Kalman-Bucy Filtering approach. The kinematics resulted in (B.6) represents a non-linear system, which requires an Extended-Kalman Filtering (EKF) [63]. First, it is necessary to linearize the system, using $F(k)$ and $H(k)$ (the Hessian is obviously the identity, given the linearity of g), by

$$F(k) = \nabla f|_{\mathbf{x(k)}} = \left[\begin{array}{ccc} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{array} \right]\Big|_{\mathbf{x(k)}} = \left[\begin{array}{ccc} 1 & 0 & -V \cdot \sin\theta(k) \cdot T \\ 0 & 1 & V \cdot \cos\theta(k) \cdot T \\ 0 & 0 & 1 \end{array} \right]\Bigg|_{\mathbf{x(k)}} \tag{B.12}$$

$$H(k) = \nabla g|_{\mathbf{x(k)}} = \left[\begin{array}{ccc} \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} & \frac{\partial g}{\partial \theta} \end{array} \right]\Big|_{\mathbf{x(k)}} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right] \tag{B.13}$$

Prediction Step

$$\begin{aligned} \hat{\mathbf{x}}(\mathbf{k+1} \mid \mathbf{k}) &= \mathbf{f}(\hat{\mathbf{x}}(\mathbf{k} \mid \mathbf{k})) \\ \hat{\mathbf{z}}(\mathbf{k+1} \mid \mathbf{k}) &= \mathbf{g}(\hat{\mathbf{x}}(\mathbf{k} \mid \mathbf{k})) \end{aligned} \tag{B.14}$$

Filtering Step

$$\begin{aligned} &\hat{\mathbf{x}}(\mathbf{k+1} \mid \mathbf{k+1}) = \hat{\mathbf{x}}(\mathbf{k+1} \mid \mathbf{k}) + \mathbf{K}(\mathbf{k+1}) \left[\mathbf{z}(\mathbf{k+1}) - \hat{\mathbf{z}}(\mathbf{k+1} \mid \mathbf{k})\right] \\ &K(k+1) = P(k+1 \mid k) H^T \left[H P(k+1 \mid k) H^T + R\right]^{-1} \\ &P(k+1 \mid k) = F(k) P(k \mid k) F^T(k) + Q \\ &P(k+1 \mid k+1) = P(k+1 \mid k) \left[I - K(k+1) H\right] \end{aligned} \tag{B.15}$$

Smoothing Step

$$\begin{aligned}
&\hat{\mathbf{x}}(\mathbf{k} \mid \mathbf{T}) = \hat{\mathbf{x}}(\mathbf{k} \mid \mathbf{k}) + \mathbf{G}(\mathbf{k}) \left[\hat{\mathbf{x}}(\mathbf{k+1} \mid \mathbf{T}) - \hat{\mathbf{x}}(\mathbf{k+1} \mid \mathbf{k})\right], \quad \mathbf{k} \leq \mathbf{T} \\
&G(k) = P(k \mid k) F^T(k) P^{-1}(k+1 \mid k) \\
&P(k \mid T) = P(k \mid k) + G(k) \left[P(k+1 \mid T) - P(k+1 \mid k)\right] G^{-1}(k)
\end{aligned} \tag{B.16}$$

Initialization Step

$$\begin{aligned}
&\hat{\mathbf{x}}(\mathbf{0} \mid \mathbf{0}) = \bar{\mathbf{0}} \\
&P(0 \mid 0) = P_0 = \alpha I, \alpha \in \mathbb{R}^+
\end{aligned} \tag{B.17}$$

The algorithm based on EKF to estimate the robot's pose, illustrated in Figure B.6, is briefly resumed by the following steps:

1. Kalman Filter Initialization
2. Prediction at iteration $k+1$: $\hat{\mathbf{x}}(\mathbf{k+1} \mid \mathbf{k})$
3. If no observation jump to (4)
   (a) Filter at iteration $k+1$: $\hat{\mathbf{x}}(\mathbf{k+1} \mid \mathbf{k+1})$
   (b) Smoothing from the iteration of the last observation to k
4. $k \leftarrow k+1$

