Universidad de Alcalá
COMISIÓN DE ESTUDIOS OFICIALES
DE POSGRADO Y DOCTORADO

ACTA DE EVALUACIÓN DE LA TESIS DOCTORAL

(FOR EVALUATION OF THE ACT DOCTORAL THESIS)

Año académico (*academic year):* 2016/17

DOCTORANDO *(candidate PHD)*: **LLAMAZARES LLAMAZARES, ÁNGEL**
D.N.I./PASAPORTE (*Id.Passport*): ******9717S**
PROGRAMA DE DOCTORADO *(Academic Committee of the Programme)*: **D441**
DEPARTAMENTO DE *(Department)*: **ELECTRÓNICA**
TITULACIÓN DE DOCTOR EN *(Phd title)*: **DOCTOR/A POR LA UNIVERSIDAD DE ALCALÁ**

En el día de hoy 13/07/17, reunido el tribunal de evaluación, constituido por los miembros que suscriben el presente Acta, el aspirante defendió su Tesis Doctoral **con Mención Internacional** (*In today assessment met the court, consisting of the members who signed this Act, the candidate defended his doctoral thesis with mention as International Doctorate),* elaborada bajo la dirección de (*prepared under the direction of)* MANUEL OCAÑA MIGUEL // .

Sobre el siguiente tema *(Title of the doctoral thesis)*: *LASER-BASED DETECTION AND TRACKING OF MOVING OBSTACLES TO IMPROVE PERCEPTION OF UNMAMMED GROUND VEHICLES*

Finalizada la defensa y discusión de la tesis, el tribunal acordó otorgar la CALIFICACIÓN GLOBAL[1] de (**no apto, aprobado, notable** y **sobresaliente)** *(After the defense and defense of the thesis, the court agreed to grant the GLOBAL RATING (fail, pass, good and excellent)*: SOBRESALIENTE

Alcalá de Henares, a 13 de Julio de 2017

Fdo. *(Signed)*: Luis Magdalena Fdo. *(Signed)*: Vicente Milanés Fdo. *(Signed)*: Miguel A. García

FIRMA DEL ALUMNO (candidate's signature),

Fdo. *(Signed)*: Ángel Llamazares

Con fecha 24 de julio de 2017 la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado, a la vista de los votos emitidos de manera anónima por el tribunal que ha juzgado la tesis, resuelve:

☒ Conceder la Mención de "Cum Laude"
☐ No conceder la Mención de "Cum Laude"

La Secretaria de la Comisión Delegada

[1] La calificación podrá ser "**no apto**" "**aprobado**" "**notable**" y "**sobresaliente**". El tribunal podrá otorgar la mención de "cum laude" si la calificación global es de sobresaliente y se emite en tal sentido el voto secreto positivo por unanimidad. (*The grade may be "**fail**" "**pass**" "**good**" or "**excellent**". The panel may confer the distinction of "cum laude" if the overall grade is "Excellent" and has been awarded unanimously as such after secret voting.).*

UNIVERSIDAD DE ALCALÁ. PATRIMONIO DE LA HUMANIDAD

INCIDENCIAS / OBSERVACIONES:
(Incidents / *Comments)*

Universidad de Alcalá

COMISIÓN DE ESTUDIOS OFICIALES
DE POSGRADO Y DOCTORADO

En aplicación del art. 14.7 del RD. 99/2011 y el art. 14 del Reglamento de Elaboración, Autorización y Defensa de la Tesis Doctoral, la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado y Doctorado, en sesión pública de fecha 24 de julio, procedió al escrutinio de los votos emitidos por los miembros del tribunal de la tesis defendida por *LLAMAZARES LLAMAZARES, ÁNGEL*, el día 13 de julio de 2017, titulada *LASER-BASED DETECTION AND TRACKING OF MOVING OBSTACLES TO IMPROVE PERCEPTION OF UNMAMMED GROUND VEHICLES*, para determinar, si a la misma, se le concede la mención "cum laude", arrojando como resultado el voto favorable de todos los miembros del tribunal.

Por lo tanto, la Comisión de Estudios Oficiales de Posgrado **resuelve otorgar** a dicha tesis la

MENCIÓN "CUM LAUDE"

Alcalá de Henares, 27 julio de 2017
EL PRESIDENTE DE LA COMISIÓN DE ESTUDIOS
OFICIALES DE POSGRADO Y DOCTORADO

Firmado digitalmente por VELASCO PEREZ JUAN RAMON - DNI 03087239H
Fecha: 2017.07.30 18:24:18 +02'00'

Juan Ramón Velasco Pérez

Copia por e-mail a:
Doctorando: LLAMAZARES LLAMAZARES, ÁNGEL
Secretario del Tribunal: MIGUEL ÁNGEL GARCÍA GARRIDO.
Director de Tesis: MANUEL OCAÑA MIGUEL

UNIVERSIDAD DE ALCALÁ. PATRIMONIO DE LA HUMANIDAD

Universidad de Alcalá

ESCUELA DE DOCTORADO
Servicio de Estudios Oficiales de Posgrado

DILIGENCIA DE DEPÓSITO DE TESIS.

Comprobado que el expediente académico de D./Dª ÁNGEL LLAMAZARES LLAMAZARES reúne los requisitos exigidos para la presentación de la Tesis, de acuerdo a la normativa vigente, y habiendo presentado la misma en formato: ☒ soporte electrónico ☐ impreso en papel, para el depósito de la misma, en el Servicio de Estudios Oficiales de Posgrado, con el nº de páginas: 242 se procede, con fecha de hoy a registrar el depósito de la tesis.

Alcalá de Henares a 28 de ABRIL de 2017

Rosa Carrillo Salas

Fdo. El Funcionario

Universidad de Alcalá

PhD. Program in Electronics: Advanced Electronic Systems. Intelligent Systems

Laser-Based Detection and Tracking of Moving Obstacles to Improve Perception of Unmanned Ground Vehicles

PhD. Thesis presented by

Angel Llamazares Llamazares

2017

Universidad de Alcalá

PhD. Program in Electronics: Advanced Electronic Systems. Intelligent Systems

Laser-Based Detection and Tracking of Moving Obstacles to Improve Perception of Unmanned Ground Vehicles

PhD. Thesis presented by

Angel Llamazares Llamazares

Advisor

Dr. Manuel Ocaña Miguel

Alcalá de Henares, 2017

Universidad de Alcalá

DEPARTAMENTO DE ELECTRÓNICA
Escuela Politécnica
Campus Universitario s/n
28805 Alcalá de Henares (Madrid)
Teléfono: 91 885 65 40
Fax: 91 885 65 91
dpto.electronica@uah.es

Dr. D. Manuel Ocaña Miguel, Profesor Titular de la Universidad de Alcalá

INFORMA:

Que la Tesis Doctoral titulada "Laser-Based Detection and Tracking of Moving Obstacles to Improve Perception of Unmanned Ground Vehicles", presentada por D. Ángel Llamazares Llamazares, y realizada bajo mi dirección, dentro del campo de los sistemas de navegación autónoma, reúne los méritos de calidad y originalidad para optar al Grado de Doctor.

Alcalá de Henares, a 20 de abril de 2017.

UNIVERSIDAD DE ALCALÁ DEPARTAMENTO DE ELECTRÓNICA EDIFICIO POLITÉCNICO

Fdo.: Dr. D. Manuel Ocaña Miguel.

UNIVERSIDAD DE ALCALÁ. PATRIMONIO DE LA HUMANIDAD

Universidad de Alcalá

DEPARTAMENTO DE ELECTRÓNICA
Escuela Politécnica
Campus Universitario s/n
28805 Alcalá de Henares (Madrid)
Teléfono: 91 885 65 40
Fax: 91 885 65 91
dpto.electronica@uah.es

Dr. Dña. Sira Elena Palazuelos Cagigas, Directora del Departamento de Electrónica de la Universidad de Alcalá

INFORMA: Que la Tesis Doctoral titulada "Laser-Based Detection and Tracking of Moving Obstacles to Improve Perception of Unmanned Ground Vehicles", presentada por D. Ángel Llamazares Llamazares, y realizada bajo la dirección del Dr. D. Manuel Ocaña Miguel, dentro del campo de los sistemas de navegación autónoma, reúne los méritos de calidad y originalidad para optar al Grado de Doctor.

Alcalá de Henares, a 20 de abril de 2017.

Fdo.: Dr. Dña. Sira Elena Palazuelos Cagigas.

UNIVERSIDAD DE ALCALÁ. PATRIMONIO DE LA HUMANIDAD

"Kites rise highest
against the wind, not with it"
Winston Churchill

Agradecimientos

Gracias a ti, si a ti, lector de estas líneas, porque si estás leyendo este libro seguro que mereces que te lo agradezca, en mayor o menor medida, no lo sé... pero estoy convencido de que tienes algo que ver en que haya conseguido esta meta, sí esta carrera de fondo. Espero haber sabido agradecértelo, en palabras o en hechos... según el momento, si no ha sido así, aprovecho esta ocasión,

Una vez más, Gracias.
Llamazares.

Resumen

El objetivo de esta tesis es desarrollar un sistema que mejore la etapa de percepción de vehículos terrestres no tripulados *(UGVs)* heterogéneos, consiguiendo con ello una navegación robusta en términos de seguridad y ahorro energético en diferentes entornos reales, tanto interiores como exteriores. La percepción debe tratar con obstáculos estáticos y dinámicos empleando sensores heterogéneos, tales como, odometría, sensor de distancia láser *(LIDAR)*, unidad de medida inercial *(IMU)* y sistema de posicionamiento global *(GPS)*, para obtener la información del entorno con la precisión más alta, permitiendo mejorar las etapas de planificación y evitación de obstáculos.

Para conseguir este objetivo, se propone una etapa de mapeado de obstáculos dinámicos *(DOMap)* que contiene la información de los obstáculos estáticos y dinámicos. La propuesta se basa en una extensión del filtro de ocupación bayesiana (BOF) incluyendo velocidades no discretizadas. La detección de velocidades se obtiene con Flujo Óptico sobre una rejilla de medidas *LIDAR* discretizadas. Además, se gestionan las oclusiones entre obstáculos y se añade una etapa de seguimiento multi-hipótesis, mejorando la robustez de la propuesta *(iDOMap)*.

La propuesta ha sido probada en entornos simulados y reales con diferentes plataformas robóticas, incluyendo plataformas comerciales y la plataforma *(PROPINA)* desarrollada en esta tesis para mejorar la colaboración entre equipos de humanos y robots dentro del proyecto *ABSYNTHE*. Finalmente, se han propuesto métodos para calibrar la posición del *LIDAR* y mejorar la odometría con una *IMU*.

Palabras clave: DATMO, Mapeado de obstáculos dinámicos, BOF

Abstract

The goal of this thesis is to develop a system that improves the perception stage of heterogeneous *Unmanned Ground Vehicles (UGVs)* in order to achieve a robust navigation in terms of safety and energy saving in several real indoor and outdoor environments. The perception must deal with static and dynamic obstacles using heterogeneous sensors, such as, the odometry, *LIght Detection And Ranging (LIDAR)*, *Inertial Measurement Unit (IMU)* and *Global Positioning System (GPS)*, in order to collect the surrounding information with the highest precision, allowing to improve the planning and obstacle avoidance stages.

To achieve this objective, a *Dynamic Obstacles Mapping* approach *(DOMap)* is proposed to obtain the local map with static and dynamic obstacles information. The proposal is based on the *Bayesian Occupancy Filter (BOF)*, extended in order to not assuming discretized velocities. The velocities detection has been obtained using Optical Flow over a discretized grid of *LIDAR* measurements. In addition, the occlusions between obstacles had been handled and a multi-hypothesis tracking stage has been added, improving the robustness of the proposal. Therefore, the obtained map has information about occupancy and velocities of the surrounding obstacles *(iDOMap)*.

The proposal has been tested in simulated and real scenarios with different robotic platforms, including commercials ones and the developed *PROPINA* platfom to improve collaboration between humans and robots teams within *ABSYNTHE* project. Finally, *LIDAR* pose calibration and an improved odometry with *IMU* methods have been proposed.

KeyWords: DATMO, Dynamic Obstacles Mapping, BOF

Table of Contents

List of Figures

List of Tables

List of Acronyms

ABSYNTHE	Abstraction, Synthesis and Integration of Information for Human-Robot Teams.
ADAS	Advanced Driver Assistance Systems.
AHRS	Attitude and Heading Reference Systems.
AICO	Approximate Inference Control.
AMCL	Adaptive Monte Carlo Localization.
ARIA	Advanced Robot Interface for Applications.
BCM	Beam-Curvature Method.
BOF	Bayesian Occupancy Filter.
BOFUG	Bayesian Occupancy Filter Using Groups.
BOFUM	Bayesian Occupancy grid Filter for dynamic environments Using prior Map knowledge.
CARMEN	Carnegie Mellon Robot Navigation Toolkit.
CPR	Cycles Per Revolution.
CRF	Conditional Random Field.
CVM	Curvature-Velocity Method.
DATMO	Detection and Tracking of Moving Obstacles.
DCVM	Dynamic Curvature-Velocity Method.
DDMCMC	Data-driven Markov Chain Monte Carlo.
DGPS	Differential Global Positioning System.
DOMap	Dynamic Obstacles Map.

DW4DO	Dynamic Window For Dynamic Obstacles.
DWA	Dynamic Window Approach.
EGBIS	Efficient Graph-Based Image Segmentation.
EKF	Extended Kalman Filter.
EM	Expectation-Maximization.
EMST	Euclidean Minimum Spanning Tree.
FCTA	Fast Clustering-Tracking Algorithm.
FOV	Field Of View.
GNN	Global Nearest Neighbor.
GPS	Global Positioning System.
GUI	Graphical User Interface.
HF	Histogram Filter.
HMI	Human-Machine Interface.
HSBOF	Hybrid Sampling Bayesian Occupancy Filter.
ICP	Iterative Closest Point.
IDC	Iterative Dual Correspondence.
iDOMap	improved Dynamic Obstacles Map.
IFR	International Federation of Robotics.
IMM	Interactive Multiple Model.
IMU	Inertial Measurement Unit.
INS	Inertial Navigation System.
IPAB	Institute of Perception, Action and Behaviour.
JCBB	Joint Compatibility Branch and Bound.
JPDAF	Joint Probabilistic Data Association Filter.
KF	Kalman Filter.

LCM	Lane-Curvature Method.
LIDAR	LIght Detection And Ranging.
LOP	Linear Opinion Pool.
LOS	Line-Of-Sight.
LTS	Long Time Support.
MCMC	Markov Chain Monte Carlo.
MEMS	Microelectromechanical Systems.
MHT	Multiple hypothesis tracking.
MRDS	Microsoft Robotics Development Studio.
ND	Nearness Diagram.
NED	North East Down.
NNR	Nearest Neighbor Rule.
OSRF	Open Source Robotics Foundation, Inc..
PDF	probabilistic distribution function.
PF	Particle Filter.
PHD	Probability Hypothesis Density.
PID	Proportional-Integral-Derivative.
POMDP	Partially Observable Markov Decision Process.
PROPINA	Plataforma RObotica Para la INvestigAción.
PWM	Pulse-Width Modulation.
RANSAC	RANdom SAmple Consensus.
RFID	Radio Frequency IDentification.
RoboShop	ROBOtic Guide for SHOP.
ROS	Robot Operating System.
RSSI	Received Signal Strength Intensity.
SJPDAF	Sample-based Joint Probabilistic Data Association Filters.