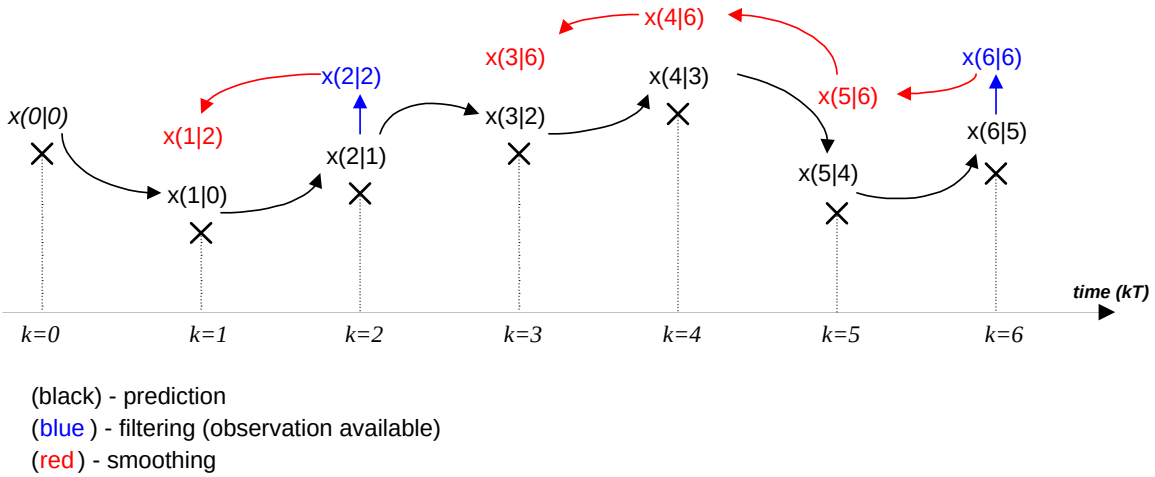


Figure B.6: Extended Kalman Filter illustration

The smoothing step of the EKF algorithm is useful to reduce the peaks caused when acquiring new measurements of GPS. The smoothing step is applied some iterations backwards. The number of iterations is equivalent to three different acquisitions of GPS.

Bibliography

[1] All About Ants. http://www.infowest.com/life/aants.htm.

[2] Honda Asimo - Humanoid. http://asimo.honda.com/.

[3] Honeybee and Robots. http://www.convict.lu/Jeunes/Navigation.htm.

[4] Kawada Industries HRP-2P - Humanoid. http://www.kawada.co.jp/ams/hrp-2/index_e.html.

[5] Mars Exploration Rovers. http://marsrovers.jpl.nasa.gov/home/index.html.

[6] Migration of Birds Orientation and Navigation. http://www.npwrc.usgs.gov/resource/othrdata/migratio/orient.htm.

[7] Omnidirectional Vision, 0-360º. http://www.0-360.com/.

[8] Sony ERS-7 Entertainment Robot AIBO. http://www.sony.net/Products/aibo/index.html.

[9] Sony QRIO - Humanoid. http://www.sony.net/SonyInfo/QRIO/top_nf.html.

[10] Ashraf Aboshosha and Andreas Zell. Disambiguating robot positioning using laser and geomagnetic signatures. Proceedings of the 8th Conference on Intelligent Autonomous Systems, Amsterdam, The Netherlands, March 2004.

[11] E. Altman. Constrained Markov Decision Processes. Chapman & Hall, 1999.

[12] Juam Andrade-Cetto and Alberto Sanfeliu. Concurrent map building and localization on indoor dynamic environment. International Journal of Pattern Recognition and Artificial Intelligence, 16(3):361 – 374, 2002.

[13] Elizeth Araújo and Roderik Grupen. Feature extraction for autonomous navigation using an active sonar head. Proceedings of the 2000 IEEE Conference on Robotics and Automation, San Francisco, 2000.

[14] R. C. Arkin. Behavior-Based Robotics. MIT Press, Cambridge, MA, 1998.

[15] David Avis and Hiroshi Imai. Locating a robot with angle measurements. Journal of Symbolic Computation, 10:311 – 326, 1990.

[16] T. Bailey and E. Nebot. Localization in large-scale environment. Robotics and Autonomous Systems, 37:261 – 281, 2001.