SLAM	Simultaneous Localization And Mapping.
SMC-BOF	Sequential Monte Carlo Bayesian Occupancy Filter.
STDR	Simple Two Dimensional Robot Simulator.
tf	Transform.
UGV	Unmanned Ground Vehicle.
UKF	Unscented Kalman Filter.
UMBmark	University of Michigan Benchmark.
VFH+	Vector Field Histogram +.
VOXEL	VOLumetric piXEL.

Chapter 1

Introduction

1.1 Motivation

Autonomous robots or vehicles have nowadays become popular for applications such as surveillance and passengers transport. In both cases, the safety and efficiency of these systems are depending on the ability of the autonomous navigation system to deal with unpredictable dynamic changes in the environment. Robots are ubiquitous in science, industry, and security. Their consideration employing mathematical and computational techniques has generally been unfeasible until recently, not only because of lack of analytical tools, but also because of the need to deploy large numbers of sensing, computing, communication, and actuation devices. Due to these devices have become less expensive and more available, robots have been considered and deployed in an ever larger number of situations.

The robot sales in 2015 increased to the highest level ever recorded, by 15% to 253,748 units, shown in Figure 1.1. Industry was the main buyer of robot in 2015, increasing 33% compared to 2014. In detail, electronic industry increased 41%, the metal industry 39%, the plastics, rubber and chemical industry 16%. On the other hand, in 2015 the automotive industry only increased in a moderate way compared to the last five years. Furthermore, the forecast of the *International Federation of Robotics (IFR)* until 2019 is that the industrial robot's worldwide sales increase 13% per year, reach-

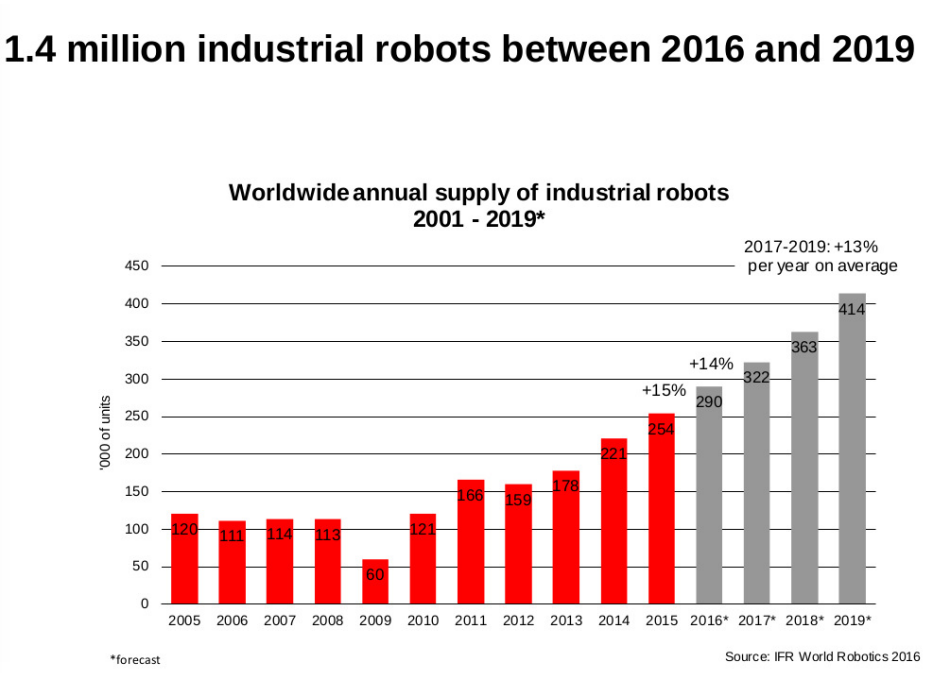


Figure 1.1: Industrial Robot's Worldwide Sales.

ing about 413,000 units in 2019. That means, from 2016 to 2019 1.4 million new industrial robots will be installed (Figure 1.1)

So far, service robots for personal and domestic use are mainly in the areas of domestic (household) robots, which include vacuum and floor cleaning [iRobot, 2002, iRobot, 2007], lawn-mowing robots [Husqvarna, 2008], and entertainment and leisure robots [Sony, 1999], including toy robots, hobby systems, education and research (LEGO® Mindstorms® [Lego, 1998]). But now, the services robot are becoming a booming segment of the global robotics industry with a 20% grow rate every year (Figure 1.2), taking into account the two main segments of the service robot:

- **Professional service Robot:** in 2015 increased 25% the sales, reaching 41060 units, compared to 2014. The forecast for this segment in the period 2016-2019 is that increase to about 333200 units.

- **Personal service robot:** in 2015, about 5.4 million of this robots were sold, increasing 16 % compared to 2014. The forecast for this segment in the period 2016-2019 is that it will increase to about 42 million units.

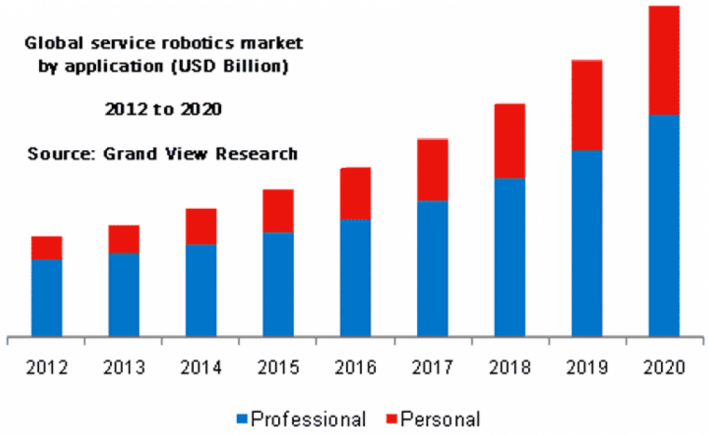


Figure 1.2: Service Robot's Worldwide Sales.

Figure 1.3: Autonomous Industrial and Services Robots.

Mobile robots can be used for a number of different applications as mentioned before. However, independently of the application, they always have to navigate in their environment. Localization and mapping are the essence of successful navigation in mobile platform technology. They are commonly related to cartography, combining

science, technique and computation to build a trajectory map that reality can be modelled in ways that communicate spatial information effectively. Although, both problems are deeply interconnected, they can be formulated independently. Robot localization is the problem of estimating the robot's coordinates relative to a map of its environment using its sensor data [Fox et al., 2000]. Nevertheless, Robotic mapping addresses the problem of acquiring spatial models of physical environments relative to the robot location [Thrun, 2002].

Mobile robots need sensors and actuators in order to accomplish navigation tasks. Sensors are used to obtain information about its surroundings and actuators for locomotion and manipulation tasks. Moreover, they must be able to process all the sensor's information to command the actuators typically by means of on-board computers. The generic robot system diagram is shown in Figure 1.4.

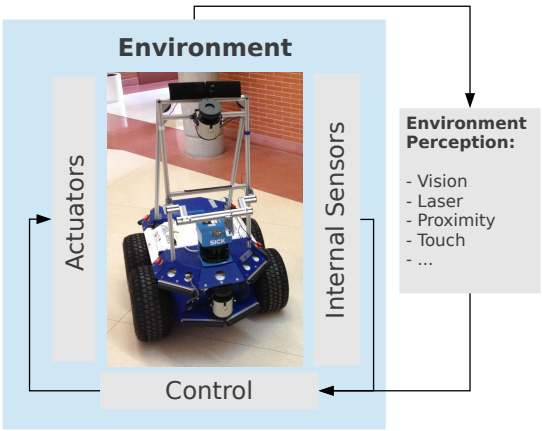


Figure 1.4: Robot system diagram.

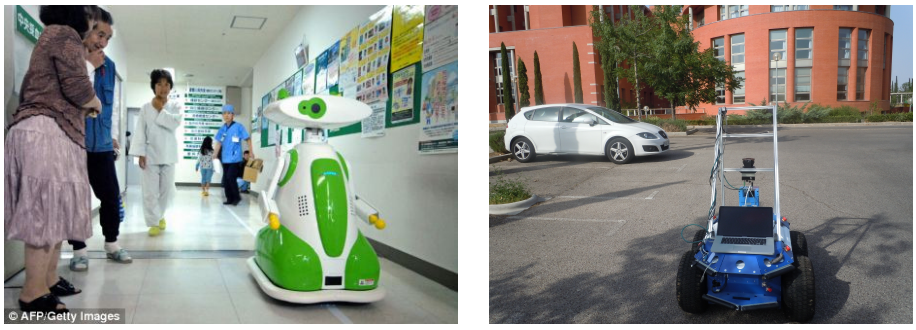
The great progress on service robotics and computing software in the last decades has increased the demand for autonomous navigation and guidance applications. Navigation and guidance are defined as the processes of determining and controlling the position of a vehicle or the so-called localization and mapping problems. These problems address the questions *"Where am I?"* and *"What does the world look*

like?" which are key task for every mobile robot. The localization stage can be greatly improved by using *Global Positioning System (GPS)* or its enhanced version: *Differential Global Positioning System (DGPS)* [Enge and Misra, 1999]. This provides a global reference for vehicle with the accuracy of a few centimetres in the case of DGPS. However, when a GPS receiver works in an urban environments with high buildings or underground tunnels, the signal can suffer multipath fading or even *Line-Of-Sight (LOS)* blockage which renders this sensor inoperative. However, GPS is generally not suitable to establish indoor navigation systems, since microwaves will be attenuated and scattered by roofs, walls and other objects.

The autonomous navigation systems have been studied extensively in the literature [Montemerlo et al., 2008] [Thrun et al., 2006]. Such systems should be able to perform perception, localization, planning and actuation stages in a correct way. To perceive the state of the environment the on-board sensors are used. The localization stage is usually based on a fusion of sensory information and a priori map [Schleicher et al., 2010] [Hentschel et al., 2008] to determine the location of the vehicle within the global frame. Such maps are usually obtained in a semi-autonomous way [Thrun et al., 1998] or using *Simultaneous Localization And Mapping (SLAM)* techniques [Newman et al., 2006] [Schleicher et al., 2010] [Bekris et al., 2006] to reduce the uncertainty of localization and mapping processes by doing both at the same time. Once the vehicle's location has been determined, the sequence of actions necessary to reach the goal can be computed within the planning stage. The resulting plan is usually a mesh of connected way points that has to satisfy various constraints such as: holonomic constraints, safety, traffic rules, energy efficiency, etc. Finally, the actuation stage is responsible for executing the plan.

In addition to localization and planning, a robust autonomous mobile platform requires an obstacle avoidance system. Such system ensures that a vehicle navigates safely around the obstacles while trying to reach its global goal. Obstacle avoidance can be divided into global and local approaches. While the former approach assumes a complete model of the environment, such as the potential field methods, local methods require only partial observability of the envi-

ronment at the cost of guaranteeing only local optimality. However, computational cost is much lower for local methods and they can be often implemented in the form of reactive controllers. These reactive methods take control of the robot when an obstacle is detected to prevent collision. They use the nearest portion of the environment modelled using the current sensor observation. Some representative examples are *Vector Field Histogram + (VFH+)* [Ulrich and Borenstein, 1998], *Nearness Diagram (ND)* [Minguez and Montano, 2004], *Curvature-Velocity Method (CVM)* [Fox et al., 1997], its improved version, *Lane-Curvature Method (LCM)* [Ko and Simmons, 1998] and the *Beam-Curvature Method (BCM)* [Fernández et al., 2004].



(a) Service robot in a hospital. (b) Security robot in a car park.

Figure 1.5: Robots in dynamic environments.

One of the major drawbacks of the reactive methods is that they do not take into account the dynamic changes of the environment and assume that all obstacles are static. Therefore, they can not predict their motion. This is, however, an unrealistic assumption especially when the vehicle deals with a high uncertainty over the position, shape and velocity of the obstacles and it is still a challenge for real world applications [Coué et al., 2006] [Fulgenzi et al., 2007]. It is especially important when a service robot is applied to cluttered or crowed environment, such as a hospital (Fig. 1.5(a)) or the surveillance of a car park (1.5(b)).

Furthermore, this is important in terms of safety and energy saving when an autonomous vehicle is moving in the real word, because

knowing the dynamic changes around the vehicle, the algorithms can calculate the optimal path to achieve the goal saving energy and in the most safety way. To reach these objectives there are different stages to improve:

- **Perception**: the representation and modelling of the environment using the information from different sensors.
- **Localization**: the obtaining the position of the robot in the environment.
- **Planning**: the computation the paths or routes to follow to achieve the goal.
- **Control**: the execution of the commands needed to follow the paths in a proper way.

In the last years, one of the most important breakthroughs in autonomous navigation for mobility robots has been Boss (Figure 1.6) which is an autonomous vehicle created by Carnegie Mellon University's Tartan Racing Team [Tartan Racing, 2007]. Boss won the 2007 DARPA Urban Challenge [Urmson et al., 2008], a prize competition for driverless vehicles in a urban scenario dealing with other vehicles and following the traffic rules. To navigate, Boss used twelve lasers, cameras and radars to obtain information of the environment, supplementing the position sensing GPS system. A three-layer planning system combines mission, behavioural, and motion planning to drive in urban environments. The mission planning layer considers which street to take to achieve a mission goal. The behavioural layer determines when to change lanes and precedence at intersections and performs error recovery manoeuvres. The motion planning layer selects actions to avoid obstacles while making progress toward local goals.

In that challenge the environment is fixed and controlled in some way, because the path is almost always delimited by curb of the road, the traffic signals give information about the environment, where is the robot and the theoretical behaviour of the vehicles around the robot. For that reasons there are some advantages respect to the

general cases where the autonomous robot has to deal with a irregular, unpredictable and dynamic environment.

Figure 1.6: Boss: Autonomous vehicle.

1.2 Scope

This thesis is placed within several projects. The first of them, is the *Plataforma RObotica Para la INvestigAción (PROPINA)* project, that is the developed of our own robotic platform, from scratch to end in a real hardware platform, based on the experience of the RobeSafe Research Group with commercial robotic platforms, with the objectives of reducing its cost and improving their weaknesses. The second one, is the *ROBOtic Guide for SHOP (RoboShop)* project, to create a complete robot navigation system, that can be used as shop assistance or a robotic guide in a museum or in a mall, outfitting the PROPINA platform with more sensors to measure the environment, such as *LIght Detection And Ranging (LIDAR)*, *Inertial Measurement Unit (IMU)*, depth camera, RGB camera and a touch screen as a *Human-Machine Interface (HMI)*. The third one is the *Abstraction, Synthesis and Integration of Information for Human-Robot Teams (ABSYNTHE)* project [Alonso et al., 2012] which main aims are the development of concepts, tools, and approaches for the collaborative production and application of high-level semantic descriptions of computational objects with a view to facilitate the joint intelligent

processing and exploitation of knowledge by mixed teams of human and robots. This project needs to deal with the following problems and issues:

- **Qualitative Object Description**

  Effective collaboration between team members requires the ability to quickly exchange, at various levels of abstraction, salient characteristics of the data representing complex conceptual structures.

- **Cooperative robotics**

  The interaction of robots with their surroundings requires the acquisition and representation of objects contained within their immediate neighbourhood. As is also the case with humans interacting with their environment, this representation requires the mapping of perceptual information (e.g., images, radar returns) into symbols describing environmental features (e.g., walls, tables, obstacles) through a cognitive process known as anchoring.

- **Autonomous navigation in complex environments**

  Autonomous robots cannot currently emulate the ability of humans to navigate through a wide variety of environments. To attain a similar level of proficiency and ability to adapt to new situations it is necessary to develop high-level knowledge capable of being employed to rapidly produce tactics applicable to new navigational challenges (e.g., adapt 3D traversing techniques to new environments).

- **Integration of high-level sensor information**

  In collaborative architectures such as those of human-robot teams, perceptual information is typically collected by a variety of distributed, multi-modal, sensors of different characteristics that produce evidential data at multiple levels of resolution. The development of techniques mapping between these diverse descriptions and integrating (fusing) information as a function of its expected usage is a major need.

This thesis is focused on this last two problems. To achieve an autonomous navigation while the robots are in the same scenario that humans or other vehicles, whether indoor or outdoor it is mandatory to detect them in the best way and with the most accurate and rich information, as their positions, movements, directions and velocities. Additionally, this accurate and rich information is not only useful for the robot, also it is very interesting for the people in the same scenario in order to improve their knowledge of the environment.

1.3 Proposal

From 2004, members of RobeSafe Research Group[1] at the Department of Electronics have been working in the problem of autonomous navigation [Ocaña et al., 2004] [López, 2004] [Ocaña, 2005]. Several important results were achieved in this area, more precisely in indoor environments with WiFi, a robot navigation system based on signal strength and ultrasounds was developed [Ocaña et al., 2005] or a robot navigation system based on *Partially Observable Markov Decision Process (POMDP)* and using vision and proximity sensors [López, 2004].

RobeSafe Research Group has focused its efforts on some important aspects in order to develop robot navigation systems. RobeSafe Research Group is interested in developing non-invasive systems, which means to use the environment infrastructure without adding extra devices or technologies. Finally, developing solutions for the real world always is a premise for RobeSafe Research Group.

The aim of this thesis is to develop a system to improve the perception stage of heterogeneous robots in order to achieve a robust navigation in terms of safety and energy saving in multiples real scenarios, such as indoor and outdoor environments. The perception must deal with static and dynamic obstacles using heterogeneous sensors, such as the odometry, LIDAR, IMU and GPS in order to collect the surrounding information with the highest precision in real-time, allowing improvements to the planning and obstacle avoidance algorithms.

[1] www.robesafe.com

1.4 Document structure

After the introduction in Chapter 1, Chapter 2 contains a brief review of the most significant research on *Detection and Tracking of Moving Obstacles (DATMO)*.

In Chapter 3 some of the commercial robotic platforms, commonly used in research, have been analysed. Also, the robotic platform developed in this thesis has been explained.

Chapter 4 presents the *Dynamic Obstacles Mapping Proposal*. Results for experiments under simulated and real conditions are presented and discussed.

Chapter 5 presents the final application results of applying the best algorithms of previous chapters.

Finally, Chapter 6 contains the conclusions and main contributions of this work, and future research lines that may spring from it.

Chapter 2

State of the Art

Before planning stage can be performed, the robot needs to know the environment where it moves. For that reason the perception and mapping stages must be performed previously. This chapter presents and introduces the state of the art in perception and mapping based on on-board sensors and a brief survey of the state of the art in DATMO is introduced. The aim of what follows is to provide an overview of the most remarkable methods at each field specially in indoor environments, and those that are related to the contents of the following sections of this thesis. On those sections, related methods to the presented proposal will be explained in more detail. This chapter closes with a discussion on the most adequate methods to be studied for the intended application of this work, and the specific aims of this thesis.

2.1 Perception and Mapping

On-board sensors are used to perceive the state of the environment and provide information to the mapping stage, that are usually included in the robot platforms. Then, the local and global planning stages can be perform. This system has to deal with the problem of estimating the robot's surrounding by the information from the sensors and sometimes, using a map of its environment. The on-board sensors commonly used in robotics are:

- ***Odometers***: allow to count the revolutions of the robot wheels, is the most common sensors used in the relative positioning.

- ***Inertial Measurement Unit (IMU)***: composed by accelerometers, gyroscopes and magnetometers. Is usually added to the robotic platforms in order to perform a *Inertial Navigation System (INS)* to improve the odometry.

- ***Range sensors:*** from different technologies, such as *infrared*, *sonar* or LIDAR that provides distance measurements in order to collect information about the surrounding environment.

The sensors and actuators help navigation tasks, however, both of them have to deal with several issues [Thrun, 2002]. Sensors are subject to errors, often referred to as measurement noise, and strict range limitations. Even, the robot motion is influenced by errors too. Hence, the controls alone are insufficient to determine a robot's pose (location and orientation) relative to its environment. For that reason, there are some challenges to overcome:

- **Measurement noise**: modelling problems, are usually relatively easy to solve if the noise in different measurements is statistically independent. Unfortunately, in robotics, the measurement errors are statistically dependent. This is because errors in control accumulate over time, and they affect the way future sensor measurements are interpreted.

- **High dimensionality of the maps**: a detailed two-dimensional floor plan, which is an equally common representation of robotic maps, often requires thousands of characteristics. But a detailed 3D visual map of a building may easily require millions of characteristics.

- **Data association problem**: determining if sensor measurements taken at different points in time correspond to the same physical object in the world.

- **Changing environments**: some changes may be relatively slow, such as the structural changes in buildings. Others are faster, such as the change of door status or the location of furniture items. Unfortunately, there are almost no mapping algorithms that can learn meaningful maps of dynamic environments. Instead, the predominant paradigm relies on a static world assumption, in which the robot is the only time-variant quantity (and everything else that moves is just noise). Consequently, most techniques are only applied in relatively short time windows, during which the respective environments are static.

The majority of researchers in the literature use a probabilistic framework to deal with those issues. They all employ probabilistic models of the robot and its environment, and they all rely on probabilistic inference. The fact that robot localization and mapping are characterized by uncertainty and sensor noise is the reason for the popularity of probabilistic techniques. Probabilistic algorithms approach the problem by explicitly modelling different sources of noise and their effects on the measurements.

2.1.1 Perception

The most common methods of position estimation are absolute and relative positioning, usually working together. In the case of relative positioning [Borenstein and Koren, 1987], usually is based on *dead-reckoning*, estimating the current position by determining the offset from the initial position, computed by counting the revolutions of the robot wheels [Krantz, 1996]. This method is inexpensive, easy and real-time possible. On the other hand, the *dead-reckoning* has the disadvantage of error accumulation over the time [Zhou and Chirikjian, 2003]. That errors will become so large that the robot's internal position estimate may be unacceptably wrong after a 10m of travel [Gourley and Trivedi, 1994]. There are two kinds of errors:

- *Non-systematic:* Consequence of the slipping and skidding produced in the turns of the robot or by the irregular ground, like cracks or bumps.

- *Systematic errors:* These errors are particularly damaging, because they are accumulated constantly and they are due to the irregularity of the robot mechanic, like the unequal wheel diameters or the uncertainty about the distance between the contact points of the wheels and the floor because those wheels contact the floor not in one point [Borenstein, 1994].

LIDAR scanners and range sensors are widely used in mobile robotics applications such as obstacle detection and avoidance [Chang et al., 2008] [Zeng and Weng, 2007] [Choi et al., 2007], object tracking [Kogut et al., 2007], map building [Ueda et al., 2006], feature extraction [Nguyen et al., 2005] [Borges and Aldon, 2004] [Premebida and Nunes, 2005] and self-localization [Sohn and Kim, 2005] [Lingemann et al., 2005].

A 2D LIDAR scanner provides distances and angles to the surrounding objects by scanning the environment in a plane, usually parallel to the ground. This technique is not enough to detect obstacles like stairs, a land irregularity or floor level variations. These kinds of problems are overcome with the 3D LIDAR scanners. With these sensors, all features are directly extracted in 3 dimensions. In the other hand, their price is much more expensive than the first ones. Another solution for these problems is a combination of a 2D LIDAR and a *pan-tilt* unit but, on the contrary, this system requires to spend more time to obtain the measures, due to the *pan-tilt* unit has to be moved. [Iocchi and Pellegrini, 2007].

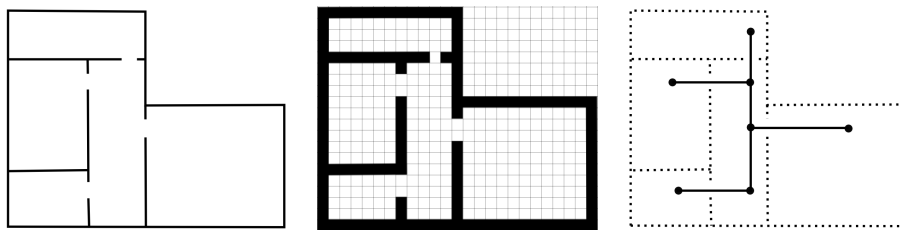
2.1.2 Mapping

In order to perform the local navigation or global path planning a representation of the environment (*mapping*) is needed. That map can be *local* (represents the surroundings of the robot, most suitable for local navigation) or *global* (the environment in where the robot moves). Those maps can be built based on another map, based on measures from the on-board sensors, based on expert knowledge, etc.

Mapping is performed with the robot's on-board sensors. Commonly used sensors are: sonar sensors, LIDAR sensors, cameras, depth cameras, infrared sensors, etc. For local mapping purposes,

the principal information needed is to know where there free space is and where the obstacles surrounding the robot are. For indoor mobile robots, it is assumed that any detection in the same height of the robot is an obstacle. For that reason the most used sensors are range only sensors (sonar, infrared, LIDAR), situated parallel to the floor. Nevertheless, for outdoor mobile robots and for some complex robots or environments, 3D information is needed. That information can be obtained from 3D-LIDAR [Montemerlo et al., 2009], moving 2D-LIDAR [Nuchter et al., 2003], stereo cameras [Saez and Escolano, 2004], etc.

Usually, indoor mobile robots moves in a 2D world. These 2D representations of the environment can be classified into *topological* and *metric* representations [Thrun, 1998]. *Topological maps* represents characteristic points in the environment and the relations between them. These maps tend to be simple, close to the human interpretation of the environment and they are very useful for top level task or path planning but are not useful for local navigation. In the other hand, metric maps represent a continuous space discretized in a way that can be used by a robot. Figure 2.1 shows the same map (Figure 2.1(a)) represented in both ways: *metric map* (Figure 2.1(b)) divides the continuous map in cells of the same size and *topological map* (Figure 2.1(c)) represents each room and the transition between them.



(a) Original Map (b) Metric Map (c) Topological Map

Figure 2.1: Mapping: Metric and Topological.

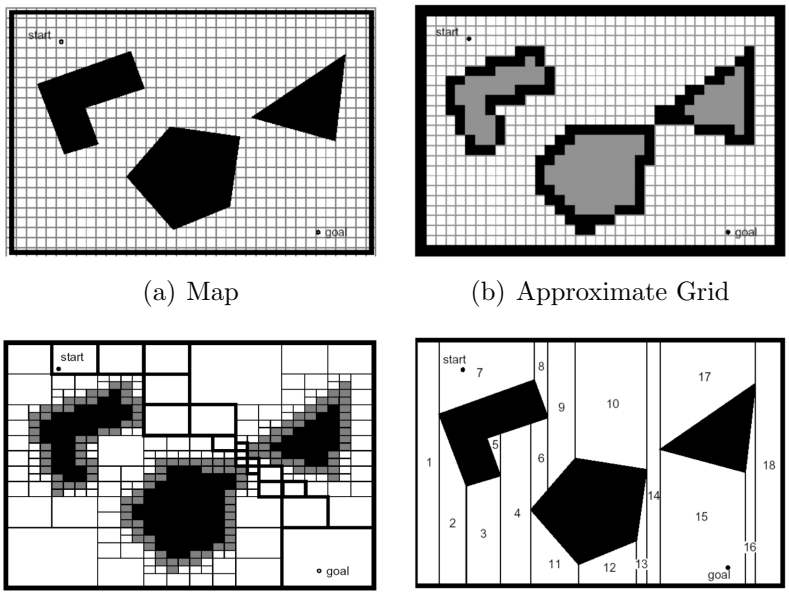
Building a metric map from continuous information needs a discretization. One effective way to perform that discretization in 2D environments is the cell decomposition method. This method divides

the environment in cells that are connect with the adjacent ones and allows to represent if each cell is *occupied* (not traversable by the robot) or *free* (traversable by the robot). Even a probabilistic occupancy level can be represented in a way that the loss of information caused by the discretization, or the inexactitudes of the sensors measures can be filtered [Coué et al., 2006]. [Latombe, 1991] exposes some methods for cells discretization:

- **Approximate cell decomposition:** the environment is divided in cells of the same size and shape (see in Figure 2.2(b)).
- **Adaptive cell decomposition:** the environment is divided in cells of different size. One particular case of adaptive decomposition is the *quadtree* [Samet, 1988]. The method divides an environment into *quadtrees* begins with one cell that represents the whole map. If that cell is partially occupied, it is divided into four ones. That division is repeated on each cell until the resolution limit is reached or there are not partially occupied cells in the environment. That representation allows to have the same information as approximate cell decomposition using less memory (see in Figure 2.2(c)).
- **Exact cell decomposition:** the cells does not have a predefined size or shape and are determined based on the environment. The union between cells represent the free space in the map (see in Figure 2.2(d)).

Figure 2.2 shows the three previous explained methods for cell discretization applied to the same map (Figure 2.2(a)).

The grid cell representation can be extended to full 3D maps, where the representation is based on *VOLumetric piXEL (VOXEL)* [Foley et al., 1990] instead of grid cells. Also each grid can contain more information apart from the occupancy value, allowing fusing *topological* and *metric* maps by mean of adding information, for example, about features on the environment that lies inside each cell.



(a) Map (b) Approximate Grid

(c) Adaptive Grid (d) Exact Grid

Figure 2.2: Cells Discretization Techniques.

2.2 Detection and Tracking Moving Obstacles

In order to achieve a safe navigation in a partial or completely unknown environment, also including possible dynamic objects, the mapping methods shown in Section 2.1.2 are not enough. They only take into account the occupancy, not dealing with dynamic obstacles. For that reason, the robot has to manage a representation of the world that be achievable, that include information to predict the future of the environment, continuously update and with a reliability level. There are different approaches that can deal with dynamic obstacles, called DATMO methods:

- The first kind of methods are based of the assumption acknowledge of all objects around the robot, learning their tracks and velocities on an *off-line* process. These methods are only applicable in structural and constant environments, such as the

industrial ones. For that reason, they are out of the aim of this thesis.