[17] L. Baum and J. Egon. An equality with applications to statistical estimation for probabilistic functions of markov process and to a model for ecology. Bull. Amer. Meteorol. Soc, 73:360 – 363, 1967.

[18] L. Baum and G. Sell. Growth functions for transformations on manifolds. Pacific Journal of Mathematics, 73(2):360 – 362, 1968.

[19] Niclas Bergman. Expectation maximization segmentation. (LiTH-ISY-R-2067), October 1998.

[20] Margrit Betke and Leonid Gurvits. Mobile robot localization using landmarks. IEEE Transactions on Robotics and Automation, 13(2):251 – 263, April 1997.

[21] Estela Bicho. Dynamic approach to behavior-based robotics: design, specification, analysis, simulation and implementation. Shaker Verlag, Aachen, ISBN3-8265-7462-1, 2000.

[22] Yehuda Bock and Norman Leppard. Global Positioning System: An Overview. Springer Verlag, 1990.

[23] Michael Bosse, Paul Newman, John Leonard, and Seth Teller. Slam in large-scale cyclic environments using the atlas framework. International Journal of Robotics Research, 2004.

[24] A. Briggs, C. Detweiler, P. Mullen, and D. Scharstein. Scale-space features in 1d omnidirectional images. Proceedings of Omnivis 2004 - The fifth Workshop on Omnidirectional Vision (in conjunction with ECCV 2004), pages 115 – 126, May 2004.

[25] Richard Bucy and Peter Joseph. Filtering for Stochastic Processes, with applications to Guidance. Amer Mathematical Society, 1968.

[26] Jenn Casper and Robin Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. IEEE Transactions on Systems, Man and Cybernetics Part B, 33(3):367 – 385, June 2003.

[27] Daniel Castro, Urbano Nunes, and António Ruano. Obstacle avoidance in local navigation. Proceedings of the 10th Mediterranean Conference on Control and Automation, (July), 2002.

[28] Daniel Castro, Urbano Nunes, and António Ruano. Features extraction for moving objects tracking system in indoor environments. Proceedings of The 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, July 2004.

[29] R. Chatila, S. Lacroix, and M. Devy. Autonomous outdoor mobile robot navigation. Robotics and Autonomous Systems 1997, 1997.

[30] Howie Choset. Topological simultaneous localization and mappin (slam): Toward exact localization without explicit localization. IEEE Transaction on Robotics and Automation, 17(2), April 2001.

[31] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. IEEE Transactions on Robotics and Automation, 17(3):229 – 241, June 2001.

[32] Jonathan Dixon and Oliver Henlich. Mobile robot navigation - final report. Imperial College, London - information systems engineering year 2, Surprise 1997, June 1997.

[33] D. Driankov and A. Saffiotti, editors. Fuzzy Logic Techniques for Autonomous Vehicle Navigation. Studies in Fuzziness and Soft Computing. Springer-Phisica Verlag, 2001. ISBN 3-7908-1341-9.

[34] Magnus Egerstedt. Behavior based robotics using hybrid automata. Hybrid Systems: Computation and Control, Lecture Notes in Computer Science. Springer-Verlag, pages 103 – 116, June 2000.

[35] Alberto Elfes. Occupancy Grids: A Probabilistic Framework for Robot Perception and Navi-gation. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 2001.

[36] E. Fabrizi and A. Saffioti. Extracting topology-based maps from gridmaps. Proceedings of the 2000 IEEE Conference on Robotics and Automation, San Francisco, pages 2972 – 2978, April 2000.

[37] L. Feng, J. Borenstein, and H. R. Everett. Where am i? sensors and methods for autonomous mobile robot positioning. Tech. Rep. UM-MEAM-94-21, Univ. Michigan, Ann Arbor, 1994.

[38] J. Fenwick, P. Newman, and J. Leonard. Collaborative concurrent mapping and localization. IEEE Conf on Robotics and Automation - Washington, DC, May 2002.

[39] Mark Fiala and Anup Basu. Robot navigation using panoramic landmark tracking. Proceedings of Vision Interface, Calgary, Canada, pages 117 – 124, May 2002.