- The second kind of approaches try to estimate the position and velocity of the objects in an *on-line* process. Usually in two steps: *data association* phase and a *multi target tracking* algorithm. These methods do not take into account the unknown space and can suffer problems in a crowded environment. These methods are *Objects Based DATMO* (Section 2.2.1) and can be divided in *Model-free* and *Model-based* approaches, described in Sections 2.2.1.1 and 2.2.1.2 respectively.

- The third alternative to these classic *object framework* are the methods that maintain a probabilistic knowledge of the environment in a *dynamic occupancy grid* domain, including the actual occupation and the estimation of the cell's velocities. These methods are the *Grid Based DATMO* approaches, described in Section 2.2.2.

2.2.1 Objects based DATMO approaches

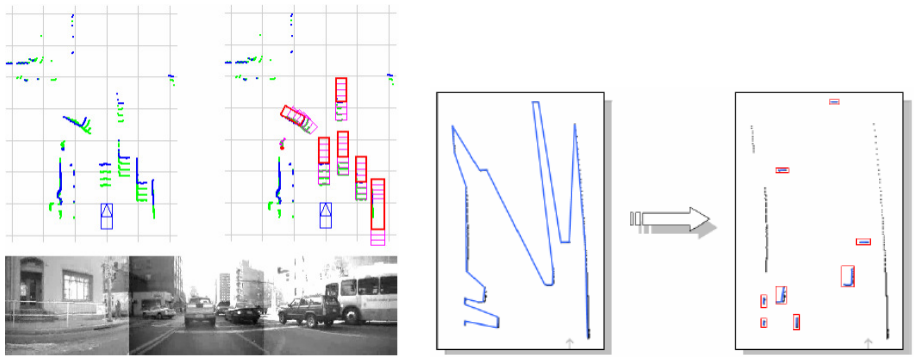
The DATMO problem has been under research in the last decades. DATMO is the process of observe the states of objects surrounding in a dynamic environment with the exteroceptive sensors, assuming an accurate pose estimate available, to obtain the position and the trajectories of the dynamic objects. Usually, the objects based approaches try to track different objects independently, keeping them in a list. This is a *multi-objects* or *multi-target* tracking problem, whose main difficulty is the *data association* problem. This problem rely on the knowledge of:

- Whether new measurement from sensors correspond to an existing object

- When an object should be: *created* as new object, *maintained* or *deleted*.

Many works focus on the *multi-objects* tracking problem assuming that the measurements are uniquely from moving objects. Although,

static objects or spurious measurements exists in most of the real applications. The sensors widely used in this area are the range sensors (sonar or LIDAR), or cameras. Although, the LIDAR is usually the main sensor due to its high accuracy and resolution to detect objects. Some illustrative examples are shown in Figure 2.3.



(a) [Vu and Aycard, 2009] (b) [Nashashibi and Bargeton, 2008]

Figure 2.3: DATMO examples: LIDAR based object detection.

Previously to the *data association* stage, is necessary segment the data. The simplest way is the *clustering*, based on the range discontinuities of the data. Others approaches trying to search some features in the data, such a *lines* or *"L" shapes*. The clusters has to be assigned to the target, taking into account that multiple clusters can come from the same target.

The *data association* stage can be computed with simple methods, such as *Nearest Neighbor Rule (NNR)*, assigning each data cluster to the closest target, widely used at high enough update rates to assume the assignment unambiguous. One step forward are some variations of these methods, as *Global Nearest Neighbor (GNN)*, that ensures that each cluster is associated to one target. This method is appropriate in recognition based approach, where each cluster represents the whole object. In the case that the ambiguity should be high, there are other commonly used methods:

- *Multiple hypothesis tracking (MHT)* by [Reid, 1979]: this algorithm can handle measurements that came from a varying

number of targets, maintaining multiple association hypotheses between clusters of measurements and targets. The number of possible hypotheses increases each time step due to new possible hypotheses are created from initializing new target from each new measurement in addition to the previously existing hypotheses. Each hypothesis obtain a score summing the track existence scores of all target within it. For that reason the combinatorial increase, being necessary techniques to reduce them, such as tracks clustering or pruning.

- *Joint Probabilistic Data Association Filter (JPDAF)* by [Fortmann et al., 1983]: This algorithm only maintain one association hypothesis, similar to GNN, avoiding the problem of the increasing combinatorial. The association is not hard as in GNN, is a soft assignment based on the probability of each measurement being associated to each target. For target k the measurement update is taking into account all possible assignments the probability of data clusters (Equation 2.1). The probability that the measurement m are from target k be denoted by β_{mk}:

$$p(Z_t|S_t^k) = \eta \sum_{m=1}^{M_t} \beta_{mk} p(Z_t^m|S_t^k) \tag{2.1}$$

where K is the number of targets, M_t the data clusters $(Z_t^1, ..., Z_t^{M_t})$ and η is a normalization constant. There are some variants and extensions of JPDAF. ([Cox, 1993]) The variants that are used in DATMO approaches rely on parametric belief representations.

The *target tracking* stage in DATMO is performance using primarily *Kalman Filters (KFs)*. As the belief is usually non-Gaussian, is necessary use of its variants, the *Extended Kalman Filter (EKF)* and *Unscented Kalman Filter (UKF)*. These parametric filters are computational cost efficient, but it can't represent complex beliefs when arise due to the data ambiguities. In contrast, there are non-parametric filters, such as *Particle Filter (PF)* or *Histogram Filter*

(HF), that represent the belief by a set of particles, that can represent an arbitrary belief. Nevertheless, the computational cost is exponential in the dimensionality of the state and usually the base of the exponent is large.

Other approaches use *multiple model-based* moving object tracking. Due to the lack of a priori information, the on-line mode learning is a hard task. For that reason, the motion mode of moving objects usually is approximately composed of several motion models such as, constant velocity model, constant acceleration model and turning model. In this way, the mode learning problem is simplified to a multi-model selection problem. Even so, due to the time variant of the model of the moving obstacles, it is necessary a suboptimal approach with merge or reduce the number of the mode history hypotheses to avoid the exponentially increasing of them. *Interactive Multiple Model (IMM)* filter [Blom and Bar-Shalom, 1988] is widely uses in that case. This approach compute the state estimate at time k under each possible current model using a suitable mixing of the previous model-conditioned estimate as the initial condition at each filter, as shown the IMM diagram in Figure 2.4.

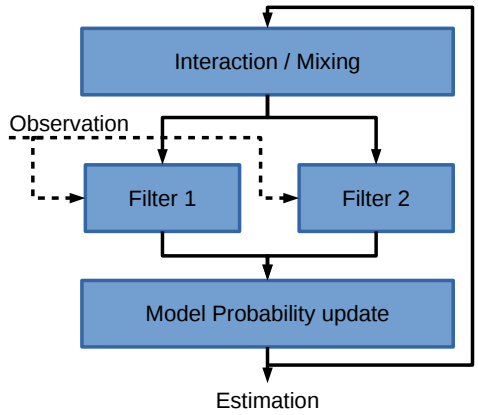


Figure 2.4: IMM filter diagram.

On the other hand, the DATMO problem can be addressed as a batch problem over a fixed time window. It could be solved using different approaches based on *Markov Chain Monte Carlo (MCMC)* [Andrieu et al., 2003]. These methods start with a desired probability

distribution, then produce samples of them by constructing a Markov chain with an equilibrium distribution equal to de desired distribution. The resulting state of the chain can be used as a sample from the desired distribution once the methods walk the chain a enough steps. To construct the Markov chain is usually easy, while obtain the number of steps needed can be difficult.

2.2.1.1 DATMO Model-based approaches

These methods know the class of the objects to be detected a priori. With this information, each object is detected based on a parametric model (of its shape) to track it separately. These methods have the restriction to detect only the specific objects described in the parametric model.

Some of these methods are focus on people detection, for example, in work [Arras et al., 2008], authors train a boosting classifier to detect legs separately with KF and constant velocity models, then, they group the legs of each person to track them with a MHT. Other authors, [Schulz et al., 2001] only detect the legs of people on the 2D range scanner and, to robust the tracking, apply the JPDAF. [Topp and Christensen, 2005] extends the previous work detecting the people whose legs are not directly visible.

Others works, instead of detecting people, focus on detect and tracking vehicles, such as [Vu and Aycard, 2009] that represents the vehicles with a box model. The data association and tracking of the dynamic obstacles were implemented with GNN and KF. Also, the authors, present an algorithm that optimize the box models over the best trajectories of the vehicles in a sliding window of laser scans to solve detection and tracking simultaneously using a *Data-driven Markov Chain Monte Carlo (DDMCMC)* algorithm. The authors improve the odometry with a fast incremental scan matching. The system was tested with Navlab dataset [Wang et al., 2004].

Another example of model-based vehicle detection and tracking is the work by [Zhao and Thorpe, 1998], authors proposed an IMM filter with three different motion models to determinate all of possible motions of the vehicles that are tracked. On the other hand, the work by [Granström, 2012] does not focus on people or vehicles, they use

a *Probability Hypothesis Density (PHD)* filter to detect and track rectangular and elliptical targets. [Chavez-Garcia and Aycard, 2015] classifies the objects in four class of interest: pedestrian, bike, car or truck, using Adaboost by [Friedman et al., 2000] (a boosting-based learning algorithm). In addition, they use an IMM with constant velocity, acceleration and turning models to track the objects and a pruned MHT to perform the data association.

2.2.1.2 DATMO Model-free approaches

These methods do not impose restrictions in obstacle's type or shape and the semantic information about the object is not needed. On the other hand, these methods only detect the current moving obstacles, not the potentially dynamic object that will move in the future.

Some of these methods, included in the systems that participate in the *DARPA Urban Challenge* [Leonard et al., 2008] [Mertz et al., 2013], works as follow: first, they segment different measurements from several lasers with a basic region-growing algorithm in segments to extract geometric features of the objects (corners and line ends of the *"L" shapes*); second, with these features, they compose a list of possible objects and, finally, they extract the objects that have a significant speed, tracking them with a KF with constants turn rate and acceleration and a fixed process noise. The output velocity has a delay that varies between 1.1 and 3.5s.

Others methods are based on estimating a static map of the environment at the same time that detect the moving obstacles using the knowledge of occupancy probabilities to compute the likely moving objects. In [Wang et al., 2003], the authors represent data in grid-maps computing an *Iterative Closest Point (ICP)* algorithm, then a MHT algorithm is used for data association and an IMM algorithm for tracking. The approach was tested in crowded urban environments at high speeds. Though, the method proposed by the authors need to include a prior digital map, not available everywhere. This work was extended by [Montesano et al., 2005], estimating the robot pose at the same time that difference the dynamics and statics objects with an extended *Iterative Dual Correspondence (IDC)* algorithm and integrating the identification of dynamics objects within the estimation

process. Also, a NNR is used for data association. Other similar work is the Toyota's tracking system [Miyasaka et al., 2009] that combine SLAM with the tracking of moving obstacles.

In [Vu et al., 2011], the authors improve the odometry with a fast incremental scan matching and the occupancy map that use is based on the grid framework by [Elfes, 1989]. The dynamic obstacles are detected taking into account their inconsistencies with the grid map. Finally, a MHT with an adaptative IMM are used for the data association and tracking stages.

Using map differences between the occupancy map of the static obstacles, they can detect moving obstacles and then identify the objects with an *Expectation-Maximization (EM)* algorithm [Biswas et al., 2002].

In the work by [Yang and Wang, 2011] authors use a variant of *RANdom SAmple Consensus (RANSAC)*, to estimate the robot pose and the moving obstacles and a decision tree based on spatiotemporal consistency tests to manage the track merging and splits. [Hähnel et al., 2003b] propose a EM algorithm to determine whether a laser point is static or dynamic solving a set of hidden indicator variables of each laser point. The same authors [Hähnel et al., 2003a], presented a probabilistic technique to create maps that contain people. To track people, the algorithm implemented is a *Sample-based Joint Probabilistic Data Association Filters (SJPDAF)* using the laser measurements. To identify the moving people and the static obstacles, the authors use an occupancy probability maps.

Other approach is to focus on the detection part, solving the data association and moving object detection problems using a joint *Conditional Random Field (CRF)* framework, as in the works [Tipaldi and Ramos, 2009], [van de Ven et al., 2010]. In [Wang et al., 2015], the authors deal with the data association in two levels: a coarse level predicting the boundary points of the objects computing an *Euclidean Minimum Spanning Tree (EMST)-Efficient Graph-Based Image Segmentation (EGBIS)* clustering algorithm, then compute an ICP, and present a variant of *Joint Compatibility Branch and Bound (JCBB)* in the fine level.

2.2.2 Grid Based DATMO

These methods represents the environment as an occupancy grid, where each cell is tracked at a sub-object level, instead of segment the environment into objects to track them. These approaches avoid the *object concept*, also the problem of *multi-object* detection and tracking, sometimes very difficult to solve. Other advantage of these methods is that the sensor's data fusion from different sensors can be compute at the raw data level using occupancy grid maps, where the data association is not needed, unlike in the previous DATMO approaches.

One the most popular approaches an the base of several works is the *Bayesian Occupancy Filter (BOF)* by [Coué et al., 2006]. The BOF evaluate the environment occupancy regardless the kind of the object. The occupancy of the cells are computed as an probabilistic formulation initially proposed by [Elfes, 1989], dealing with uncertainty due to the noise or the inaccurate sensors through the *Bayes theorem*.

On the other hand, there are others approaches that are based on the *Evidence Theory*, also called *Dempster-Shafer theory* [Dempster, 1967] taking into account the inconsistencies between consecutive grids as evidences of conflict. This method has the advantage of modelling a cell in three states: *free*, *occupied* or *not observed yet*. Although, a few works [Kurdej et al., 2012] [Moras et al., 2014] are based on these method, due to tuning the parameters needed is usually a challenging task. For that reason this thesis is focus on the methods based on BOF.

The BOF is described as a representation of the space in a cells grid. Each cell describes a portion of the environment and includes different kinds of information about this part of the space, such as, occupancy, velocities, riskiness, etc. This representation has several advantages:

- To be a generalist representation allows to fuse the information from different sensors and using in indoors and outdoors.

- The use of *prediction* and *estimation* stages taking into account

the previous occupancy history, helps to overcome some occlusions.

- The probabilistic framework based on *Bayes theorem* handles the uncertainty and the noise in the sensor's measures, increasing the robustness.
- As the cells are independent, this allows to parallelize processes, increasing the performance.

The work by [Saval-Calvo et al., 2017] defined the five layers of the BOF corresponding to the five main parts, from the bottom layer closer to the raw data sensor to the top one, where the high-level algorithms are implemented, shown in Table 2.1

Table 2.1: Levels of BOF.