[40] Graham D. Finlayson, Bernt Schiele, and James L. Crowley. Comprehensive colour image normalization. Lecture Notes in Computer Science Journal,, pages 475 – 490, 1998.

[41] J. Folkesson and H. Christensen. Graphical slam - a self-correcting map. Proceedings of IEEE International Conference on Robotics and Automation, 1:383 – 390, May 2004.

[42] M. Franz and H. Mallot. Biomimetic robot navigation. Robotics and Autonomous Systems 30, 2000.

[43] M. Franz, B. Scholkopf, H. Mallot, and H. Bulthoff. Learning view graphs for robot navigation. Autonomous Robots 5, 1998.

[44] J. Gaspar, N. Winters, and J. Santos-Victor. Vision based navigation and environmental representations with an omni-directional camera. IEEE TRA, 16(6), December 2000.

[45] A. Gelb. Applied Optimal Estimation. The M.I.T. Press, Cambridge and Massachusetts, 1999.

[46] N. Gracias and J. Santos-Victor. Underwater mosaicing and trajectory reconstruction using global alignment. IEEE OCEANS, Honolulu, November 2001.

[47] Simona E. Grigorescu, Nicolai Petkov, and Peter Kruizinga. Comparison of texture features based on gabor filters. IEEE Transactions on Image Processing, 11(10):1160 – 1167, October 2002.

[48] J. Guivant, E. Nebot, and H. Durrant-Whyte. Simultaneous localization and map bulding using natural features in outdoor environments. Proceedings of The 6th International Conference on Intelligent Autonomous Systems, Italy, 1:581 – 588, July 2000.

[49] Klaus Hansen and Jens Damgaard Andersen. Understanding the hough transform: Hough cell support and its utilization. Journal IVC, 15:205 – 218, 1997.

[50] Chris Harris and Mike Stephens. A combined corner and edge detector. Proceedings of 4th Alvey Vision Conference, pages 147 – 151, 1988.

[51] Dirk Hähnel, Dirk Schulz, and Wolfram Burgard. Mobile robot mapping in populated environments. Advanced Robotics, 17(7):579 – 598, 2003.

[52] Dirk Hähnel, Rudoph Triebel, Wolfram Burgard, and Sebastian Thrun. Map building with mobile robots in dynamic environments. Proceedings IEEE International Conference on Robotics and Automation, pages 1557 – 1563, 2003.

[53] Shigeo Hirose and Edwardo F. Fukushima. Snakes and strings: New robotic components for rescue operations. The International Journal of Robotics Research, 23:341 – 345, April 2004.

[54] Bert Holldobler and Edward Osborne Wilson. Ants. Belknap Pr, ISBN3-06740-4075-9, 1990.

[55] D. Hristu-Varsakelis and S. Andersson. Directed graphs and motion description languages for robot navigation. Proceedings of the 2002 IEEE Conference on Robotics and Automation, 2002.

[56] I. T. Jolliffe. Principal Component Analysis. Springer-Verlag, New York, 1986.

[57] J.Santos-Victor, G.Sandini, F.Curotto, and S.Garibaldi. Divergent stereo in autonomous navigation: From bees to robots. International J. Computer Vision, 14(2):159 – 177, 1995.

[58] Hyun-Deok Kang and Kang-Hyun Jo. Self-localization of mobile robot using omnidirectional vision. Proceedings 7th Korea-Russia International Symposium, 2003.

[59] W. Kasprzak and W. Szynkiewiez. Using color image features in discrete self-localization of a mobile robot. International IEEE Conference on Methods and Models in Automation and Robotics,, pages 1101 – 1106, August 2003.

[60] L. Kavraki, P. Svestka, J.C. Latombe, , and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, 12(4):566 – 580, August 1996.

[61] M. Kijima. Markov Processes for Stochastic Modeling. Chapman & Hall, 1997.

[62] Markus Knapek, Ricardo Oropeza, and David Kriegman. Selecting promising landmarks. International Conference on Robotics and Automation 2000, 2000.

[63] S. Koening and R. Simmons. Unsupervised learning of probabilistic models for robot navigation. In Proceedings of the International Conference on Robotics and Automation, pages 2301 – 2308, 1996.