Level	**Task**
5th	High-level Applications
4th	Clustering
3rd	BOF core
2nd	Grid estimation
1st	Pre-process

The definition of BOF following the description in [Fulgenzi, 2009] is as follow: the 2D Euclidean space is divided in a finite number of cells, each represents a position in the plane. The state of the system $O(t)$ at time t is the list of the states of all the cells of the grid: *Occ*, when the cell is *occupied* or *Emp* if the correspondent space is *free*. Given a probabilistic sensor model $P(z(t)|o(t))$ where $z(t)$ is the current observation, the grid is updated following the Bayes' rule. Under the hypothesis that each cell of the grid is independent from its neighbour cell, each cell state estimation is updated independently [Elfes, 1989]. To handle dynamic obstacles, each cell of the BOF maintains not only an estimation of its occupation probability,but also a discretized representation of the *probabilistic distribution function (PDF)* over velocities. A minimum and maximum velocity value is considered for eventual objects in the space, and the

PDF is approximated by a finite histogram over regularly distributed velocity values v_n with $n \in 1...N$. The discretization step is chosen according to spatial and time discretization: given q the size of a cell and τ the time step, only integer velocities in terms of $\frac{q}{\tau}$ are taken into consideration. This choice is necessary in order to perform fast and rigorous prediction and updating steps.

The variables used to formalize the BOF probability estimation are as follows:

- C is an index that identifies each 2D cell of the grid.
- A is an index that identifies each possible antecedent of the cell c over all the cells in the 2D grid.
- $Z_t \in Z$ where Z_t is the random variable of the sensor measurement relative to the cell c.
- $v \in V = v_1, ..., v_n$ where v is the random variable of the velocities for the cell c and its possible values are discretized into n cases.
- $O, O^{-1} \in O \equiv occ, emp$ where O represents the random variable of the state of c being either *occupied* or *empty*. O^{-1} represents the random variable of the state of an antecedent cell of c through the possible motion through c. For a given velocity $v_k = (v_x, v_y)$ and a given time step δt, it is possible to define an antecedent for $c = (x, y)$ as $c^{-k} = (x - v_x \delta t, \quad y - v_y \delta t)$.

The decomposition of the joint distribution of the relevant variables according to Bayes' rule and dependency assumptions as state [Tay et al., 2008b] is given by Equation 2.2:

$$\begin{aligned} &P(C, A, Z, O, O^{-1}, V) = \\ &\quad P(A)P(V|A)P(C|V, A)P(O^{-1}|A)P(O|O^1)P(Z|O, V, C) \end{aligned} \tag{2.2}$$

The parametric form and semantics of each component of the joint decomposition are as follow:

- $P(A)$ is the distribution over all the possible antecedents of the cell c. It is chosen to be uniform because the cell is considered reachable from all the antecedents with equal probability. Consequently, given k antecedents, each one has a probability $P(A) = \frac{1}{k}$.

- $P(V|A)$ is the distribution over all the $\{v_1, ..., v_n\}$ possible velocities of a certain antecedent of the cell c; its parametric form is a histogram.

- $P(C|V, A)$ is a distribution that explains whether c is reachable from $[A = a]$ with the velocity $[V = v \in \{v_1, ..., v_n\}]$. In discrete spaces, this distribution is considered a Dirac with value equal to 1 if and only if $c_x = a_x + v_x \delta t$ and $c_y = a_y + v_y \delta t$, which follows a dynamic model of constant velocity. This Dirac distribution is used in the BOF by [Coué et al., 2006], nevertheless, other general distribution approaches could be used.

- $P(O^{-1}|A)$ is the distribution over the occupancy of the antecedents. It gives the probability of the possible previous step of the current cell. Given the antecedents, this probability explains probability that the previous occupancy is reliable with the current antecedents.

- $P(O|O^{-1})$ is the conditional distribution over the occupancy of the current cell, which depends on the occupancy state of the previous cell. It is defined as a transition matrix:

  $$T = \begin{bmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{bmatrix}$$

  which allows the system to use the null acceleration hypothesis as an approximation; in this matrix, ε is a parameter representing the probability that the object in c does not follow the null acceleration model.

- $P(Z|O, V, C)$ is the conditional distribution over the sensor measurement values. It depends on the state of the cell, the velocity of the cell and obviously the position of the cell.

The Joint probability $P(O, V|Z, C)$ can be decompose into observation and prediction probabilities, estimating in this way the occupancy and velocity probabilities of each cell, following the *prediction/estimation* diagram shown in Figure 2.5:

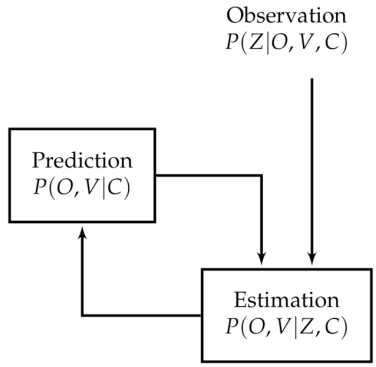


Figure 2.5: BOF stages: prediction and estimation [Tay et al., 2008a].

The authors discretized the possible velocities. In order to implement the probability distribution, is possible to do it in form of histograms, as shown in Figure 2.6:

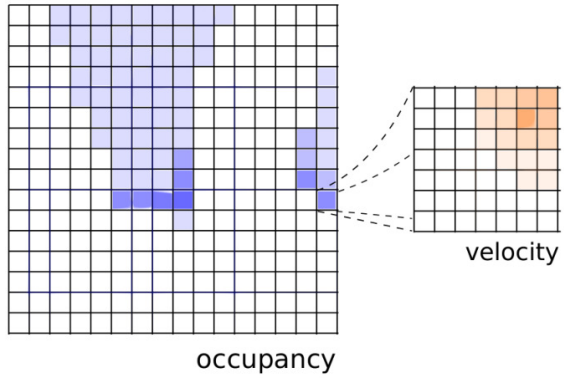


Figure 2.6: BOF Grid representation [Nègre et al., 2014].

2.2.2.1 Variants and improvements of the BOF approach

Taking into account the BOF layers described by [Saval-Calvo et al., 2017], several authors had propose improvements or variants in the different layers. The **first layer** is refer to the *processing the data* from the sensors before compute the grid. Usually, the sensors used in occupancy grids are range (LIDAR or sonars) sensors, sometimes also cameras. As these sensors do not provide the measures in a grid space, it is necessary processing this data.

Linear Opinion Pool (LOP) techniques are commonly used to treat the data. In this way, it is possible to avoid that the non-reliable data affect to the global result, having a sensor model and a weighting term. The authors in [Adarve et al., 2012] use LOP to multi-sensor fusion, such as LIDAR and stereo cameras. Other authors, [Baig et al., 2014] fusing multiple layers from laser scanner using LOP.

In the **second layer** (*Grid-estimation*): The authors in [Chen et al., 2006] use the difference on the occupancy between time steps to compute the velocity, keeping the computation of the object detection and tracking in the occupancy grid. In [Tay et al., 2008a], the authors introduced velocity information reducing noise in estimations, improving the predictability and reducing the computational cost. While [Coué et al., 2006] represents the velocity in two dimensions (V_x, V_y), corresponding to the two possibles directions, increasing the dimensions of BOF to *4D BOF*, [Chen et al., 2006] introduce a histogram of velocities in the cell (*2D Histogram BOF*). The *4D BOF* have the advantage that represent objects that overlap others and the *2D Histogram BOF* has a less computational cost.

Other variant introduced by [Yguel et al., 2006] is taking into account a non-constant velocity. Especially important in urban or indoor scenarios where the objects do not have constant velocities. To afford this, the cells are updated at different frequencies, based on their velocities. Additionally, the authors handled the aliasing problem, due to depending on the objects orientation, the number of cells occupied by them are different.

The authors in [Baig et al., 2014] added a motion detection stage to recognize the moving obstacles, based on keeping update a Free and

Occupied count Arrays, that means the number of times that each cell has been free or occupied. With that counts and an heuristic, the system is able to identify the moving part in the environment.

Variants on the core of the BOF method (**third layer**) has been introduced by several authors, reducing the computational cost using a prior knowledge of the environment or representing the dynamism of the cells using particle filters. The authors in [Gindele et al., 2009] proposed a *Bayesian Occupancy grid Filter for dynamic environments Using prior Map knowledge (BOFUM)* that predicts the more precise cells movements taking into account a prior knowledge about the cell environment. The object motion is usually dependent on the its localization, restricting his movements based on behaviours patterns. For example, in urban scenarios, it is more probably that the cars follow the lanes than cross them perpendicularly, neither they drive on the sidewalk. On the other hand, the movements of pedestrians are not so restricted. The authors apply a reachability matrix that contains the changing probability of the cells based on the area. They defined three possibles areas (lane, sidewalk and unknown) with a cost function to compute the probability of one cell being the antecedent of another based on three assumptions:

- The area changing is unlikely.
- If an object is moving off the lane and the antecedent cell are in others areas, is highly probably to be a pedestrian.
- The vehicles usually moving on the lane and the moving out the lane or lane change probabilities are low.

Others authors use BOFUM, in [Brechtel et al., 2010] the authors accelerate the convergence of the BOF using a prior map knowledge. Also, in this work including uncertainty in the motion model and a importance sampling (*IS*) recursively to approximate BOFUM calculations, it is similar to the PF in a discrete cell framework. With this approach, the authors improve the time performance of the system at least 40 times. Other improvement added is the property of the cells that can move in object groups, defined as *Bayesian Occupancy Filter Using Groups (BOFUG)* allows to infer object classes only from

the occupancy measurement. On the other hand, the disadvantages of the systems based on BOFUM is that the required maps are not always available and the aliasing problem is not resolved.

Another variant is to use PFs to obtain the occupancy and velocity of the cells. In [Danescu et al., 2011] the authors define an algorithm to compute the environment dynamics with a variable number of particles per cell depending on its occupancy. The sensors used is stereo vision and the approach is based on three stages: first, the prediction, reallocating the particles taking into account the speed and ego-motion of the vehicle; second, processing the measurement incorporating the measured data to weight the particles and re-sample them, and finally, estimation of the occupancy and velocity. The moving particles represent the environment without discretization in its positions, correcting in this way some aliasing problems. They compared the results with the *Lucas-Kanade* Optical Flow. This method has been later named as *Sequential Monte Carlo Bayesian Occupancy Filter (SMC-BOF)*. Based on this approach, authors in [Nuss et al., 2013] proposed a data fusion between laser and radar.

Based on the same idea of using PF and to increase the performance, the authors in [Nègre et al., 2014] introduced the *Hybrid Sampling Bayesian Occupancy Filter (HSBOF)* to avoid the high number of particles unnecessary used in the case of static cells. The authors do not use particles in the static cells neither in the free areas, mixing static occupancy based on the original BOF and dynamic occupancy modelled with a moving particles with the same PF principle. The particles in the dynamics cells are composed of a velocity vector and an exact position, to avoid aliasing. Also, each particle has a weight associated to represent the distribution of the speed in the cell.

As it is shown in Figure 2.7, each cell can be decomposed in three sections: *static occupied*, *dynamic* and *free*.

This approach has velocity detection problems in the case of linear geometry objects, such as the highway median strip, that are detected as dynamic.

Although in the BOF framework the *object concept* do not exist, in some applications the representation of object or track is necessary. In this case, some authors propose add a level of *clustering* above the

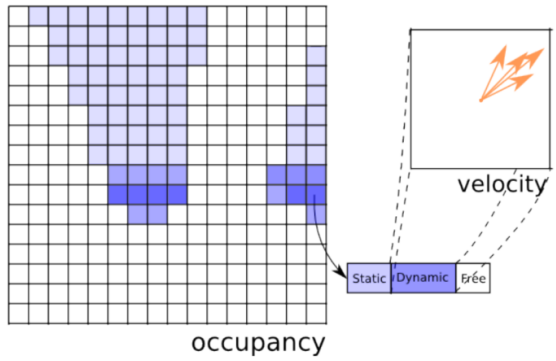


Figure 2.7: HSBOF Grid representation [Nègre et al., 2014].

BOF in order to extract the objects. These variants are located in the **fourth layer** (*Clustering*) of Table 2.1.

The work by [Mekhnacha et al., 2008] proposes clustering the cells of the BOF in objects. The input of the tracker is the occupancy and velocity grids of the BOF. The first approach that they test is a layer with a JPDAF above the BOF layer, but the hypothesis number that the JPDAF generates increases rapidly. In order to decrease the computational cost, they proposed the *Fast Clustering-Tracking Algorithm (FCTA)*. The tracker creates a grid with the same size of the BOF grid that contain the corresponding identifier between each cell and the cluster to which belongs. The algorithm is divided in three modules; *clustering*, a eight-neighbour approach with a occupancy and velocity thresholds for classify the cells in groups; *re-clustering and merging*, that handle the ambiguity in association and a *track module* based on a KF. They test the system with two pedestrian and a car using a SICK LIDAR, and can manage the occlusions between the pedestrians.

Finally, in the **fifth layer** (*High-level Applications*) of Table 2.1 are located the works that use the data output of the BOF as input of their system. In this way, the authors take advantages of the robustness on the occupancy computation and the capacity of handled the data fusion from different sensors. Some of these works are [Vu et al., 2011], explained in 2.2.1.2 and [Vu and Aycard, 2009] explained in 2.2.1.1. Based on HSBOF, the work by [Yuan et al.,

2015] combines the information of the LIDAR used for occupancy state updating only and the RADAR measurements that are affordable for likelihood evaluation. This approach leads more accurate dynamic estimates and therefore guide the particles flow in a correct and accurate manner. The authors detect the dynamic obstacles in a driving scenario and their movement, but they do not show the numerical velocity of them. The pedestrians are not well detected due to the move-stop-move behaviour.

2.3 Discussion

In the previous sections a survey of the DATMO approaches has been introduced. This section presents a comparison and a discussion of the most important works on this area.

DATMO is a rapidly developing field, especially in autonomous vehicles. It is still a very open problem, due to a DATMO fully autonomous system that can rival the human capabilities in all the cases is not yet available. However, it's not easy to compare the methods in a quantitative way due to a lack of numeric results or difficulty of a direct comparison using a common dataset. For that reason and in order to show a general view of the performance of the different methods, the comparison has been divided in two tables:

- Table 2.2 shows a comparison of some of the *Objects Based* algorithms referred in section 2.2.1. In this case, the comparison is based on some parameters, such as the *sensors* used, the *data association* and *tracking* methods, the *object model* used, and the *different dynamics objects* that can handled.

- Table 2.3 shows a comparison of some of the *Grid Based* methods referred in section 2.2.2. Is not easy to compare the different variants and improvements based on BOF approach due to their different focuses and the lack of similar experiment and numeric results. For that reason, the comparison is based on performance parameters, such as *computational cost* (memory usage and speed), *possibility of parallelization* in dedicated

hardware, *estimation of the velocity* of the cell, the *aliasing robustness* due to the grid discretization, the simplicity of the *free space representation*, the ratio between the *occupancy probability accuracy* and the unexpected noise that appears in the free cells, the *prior knowledge needed* of the map and the capability to manage dynamic obstacles.

Table 2.2: Object Based DATMO Methods comparison

Object Based DATMO Methods								
Author	**DATMO Method**		**Dynamic objects**		**Sensors**	**Environment**	**2D / 3D**	**DATMO Contribution**
	Data Association	**Tracking**	**Model**	**Kind**				
[Wang et al., 2003]	MHT	IMM	Free	People, cars, bikes, buses	Laser, odometry	Outdoor	2D	Pioneer of DATMO
[Hähnel et al., 2003a]	SJPDAF		Free	People	2D SICK Laser	Indoor	2D	Reduce spurious objects with a more robust algorithm
					Tilting 3D Laser, 2D Laser	Outdoor	3D	
[Montesano et al., 2005]	NNR	EKF	Free	People, doors	SICK 2D Laser, odometry	Indoor	2D	Differentiated static & dynamic objects in estimation problem
[Vu and Aycard, 2009]	DDMCMC		Based	People, cars, bikes, buses	2D Laser	Outdoor	2D	DDMCMC
	GNN	KF	Based	Objects that can cause collisions in traffic	2D laser, odometry, 2 short range radars			–
[Vu et al., 2011]	MHT	IMM	Free	People, cars	2D laser, odometry, 2 short range radars	Outdoor	2D	–
[Wang et al., 2015]	EMST-EGBIS, ICP + Variant of JCBB	EKF	Free	People, bikes, cars, buses, trucks	SICK LDMRS, odometry	Outdoor	2D	Variant of JCBB and EMST-EGBIS clustering algortihm
[Chavez-Garcia and Aycard, 2015]	pruned MHT	IMM	Based	People, bikes, cars, trucks	LIDAR, Sonar, cameras	Outdoor		Enhanced representation (kinetic+appearance) at the detection level

Table 2.3: Grid Based DATMO Methods comparison

Grid Based DATMO Methods									
Method - Author	**Computational Cost**	**Parallelization Possibility**	**Velocity estimation**	**Aliasing robustness**	**Representation free space**	**Accuracy / noise ratio**	**Prior knowledge needed**	**Handling Dynamic obstacles**	**DATMO Contribution**
4D BOF [Coué et al., 2006]	Very High	High	Medium	Low	Simple	Low	No	Medium	Pioneer of Grid Based DATMO
BOFUM [Gindele et al., 2009]	High	Medium	High	Medium	Simple	Medium	Yes	High	Introduced Map restrictions to reduce computational cost
BOFUG [Brechtel et al., 2010]	Medium	Medium	High	Medium	Simple	Medium	Yes	High	Added the cells property that can move in object groups
SMC-BOF [Danescu et al., 2011]	Medium	Low	High	High	Complex	High	No	High	Introduced the PF to the cells
HSBOF [Nègre et al., 2014]	Low	Medium	High	High	Simple	High	No	High	Mixed the static part of BOF with the dynamic part based on PF

Several conclusions can be drawn from the analysis carried out in previous sections and the information in Tables 2.2 and 2.3:

- *Objects Based Methods* do not take into account the unknown space. Also can suffer problems in a crowded environment, due to some of the data association methods, such as MHT and JPDAF suffer a rapidly combinatorial increase (*combinatorial explosion of hypothesis*). Other disadvantage is that these methods do not take into account the participants of unusual appearance, such as unusual vehicles (construction equipment or cars with caravans...), people wearing fully loose clothes or costumes, people pushing objects (baby carriage, shopping cart...) or small participants (kids, pets...). One advantage of these approaches is that usually provides more accurate velocity estimation of the objects (previous known objects or trained models).

- Since DATMO systems are used to obstacle avoidance or as *Advanced Driver Assistance Systems (ADAS)*, both of them require a very quick response. Taking into account the high dimension of the data provided from the sensors (LIDAR, stereo cameras...) it is necessary to reduce this dimension while system performance is not endangered. In this way, the *Grid Based Methods* address this issue.

- *Grid Based methods* allow to estimate each velocity cell, instead of having to model the dynamic environment. However, the cell size has to be small enough to be able to detect slow velocities, due to the velocities are computed when an object change from one cell to neighbour one, increasing the computational cost. Some approaches, such as, BOFUM and BOFUG accelerate the convergence of the BOF, but the required maps are not always available. Taking into account the sensor observation history, the system is more robust in changing environments. Then, temporary objects, occlusions and detection problems can be handled. In the case that represent the environment of a moving subject (*Unmanned Ground Vehicle (UGV)*, car...) has to

deal with the relative velocity between the objects and the subject. This implies to maintain a distribution in the velocities at least twice wider than the maximal motion speed. Thus, the size of the data structure can be huge, limiting the applications of these approaches. These methods are especially useful to fuse measurements from different sensors. Also are useful for identifying wide variety of objects with different shapes or appearance without previous knowledge of them.

- Some approaches of both cases (*Objects and Grid base*d), only take into account de edge of the monitored area to create the new objects to track. This is dangerous in indoor environments, due to the moving objects can be appear suddenly behind doors or other static objects that are not in the edges of the local map.

2.4 Objectives

After the review of the state of the art, and considering the discussion presented in the introduction, the objectives of this thesis are as follows:

- To develop a system that improves the perception stage of the robot in order to achieve a robust navigation in terms of safety and energy saving. The perception must deal with static and dynamic obstacles using a LIDAR, in order to collect the surrounding information, and a self velocity estimator with a GPS (in outdoors) or an improved odometry with an IMU (in indoors).

- Provide flexibility to the system for using it on commercial and developed robotic platform with heterogeneous sensors.

- The system should provide information about the occupancy and velocities of static and dynamic obstacles in the surroundings of the robot. This information should be useful to improve the performance of the planning and obstacle avoidance algorithms.

- To improve the developed system using probabilistic based methods in terms of occupancy and velocities of the obstacles, improving the robustness of the system from noise provided by measurement in the sensors.
- To test the system in indoor and outdoor environments, both simulated and real scenarios should be available.

Chapter 3

Platforms

In this chapter, the *Robotics Middleware* chosen and the hardware robotic platforms used in this thesis will be introduced. It is important to decide which platforms should be used at each scenario, for that reason, in the next sections the hardware platforms will be analysed to show their strengths and weakness. From the robotic commercial platforms to the robotic platform developed in the RobeSafe Research Group will be presented. Finally the different platforms will be compared in order to decide which is the most adequate platform for different environments described in ABSYNTHE project, (Section 1.2) and where the experiments of this thesis will be performed.

3.1 Robotics Middleware

In order to develop research and applications in robotics field, Several *Robotics Middlewares* have been released in the last years, such as *Carnegie Mellon Robot Navigation Toolkit (CARMEN)* [1], *Microsoft Robotics Development Studio (MRDS)* [2], MissionLab [3], *Advanced Robot Interface for Applications (ARIA)* [4], The Player Project [5] and

[1] carmen.sourceforge.net
[2] www.microsoft.com/en-us/download/details.aspx?id=29081
[3] www.cc.gatech.edu/ai/robot-lab/research/MissionLab
[4] robots.mobilerobots.com/wiki/ARIA
[5] www.playerstage.sourceforge.net

Robot Operating System (ROS) [6]. There are many other projects, but these are ones of the most representative. Only some of them have been maintained and in the last years, furthermore ROS has outperformed the other existing *Robotics Middlewares* (See Figure 3.1). Main characteristics of this platform are:

- It is *Open-Source*, so can be used in academy without increasing the cost of the project.
- It works with a large variety of operating systems, increasing its compatibility with other projects.
- It can be coded in several *high-level* languages.
- It includes a big, maintained and increasing repository of algorithms.
- It works with 2D and 3D simulators.
- It is compatible with a huge variety of commercial and researching robotic platforms. It is compatible with all the platforms and sensors that we are going to use in this thesis.

For these reasons among many others, ROS is nowadays the standard *Robotics Middleware*, as shown in Figure 3.1, that indicates the ranking of the most visited websites, where the *Ros.org* website has the rank 20000 among all websites in the world. Consequently, and due to fulfill requirements of this thesis, it is the platform that is going to use to test the proposals. Once a *Robotics Middleware* has been chosen, we are going to explain in detail how it works and highlights its strengths and weakness.

ROS [Quigley et al., 2009] was firstly released in 2007 by a group of researchers of *Willow Garage*, including some that had worked in the development of the *Player Project*. The philosophy of ROS is to develop a *Robotics Middleware* that can work *peer-to-peer*, are *multi-lingual, thin, free* and *Open-Source* oriented.

[6] www.ros.org

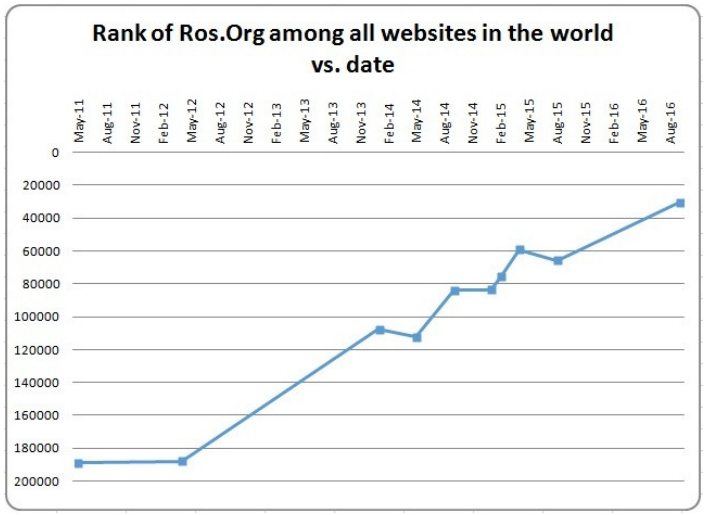


Figure 3.1: *Ros.org* website in the ranking most visited websites.

ROS is released in *distributions*, similar to the Ubuntu philosophy. Its first non-alpha distribution (*ROS Box Turtle*) was released in 2010. The first distributions were released without periodicity, but since 9th distribution (*ROS Jade Turtle*, 2015) were released annually in may. The odd numbered years releases are considered *ROS normal* and will be supported for two years. The even numbered years releases are *Long Time Support (LTS)* releases and will be supported for five years.

ROS can be installed in several operating systems, such as, Linux, Mac, Android and Arduino. The main installation support is for Ubuntu distributions of Linux and each LTS release of ROS has full support for one LTS distribution of Ubuntu.

As the project is Open-Source and it is the most popular *Robotics Middleware* nowadays, a huge repository of algorithms (from drivers to work with robots to visual computation ones) is hosted in the ROS webpage. That repository is also divided between the several distributions. Each distribution has changes in some core libraries making not fully compatible with previous developed algorithms, so that algorithms need to be maintained. Some common algorithms (for example, laser grid mapping ones) are maintained by *Open Source*

Robotics Foundation, Inc. (OSRF) foundation and released with each version but, so many others, are not maintained by the community.

ROS can work with robot and sensors teams. A core node synchronizes the whole system but the computation of each algorithm can be executed in several machines connected to it. This characteristic is especially important for this thesis.

In order to work with ROS some concepts and elements must be defined:

- **Nodes:** each individual program that is running in the system.
- **roscore:** a special node that must be always present in the system. That node is in charge of the synchronization of the whole system and contains the pre-requisites of a ROS-based system.
- **Package:** a collection of programs to perform one task (for example, the navigation package contains programs for local navigation and global path planning).
- **Topic:** a bus used by nodes to transmit data. These topics can be transmitted without a direct connection between nodes, meaning the production and consumption of data are decoupled. A topic can have various subscribers.
- **Message:** it is the information unit that can be shared between nodes using a *topic*. In ROS so many usual kind of information are defined (laser range information, image, odometry, etc.). Custom messages can be defined too. Each message (with few exceptions) has its timestamps (when the message is generated) and its *frame*. That is the part of the system (among the instructions to access each type of message) that standardize the communication tasks.
- **Frame:** Is the explicitly reference to a coordinate system in the environment, that is contained at each *message*, in order to increment the scalability and compatibility of the system.

- ***Transform (tf)*:** a special *topic* and *message* type that establish the relationship between the coordinate *frames* of the whole system. It is needed that these relations are always sent even if the relationship does not vary with time.

- **Services:** an asynchronous communication between *nodes* of type "answer/call".

Apart from the software repository, ROS includes a variety of useful software tools, such as **RViz** for data visualization, **RQT Tools** for system introspection and reconfiguration, **RosBag** for data logging and replay among so many others.

ROS is not linked to a particular simulator and has support for many of them, including Open-Source software like *Stage* or *Simple Two Dimensional Robot Simulator (STDR)*, and proprietary ones like *Webots* and *VRep* among many others. However, the 3D simulator *Gazebo*, maintained by the same foundation that maintains ROS becomes one of the widest used simulator.

Gazebo is a 3D simulator that includes rigid body physics (including friction and dynamics) and it can work with several robots at the same time. *Gazebo* is a parallel project to ROS (it started in 2002 within the *Player Project*) but each ROS distribution has official support for one distribution of *Gazebo*. This simulator can also work without ROS.

It is clear that ROS has many strengths, even more when it is compared with other *Robotics Middlewares*. However we are going to highlight some of its flaws that can be conflictive when a robotics project is developed in that platform:

- Each distribution is not fully compatible with the algorithms released for the previous ones. In that way, many submissions in the repository are not compatible with actual versions or they need maintenance.

- As the repository is collaborative, sometimes the testing and the information is not good enough or incomplete.

- A *roscore node* has to always be running in the system. As ROS support multi-robot and distributed systems, the communication with the master node it is crucial. If that communication fails or is not reliable enough, the whole system will fail. For that reason some *Multi-Master* approaches has been made, in a way that each robot or device is running its own instance of ROS and communicate with the others by TCP/IP or UDP services.

- The necessity of including the timestamp and *frame* of each *message*, and the relationship between all the coordinate systems (tf), even when that relationships does not vary in time, increases the bandwidth needed for that transmission. On the other hand, if that information were not included, a deep knowledge of the whole system would be needed to work with the information.

- Each sent *message* can be read by any *topic*. So, for distributed networks, security measures should be implemented.

3.2 Commercial Robotic Platforms

There are some manufacturers that provide robotic platforms that can be used in research, such as *Robotnik* [7], *SoftBank Robotics* [8] (previously known as *Aldebaran), Boston Dynamics* [9], *Clearpath Robotics* [10], *Robotis* [11], *Omron Adept MobileRobots* [12] (previously knowns as *MobileRobots* and *ActivMedia Robotics*) and some others. In the last years, many companies have started to develop and sell robots for industrial and research applications. Ones of the widely used platforms for researching area are the robots by *Omron Adept MobileRobot*s, such the well known *Pioneer family*, from its little

[7] www.robotnik.es
[8] www.ald.softbankrobotics.com
[9] www.bostondynamics.com
[10] www.clearpathrobotics.com
[11] en.robotis.com
[12] www.mobilerobots.com

version *AmigoBot* to the big brothers, *Seekur* and *Seekur Jr.* The RobeSafe Research Group has some of these robots to develop and test the projects (shown in Figure 3.2), from the *AmigoBot* to the *Seekur Jr.* In this thesis, the *Pioneer 3-DX, 3-AT* and *Seekur Jr.* have been checked in the Test Beds, due to they are the platforms that accomplish the system requirements.

Figure 3.2: RobeSafe Research Group's Robots.