[64] Kurt Koffka. Principles of Gestalt Psychology. Harcourt, 1967.

[65] Andreas Koschan. Improving robot vision by color information. Proceedings of the 7th International Conf. on Artificial Intellingence and Information-Control Systems of Robots, Smolenice Castle, Slovakia, World Scientific, Singapur, 247 - 258, September 1997.

[66] B. Kröse, R. Bunschoten N. Vlassis, and Y. Motomura. A probabilistic model for appearance-based robot localization. Image and Vision Computing, 19(5):381 – 391, April 2001.

[67] Peter Kruizinga and Nikolay Petkov. Nonlinear operator for oriented texture. IEEE Transactions on Image Processing, 8(10):1395 – 1407, October 1999.

[68] B. Kuipers. Modeling spatial knowledge. Cognitive Science, 2:129 – 153, 1978.

[69] B. Kwolek. Markov localization for mobile robots using omnidirectional vision and histogram matching. International IEEE Conference on Methods and Models in Automation and Robotics, Szczecin-Miedzyzdroje, Poland, (2), August 2003.

[70] A. Lambert and Th. Fraichard. Landmark-based safe path planning for car-like robots. Proceedings of the 2000 IEEE Conference on Robotics and Automation, San Francisco, pages 2046 – 2051, April 2000.

[71] Pierre Lamon, Björn Jensen Illah Nourbakhsh, and Roland Siegwart. Deriving and matching image fingerprint sequences for mobile robot localization. Proceedings IEEE International Conference on Robotics and Automation, Seoul, Korea,, pages 1609 – 1614, May 2001.

[72] Martin Landauer. Communication among Social Bees. Cambridge, Mass., 1971.

[73] J.C. Latombe. Robot motion planning. Kluwer Academic Publishers, Boston, MA, 1991.

[74] M. Law, A. K. Jain, and M. Figueiredo. Feature selection in mixture-based clustering. Neural Information Processing Systems, 2002.

[75] Alfred Leick. GPS Satellite Surveying, 2nd Edition [Published in English]. Wiley-Interscience, 1999.

[76] J. J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. International Journal of Robotics Research, 11(4):286 – 298, 1992.

[77] P. Lima, M. Isabel Ribeiro, Luis Custodio, and Jose Santos-Victor. The rescue project - cooperative navigation for rescue robots. Proceedings of ASER'03 - 1st International Workshop on Advances in Service Robotics, March 2003.

[78] Frederick C. Lincoln, Steven R. Peterson, and John L. Zimmerman. Migration of birds. U.S. Department of the Interior, U.S. Fish and Wildlife Service, Washington, D.C., page 113, 1998. http://www.npwrc.usgs.gov/resource/othrdata/migratio/migratio.htm.

[79] L. Liporace. Maximum likelihood estimation for multivariate observations of markov sources. IEEE Transactions on Information Theory, 28(5):729 – 734, 1982.

[80] Y. Liu and S. Thrun. Results for outdoor-SLAM using sparse extended information filters. Proceedings of The IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan, pages 1227 – 1233, September 2003.

[81] Jorge Salvador Marques. Reconhecimento de Padrões: Métodos Estatísticos e Neuronais. IST Press, 1999.

[82] Stephen Marsland. Novelty detection in learning systems. Neural Computing Surveys, 3:157 – 195, 2003.

[83] M. Mata, J. Armingol, A. de la Escalera, and M. Salichs. Using learned visual landmarks for intelligent topological navigation of mobile robots. Proceedings IEEE International Conference on Robotics and Automation,, pages 1324 – 1329, September 2003.

[84] Maja J. Mataric. Integration of representation into goal-driven behavior-based robots. IEEE Transactions on Robotics and Automation, 8(3):304 – 312, June 1992.

[85] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. pages 463 – 502, 1969. reprinted in McC90.

[86] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. Proceedings of the American Association for Artificial Intelligence National Conference on Artificial Intelligence, Edmonton, Canada, 2002.

[87] João Gomes Mota and Maria Isabel Ribeiro. Mobile robot localisation o reconstructed 3d models. Journal of Robotics and Autonomous Systems, Elsevier, 31(1-2):17 – 30, April 2000.