3.2.1 Pioneer Robots

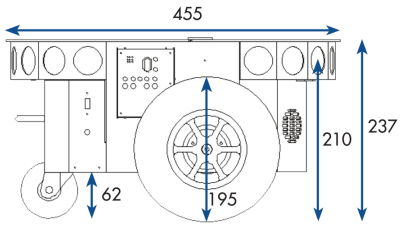
Pioneer is a family of mobile robots, both two-wheel and four-wheel drive, from the first *Pioneer 1* and *Pioneer AT*, followed by the second versions, *Pioneer 2-DX, -DXe, -DXf, -CE, -AT, -DX8, -AT8/AT8 Plus*, to the newest *Pioneer 3-DX* and *-AT* mobile robots. These small research and development platforms share a common architecture and core software with all other *MobileRobots* platforms, including *AmigoBot*, *PeopleBot V1*, Performance *PeopleBot* and *PowerBot*.

3.2.1.1 Pioneer 3-DX and Pioneer 3-AT

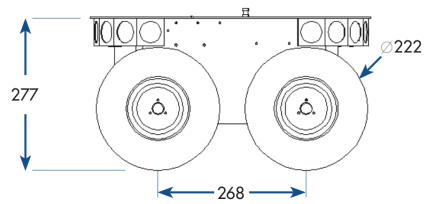
The *Pioneer 3-DX* and *Pioneer 3-AT* (shown in Figure 3.3(a) and Figure 3.3(b) respectively) are durable, differential drive robots for academic and researching purposes The Pioneer's versatility, reliability and durability have made them the most popular differential drive mobile robots in academic and researching for years. Unlike hobby and kit robots, Pioneer is ready to use, and will last through years of tough classroom and laboratory use.

(a) Pioneer 3-DX

(b) Pioneer 3-AT



(c) Pioneer 3-DX: dimensions in mm.



(d) Pioneer 3-AT: dimensions in mm.

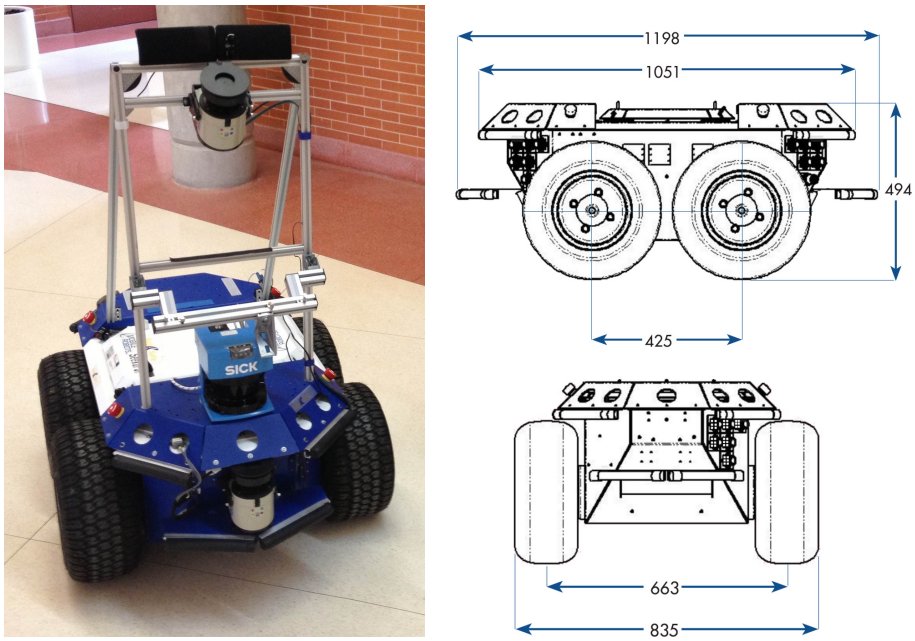
Figure 3.3: Pioneer 3 Robots.

The *Pioneer 3-DX* can achieve a maximum linear velocity of $1.2m/s$ and a maximum angular velocity of $300°/s$ with a payload of 17Kg. In the case of the *Pioneer 3-AT* achieve $0.7m/s$ and $140°/s$ respectively with a payload of 12Kg. The sensors included in both cases are a 500 *Cycles Per Revolution (CPR)* encoders and 8 sonars

in the front part. The rest of the characteristics will be shown in Section 3.4 in order to compare the different platforms used in this thesis.

3.2.2 Seekur Jr

The *Seekur Jr.*, shown in Figure 3.4, is an outdoor robot platform, also can operate in indoors with big open areas. Similar to the *Pioneer 3-AT* models, this platform is four wheel skid-steer differential drive all terrain, though with a tires of 40.6 cm and it is much stronger that the *Pioneer*, carrying a much larger payload (50 Kg) and is better protected against inclement weather.



(a) Seekur Jr (b) Seekur Jr: dimensions in mm

Figure 3.4: Seekur Jr: platform and dimensions.

This platform can achieve a $1.2m/s$ linear speed and a turn rate of $100°/s$. The body is made completely by sturdy aluminium. The robot include a segmented bumper array in the front and the back

and four emergency stop switches for safety. Also, an IMU is included in order to improve the odometry in the turns.The rest of the characteristics will be shown in Section 3.4 in order to compare the different platforms used in this thesis.

3.3 Developed Robotic Platforms

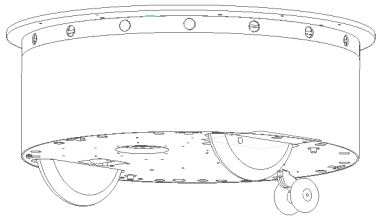
In order improve some of the characteristics of commercial platforms, as will be shown in the Section 3.4, and reduce its cost, a robotic platform has been developed from scratch. In that way, PROPINA (The name is the Spanish acronym for *"robotic research platform"*) has been developed in the RobeSafe Research Group, mainly in this thesis and in the thesis of Eduardo Molinos [Molinos, 2017]. In the next sections the details of the platform will be explained and finally a comparison between the comercial platforms and the developed one has been shown in Section 3.4.

3.3.1 PROPINA Platform

The platform has been developed as a robotic research platform to develop high-level applications using the *Robotics Middleware* ROS. It is a differential drive platform, designed to work indoors, and it is equipped with odometry and range (ultrasonic and infrared) sensors. In addition, a 3D model in *Gazebo* simulator has been designed to can be used as prior designing the actual application.

The embedded cards run ROS modules to control the motors and to perceive the information from sensors. In this way the perception is completely transparent to the remote control station. The modular design has been chosen to increase the functionality and autonomy.

PROPINA platform is equipped with these onboard sensors: 16 sonar sensors in a ring around the robot, 5 infrared sensors pointed to the ground in the front and the back of the robot to detect depressions of the terrain and optic encoders of 1000 CPR in the wheels. The logo and the preliminary design is shown in Figure 3.5. In the next sections the main components and characteristics of the platform will be explained.



(a) PROPINA design

PROPINA
PLATAFORMA ROBÓTICA PARA INVESTIGACIÓN

(b) PROPINA logo

Figure 3.5: PROPINA: design and logo.

3.3.1.1 PROPINA: Mechanical and structural design

Taking into account the previous experience using the commercial platforms, the objectives of the mechanical and structural design are:

- **Modular and versatile design:** Allowing add sensors or control modules. For that reason, the regular, symmetric and easy shapes has been selected. These shapes reduce the costs, make easier the manufacture and promote the modularity. Aluminium has been used on the chassis and fibreglass in the cover to lighten the platform. Additionally, the cover on the top of the platform facilitates the placement of additional sensors and structures.

- **Differential drive:** This traction system improves the manoeuvrability and agility of the platform. Rigid rubber wheels have been selected, similar to those used in scooters, due to several improvements that introduce to the platform:

  1. Have a good grip.
  2. Increase the payload of the platform.
  3. Improve the odometry, compared to pneumatic wheels due to the diameter of that wheels do not change depending on the temperature, neither the carried load, neither the pressure. Also, the narrow wheels selected make the robot's turns more precise, due to less contact area with the floor,

the distance between wheels keep more constant than the robots with wider tires.

- **Safety:** Any movable part of the robot should not be accessible to improve the safety. For that reason, the wheels must locate inside the structure, not allowing that the people around the robot can touch the wheels, or the wheels can trip over the some parts of the environment. Furthermore, the rounded shapes prevent it from getting stuck, consequently, improving the safety of the environment and the platform.

- **Robust:** The platform have been made for research, that's means, the design have to support an exhaustive use in different environments and scenarios.

- **Stability:** Due the possibility of the addition of modules on the top of the platform, increasing the height, the design have to consider this. The design maximizes the space between the wheels in order to increase the stability. In addition, a caster wheel has been places in the back part of the base. Additionally, the main weight of the robot, as are the batteries and the motors should be symmetrically located on the base to keep the mass centre near to the centre of the platform.

Based on these objectives, a base module and a sonar ring module have been develop, both of them with cylinder shapes, shown in Figure 3.6.

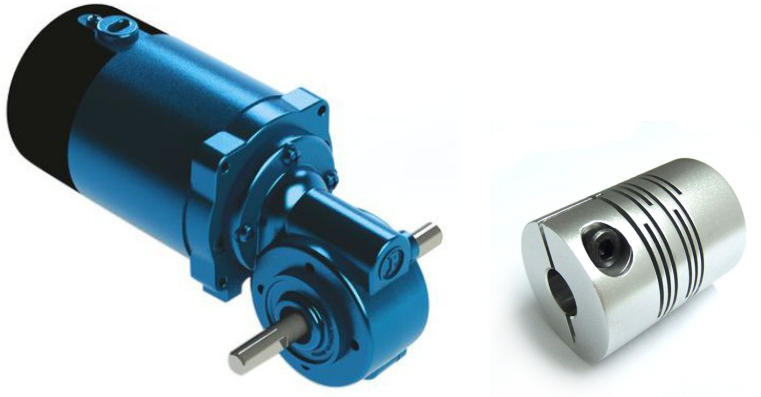
3.3.1.2 PROPINA: Electronic design

The first step in the electronic design is to chose the actuators. In this case, permanent magnet DC motor with a gearbox has been chosen due its high torque. *PM10-0033* by *Parvalux* [13] (shown in Figure 3.7(a)) has been chosen for this propose, with a speed of 195 rpm, and 2.5Nm of torque. The coupling from the motor to the wheels is made by a flexible beam couplings (shown in Figure 3.7(b))

[13] www.parvalux.com

Figure 3.6: PROPINA platform.

to prevent that the torsions and impacts in the wheels damage the motor's gearbox.



(a) PM10 Motor and gearbox.

(b) Motor-wheel flexible coupling

Figure 3.7: PROPINA: Motor and flexible coupling to the wheel.

The second step is to decide the board that will control the motors and adapts the information from the sensors. The motors are controlled by an *Arduino Mega* [14] board connected through a *Pololu VNH5019* [15] driver board. This driver board is a Dual H bridge for

[14] www.arduino.cc/en/Main/arduinoBoardMega
[15] www.pololu.com

DC motors connected to the *Arduino Mega* as it is shown in Figure 3.8. A software driver has been developed to run in the *Arduino Mega* board. This program reads the sensors and commands the actuators at the same time that send and receive all the data in a ROS format thought the *ROSserial libraries for Arduino*, allowing to control the robotic platform using a remote ROS client. The maximum linear and turn velocities had been limited by software to $1m/s$ and $100°/s$ for safety.

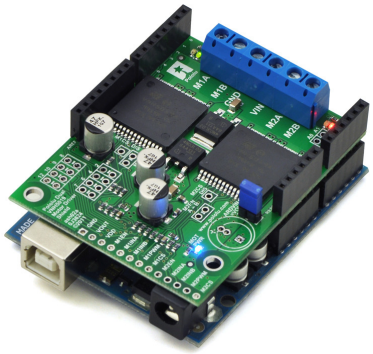


Figure 3.8: PROPINA: Motor's Driver and Arduino boards.

The third step in the electronic design is to choose the appropriate sensors. The platform includes the following sensors:

- **Encoder sensors:**

  The selection of a proper encoder is crucial, due to this sensor is needed to achieve a good odometry. Odometry is the base of the localization and it is essential to achieve a good motor control stage. For both reason, a high-resolution and optical (high immunity to interference) encoder has been chosen. For this thesis, the *RI38* optical encoder by *Hengstler* [16] (shown in Figure 3.9), with an 1000 CPR, that is twice more resolution than the Pioneer's encoders. It has been the best election in terms of price, size and characteristics.

[16] www.hengstler.de/en

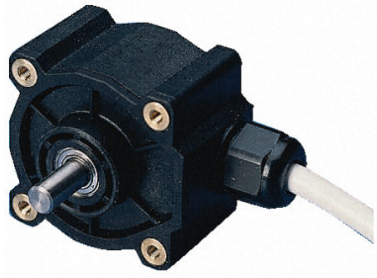


Figure 3.9: PROPINA: Wheel encoder RI38.

- **Infrared range sensors:**

  To keep the integrity of the robot, is crucial to detect the ramps or gaps in the floor, and evaluate if the platform can overcome them or should avoid them. For that purpose, a total of five infrared range sensors pointed to the floor has been chosen, three of them are located in the front of the base and two in the back. With the sensors in that location, the robot knows in every moment the distance from the base to the floor in front and in the back of the wheels. The sensor chosen is the *SHARP GP2Y0A41SK0F* [17] (shown in Figure 3.10) due to the extended use in robotics and the low price.

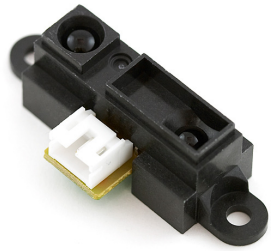


Figure 3.10: PROPINA: Infrared Sensor.

  This sensor has a range of 40 cm and a transference function as shown in Figure 3.11. The five sensors has been connected to

[17] www.sharp-world.com

the analog inputs (A0-A4) of the *Arduino Mega* board. To minimize the use of the micro-controller of the *Arduino Mega* board, the Raw Voltage data from the sensors is send to the main PC, instead of computing the distance in the Arduino Mega. In the main PC the distance measured by the infrared sensor is computed following this assumption: a fifth order polynomial approximation of the transference function, given by Equation 3.1 as shown in Figure 3.11.

$$d_{GP2Y} = -0.076 \cdot V_{in}^5 + 0.78 \cdot V_{in}^4 - 3.2 \cdot V_{in}^3 + 6.76 \cdot V_{in}^2 - 7.8 \cdot V_{in} + 4.84 \tag{3.1}$$

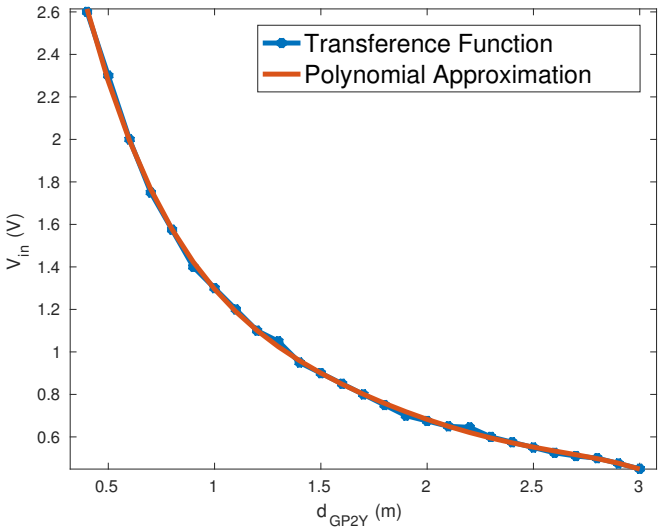


Figure 3.11: Infrared Sensor: Transference function (*blue*) and the third order polynomial approximation (*red*).