[88] Robin Murphy. Rescue robots at the world trade center. Journal of the Japan Society of Mechanical Engineers, special issue on Disaster Response Robotics, 102(1019):794 – 802, 2003.

[89] Robin Murphy, Christine Lisetti, Russell Tardif, Liam Irish, and Aaron Gage. Emotion-based control of cooperating heterogeneous mobile robots. IEEE Transactions on Robotics and Automation, special issue on Multi-Robot Systems, 18(5):755 – 757, October 2002.

[90] E. Nebot and S. Baiker. Autonomous navigation and map building using laser range sensors in outdoor applications. Journal of Robotic Systems, 2000.

[91] K. Ohno, T. Tsubouchi, B. Shigematsu, and S. Yuta. Differential gps and odometry-based outdoor navigation of a mobile robot. Advanced Robotics, 18(6):611 – 635, July 2004.

[92] C. Olson. Probabilistic self-localization for mobile robots. IEEE Transactions on Robotics and Automaton, 16(1), February 2000.

[93] A. Papoulis. Probability, Random Variables and Stochastic Processes. McGraw-Hill, 1991.

[94] E. Parzen. On estimation of a probability density function and mode. Annals of Mathematical Statistics, 33:1065–1076, 1962.

[95] David Pierce and Benjamin Kuipers. Learning to explore and build maps. Proceedings of American Association for Artificial Intelligence-94, AAAI/MIT Press, 1994.

[96] F. G. Pin and S. M. Killough. A new family of omnidirectional and holonomic wheeled platforms for mobile robots. IEEE Transactions on Robotics and Automation, 10:480 – 489, August 1994.

[97] Jan Puzicha, Joachim M. Buhmann, Yossi Rubner, and Carlo Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In Proceedingsthe IEEE International Conference on Computer Vision (ICCV1999), 2:1165 – 1173, 1999.

[98] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications. Proceeding of the IEEE, 77(2):257 – 286, 1989.

[99] D. Reisfeld and H. Yeshurun. Context free attentional operators: the generalized symmetry transform. International J. of Computer Vision, Special Issue on Qualitative Vision, 1994.

[100] Ioannis Rekleitis, Gregory Dudek, and Evangelos Milios. Probabilistic cooperative localization and mapping in practice. Proceedings of the 2003 IEEE International Conference on Robotics and Automation Taipei, Taiwan, pages 1907 – 1912, September 2003.

[101] Ioannis M. Rekleitis. A particle filter tutorial for mobile robot localization. (TR-CIM-04-02), 2004.

[102] S. J. Roberts and R. Hanka. An interpretation of mahalanobis distance in the dual space. Pattern Recognition, 15(4):325 – 333, 1982.

[103] N. Roy, W. Burgard, and D Fox S. Thrun. Coastal navigation - mobile robot navigation with uncertainty in dynamic environments. In Proceedings of Conference on Neural Information Processing Systems (NIPS), 1999.

[104] José Santos-Victor and Alexandre Bernardino. Vision-based navigation, environmental representations and imaging geometries. Robótica, Portugal, 47:21 – 29, 2002.

[105] Matthias Scheutz. Affective action selection and behavior arbitration for autonomous robots. Proceedings of International Conference on Artificial Intelligence, June 2002.

[106] Hagit Shatkay and Leslie Pack Kaelbling. Learning topological maps with weak local odometric information. International Joint Conference on Artificial Intelligence (IJCAI), pages 920 – 929, 1997.

[107] Randall Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. The International Journal of Robotics Research, Published by MIT Press, 5(4):56 – 68, 1987.

[108] Satoshi Tadokoro, Hiroaki Kitano, Tomoichi Takahashi, Itsuki Noda, Hitoshi Matsubara, Atsushi Shinjoh, Tetsuhiko Koto, Ikuo Takeuchi, Hironao Takahashi, Fumitoshi Matsuno, Michinori Hatayama, Jun Nobe, and Susumu Shimada. The robocup-rescue project: A robotic approach to the disaster mitigation problem. Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, San Francisco, CA, USA, pages 4089 – 4095, April 2000. http://dblp.uni-trier.de.

[109] Simon Thompson, Toshihiro Matsui, and Alexander Zelinsky. Localization using automatically selected landmarks from panoramic images. ACRA 2000, 2000.

[110] S. Thrun. Bayesian landmark learning for mobile robot localization. Machine Learning, 33(1):41 – 76, 1998.

[111] S. Thrun. Probabilistic algorithm in robotics. Artificial Intelligence Magazine, 21(4):93 – 109, 2000.

[112] S. Thrun and A. Bücken. Integrating grid-based and topological maps for mobile robot navigation. Proceedings of the Thirteenth National Conference on Artificial Intelligence American Association for Artificial Intelligence, Portland, Oregon, 1996.

[113] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. Machine Learning, 31:29 – 53, 1998.

[114] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000.

[115] Iwan Ulrich and Illah Nourbakhsh. Appearance-based place recognition for topological localization. Proceedings IEEE International Conference on Robotics and Automation, San Francisco,, pages 1023 – 1029, April 2000.

[116] A. Vale, N. Antunes, P. Inácio, M. Ribeiro, and J. Sequeira. Remote operation and internet. 4th Portuguese Conference on Automatic Control, Guimarães, Portugal, October 2000.

[117] A. Vale, José Miguel Lucas, and M. I. Ribeiro. Feature extraction and selection for mobile robot navigation in unstructured environments. Proceedings of The 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, July 2004.

[118] A. Vale and M. I. Ribeiro. A probabilistic approach for the localization of mobile robots in topological maps. Proceedings of the 10th IEEE Mediterranean Conf. on Control and Automation - MED2002, Lisboa, Portugal, 2002.

[119] A. Vale and M. I. Ribeiro. Environment mapping as a topological representation. Proceedings of the 11th International Conference on Advanced Robotics, Coimbra, Portugal, 1:29 – 34, June 2003.

[120] A. Vale, J. Simões, J. Machado, and P. Lima. Navigation in a maze. Robótica, Portugal, 45:66 – 70, October 2001.

[121] Sjoerd van der Zwaan, A. Bernardino, and J. Santos-Victor. Vision based station keeping and docking for floating vehicles. ECC2001, Porto, Portugal, September 2001.

[122] Nikos Vlassis, Yoichi Motomura, Isao Hara, Hideki Asoh, and Toshihiro Matsui. Edge-based features from omnidirectional images for robot localization. Proceedings IEEE International Conference on Robotics and Automation, Seoul, Korea, pages 1579 – 1584, May 2001.

[123] Thomas Wachtler, Te-Won Lee, and T. Sejnowski. The chromatic structure of natural scenes. Journal of the Optical Society of America, 18(1):65 – 77, January 2001.

[124] Frank Wallner. Position Estimation for a Mobile Robot From Principal Component of Laser Range Data. Thèse préparée au sein du laboratoire GRAVIR - IMAG de l'Institut National Polytechnique de Grenoble, France, 1997.

[125] Frank Wallner, Bernt Schiele, and James L. Crowley. Position estimation for a mobile robot from principal components of laser range data. Proceedings IEEE International Conference on Robotics and Automation, Leuven, Belgium,, pages 3121 – 3128, May 1998.

[126] R. Wehner and S. Wehner. Insect navigation:use of maps or ariadne's thread? Ethology, Ecology, Evolution, (2):27 – 48, 1990.

[127] C. Weisbin, G. Rodriguez, P. Schenker, H. Das, S. Hayati, E. Baumgartner, M. Maimone, I. Nesnas, and R. Volpe. Autonomous rover technology for mars sample return. Proceedings of the International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS '99), June 1999.

[128] N. Winters and J. Santos-Victor. Information sampling for vision-based robot navigation. Robotics and Autonomous Systems, 41:145 – 159, 2002.

[129] Michael Wohlfart. Hough transform applications in computer graphics (with focus on medical visualization).

[130] Xiang Sean Zhou and Thomas S. Huang. Edge-based structural features for content-based image retrieval. Pattern Recognition Letters, pages 457 – 468, April 2001.

[131] G. Zunino and H. I. Christensen. Simultaneous localisation and mapping in domestic environments. Multisensor Fusion and Integration for Intelligent Systems, pages 67 – 72, 2001.

List of Figures

List of Tables