- **Sonar range sensors:**

  To detect the environment around the robot is needed to include some range sensors. For that reason, a sonar module has been designed to include 16 sensor *LV-MaxSonar-EZ2 (MB1010)* by

Maxbotics [18], shown in Figure 3.12, selected due to its low price and very small size.

Figure 3.12: PROPINA: Sonar Range Sensor.

This sensors have a range of 6.45m, are powered from 2.5V to 5.5V, work at 42kHz and provide output measurements up to 20Hz. It is possible to read the output in three different ways: analog, serial and *Pulse-Width Modulation (PWM)*. In this version of PROPINA platform, only eight of the sixteen sonar has been implemented, connecting these eight sensors to the analog inputs ($A0-A7$) of *Arduino Mega* board. The sensors form two groups of four each, as the manufacturer recommended, and the simultaneous transmissions are between the sensor most separated to minimize cross-talk. This configuration is possible using the trigger input of the sensors. To filter the noisy measures form the sonar, a median filter in a 1,5 s window has been implemented. It is possible to enlarge the platform until sixteen sonars, due to can be placed all of them in the ring sonar structure, allowing his reading using the analog inputs of the second *Arduino Mega* board that the platform have.

3.3.1.3 PROPINA: Control system

The control system of the platform is formed by the motors and encoders of the wheels that are connected to the *Arduino Mega* board, as mentioned in the section 3.3.1.1. The motors are connected through a *Pololu VNH5019* driver board and the encoders channels are read

[18] www.maxbotix.com

using 4 digital inputs of the *Arduino Mega*. The complete control system is shown in Figure 3.13.

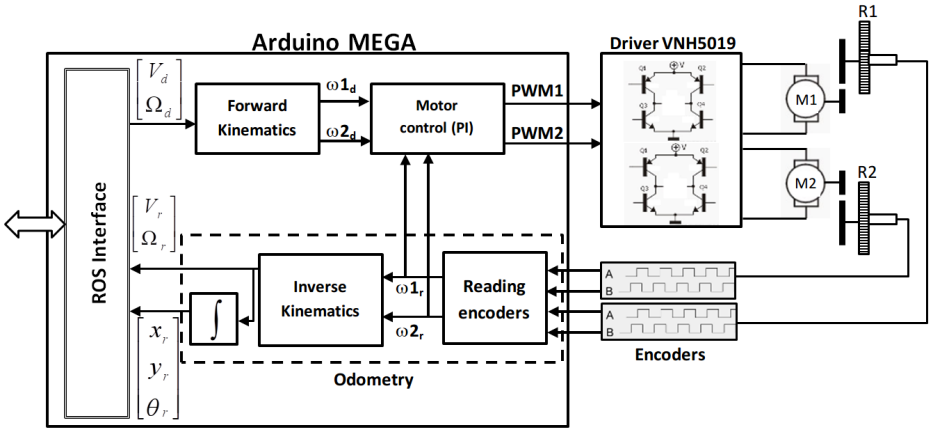


Figure 3.13: PROPINA: Control System diagram.

The description of each control block of the system is as follow:

- **The encoders reading:**

  The encoders channels, *CHA* and *CHB*, has been connected to 4 digital inputs of the *Arduino Mega* board. As all the signal's edges from the encoders generate interruptions, then the resolution of the encoders increases by 4, due to they are reading in quadrature. The counters assigned to the encoders are read and reset every 10ms to obtain the angular velocities of the wheels ($\omega 1_r$, $\omega 2_r$).

- **Motor control:**

  The control of the motors are performed by periodic interruptions every 10 ms. The interruptions should generate the PWM signal to obtain the wheels velocities ($\omega 1_r$, $\omega 2_r$) that follows the reference computed in the forward kinematics block ($\omega 1_d$, $\omega 2_d$), not influenced by the charge load.

  In order to adjust the *Proportional-Integral-Derivative (PID)* controlled parameters, an experimental identification of the

plant has been perform registering several measures in open-loop. First, different codes (from -255 to 255) has been sent to generate the PWM signal to the motor, then the wheel velocity has been registered. Second, with these measures is possible to identify the model, including the dead zone of the motor. Third, a *Simulink* [19] diagram has been created in order to adjust the PID parameters, shown in Figure 3.14.

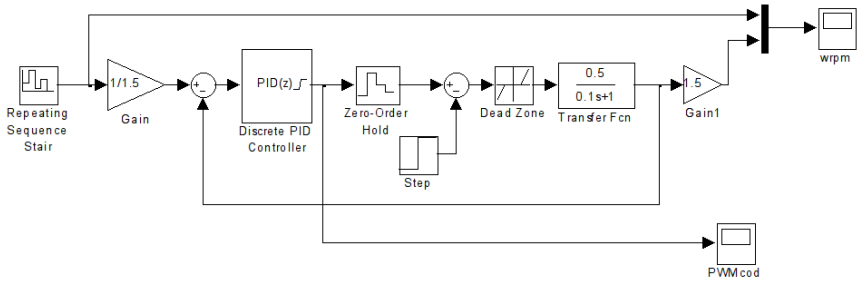


Figure 3.14: Simulink diagram.

The system input is the wheel velocity reference in *rpm*, then it is converted to encoders counts, using $N \cdot T/60$, where N is the CPR of the encoder and T the sampling period. In this way, it is possible compare directly the real wheel from the reading encoders block with the velocity reference. The PID parameters has been tuning experimentally in order to obtain an acceptable damping response and a setting time less than 100 ms. Finally, in the fine adjustment, the derivative part has been removed and a *anti-windup* block has been added in the integral part of the control.

- **Forward kinematics:**

  The inputs of the forward kinematics block are the reference velocities (linear velocity V_d and angular velocity Ω_d). Then, the block computes the angular wheels velocities ($\omega 1_d$, $\omega 2_d$)

[19] mathworks.com/products/simulink.html

following the differential drive robot equations (Equation 3.2), where r is the wheel radius and L the space between wheels.

$$\begin{aligned} \omega 1_d(rpm)) &= \frac{60}{2\pi r}(V_d + \frac{L\Omega_d}{2}) \\ \omega 2_d(rpm)) &= \frac{60}{2\pi r}(V_d - \frac{L\Omega_d}{2}) \end{aligned} \tag{3.2}$$

- **Inverse kinematics:**

  The inverse kinematics block is able to obtain the linear and angular velocities of the robot (V_r, Ω_r) from the wheel's angular velocities that came from the encoders reading block. The equations to obtain the robot velocities (Equations 3.3) are the inverse equations shown previously (Equations 3.2).

$$\begin{aligned} V_r(m/s) &= \frac{\omega 1_r(rpm) + \omega 2_r(rpm)}{2} \cdot \frac{2\pi r}{60} \\ \Omega_r(rad/s) &= \frac{\omega 1_r(rpm) - \omega 2_r(rpm)}{L} \cdot \frac{2\pi r}{60} \end{aligned} \tag{3.3}$$

- **Odometry:**

  To obtain the odometry and update the robot pose (X_r, Y_r, θ_r) the increment of the *encoder counts* during the sampling time are used $(\Delta 1, \Delta 2)$. These *encoder counts* are converted to distance using the following equations 3.4:

$$\begin{aligned} d1r(m) &= \Delta 1 \cdot \frac{2\pi r}{4N} \\ d2r(m) &= \Delta 2 \cdot \frac{2\pi r}{4N} \end{aligned} \tag{3.4}$$

  where N is the CPR of the encoder (1000 in our case). From these equations it is possible to integer the robot pose at each sampling step (k) following Equations 3.5:

$$
\begin{aligned}
x(k) &= x(k-1) + \frac{d1+d2}{2} \cdot \cos\theta(k-1) \\
y(k) &= y(k-1) + \frac{d1+d2}{2} \cdot \sin\theta(k-1) \\
\theta(k) &= \theta(k-1) + \arctan\left(\frac{d1-d2}{L}\right)
\end{aligned}
\tag{3.5}
$$

- **ROS interface:**

  At last, the ROS interface block is in charge of receiving the velocities references from external computer, in ROS format, and to send the results of the odometry block, allowing to read the odometry directly in the external computer using ROS. In order to increase the rate of the odometry reading, is necessary to decrease the computation load in the *Arduino Mega*. For that reason, the ROS interface only sends the encoders count difference between time steps and the time step to the main PC. In this way, the platform can provide the odometry up to $100Hz$.

3.3.1.4 Modelling PROPINA in Gazebo simulator

Simulators in robotics are crucial due to allow testing the designing previously to the final application. In this way, It is possible test different sensors, configurations and environments, saving cost and time. For that reason, a simulation model of the developed platform is necessary, to complete the project, and it has been developed by Eduardo Molinos in his thesis [Molinos, 2017].

Gazebo is a 3D multi-robot simulator. This simulator is able to deal with several robotic platforms, sensors, objects and scenarios. *Gazebo* simulates rigid-bodies physics, including dynamics, interactions between several objects and realistic behaviour of the sensors. The model in *Gazebo* (shown in Figure 3.15) has been built simplifying the model created in *SolidWorks* and it has the components following the diagram of the PROPINA model in *Gazebo* (Figure 3.16).

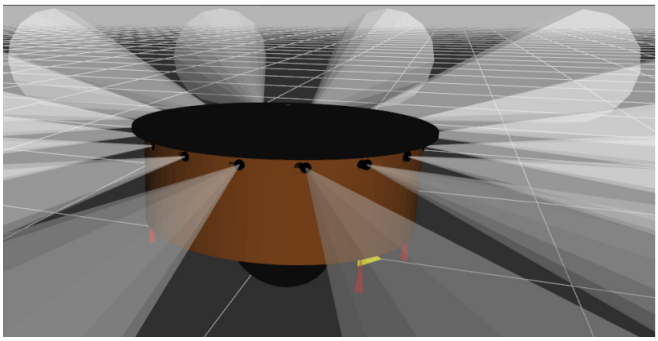


Figure 3.15: *Gazebo* model of PROPINA.

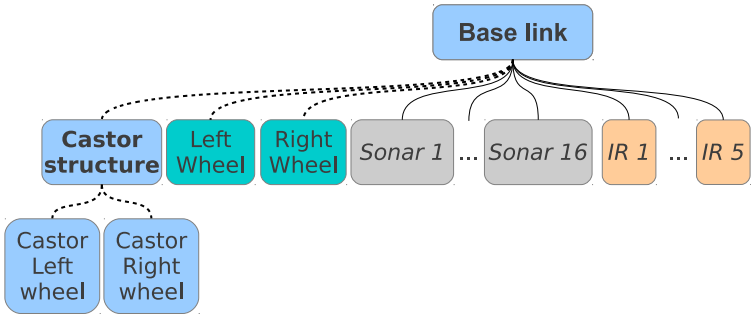


Figure 3.16: *Gazebo* diagram of PROPINA.

3.3.2 RoboShop Platform

In order to increase the usability of our previous developed platform, an extension of the PROPINA platform have been develop through RoboShop project. In this project, a complete robot navigation system, that can be used as a *shop assistance* or a *robotic guide* in a museum or in a mall has been developed. This project uses the robotic platform PROPINA, explained in Section 3.3.1, as base.

The RoboShop diagram of the whole system is shown in Figure 3.17. This diagram shows the relationships and connections between the PROPINA platform (top left) and the added hardware (bottom left). The right part of the diagram shows how the robot interact with the users through the *Graphical User Interface (GUI)* (bottom) and how it can shares information (top) with cloud services (data bases, central system...).

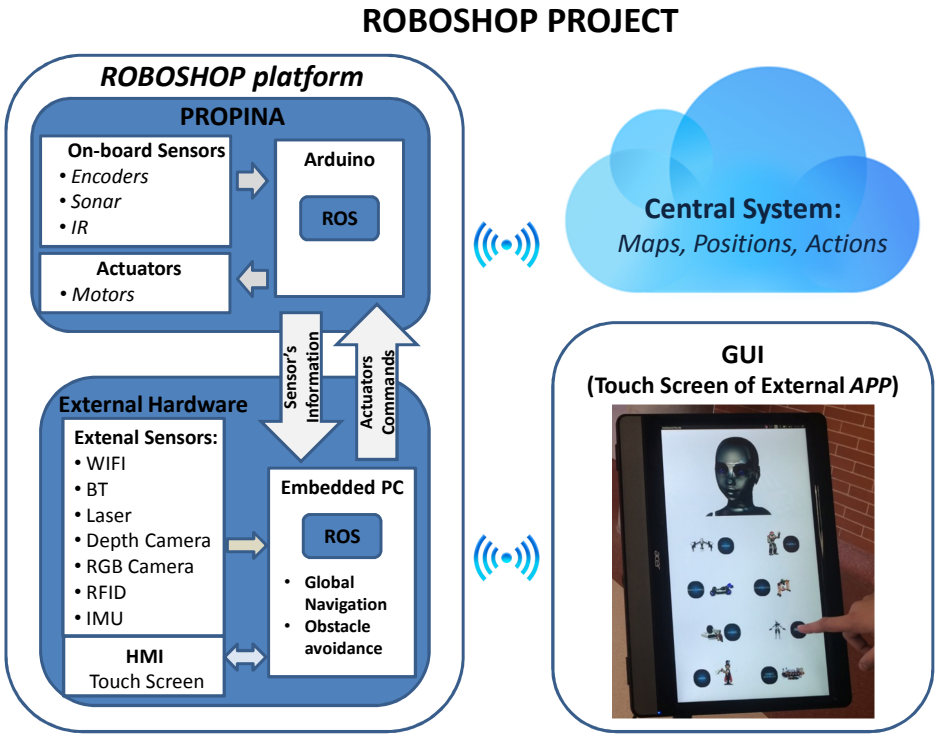


Figure 3.17: Diagram of RoboShop project.

To develop the project, it is necessary to outfit the PROPINA platform with more sensors to measure the environment, such as LIDAR, depth camera, RGB camera, and an IMU to improve the odometry. Additionally, connectivity is necessary to allows read possible beacons in the environment (*Bluetooth* of *Radio Frequency IDentification (RFID)*) and *WiFi* to allow the platform send or received information from a central system. Also an HMI based on a touch screen has been added (see Figure 3.18). The details of the design and the elements added to the PROPINA platform are shown as follow:

- Hokuyo URG-04LX LIDAR that can be used parallel or pitched to the ground to obtain measures in 3D. Adding a LIDAR to the platform, improves the perception stage due to increase the number, the accuracy and the max distance of the measures, compared to sonar sensors.

Figure 3.18: RoboShop platform.

- Depth camera *Asus Xtion Pro* that can provide 3D information much richer information than the LIDAR. However, the measures are more noisy and in a limited range of distance and field of view.

- A camera pointed to the ceiling to obtain measures that are independent of the change in environment.

- Touch Screen that includes speakers, as HMI, mounted on an structure to increase the height, providing an comfortable interaction between the robot and the user.

- Colibri IMU attached to the center of rotation to improve the odometry.

- An embedded PC that executes the navigation systems (perception, planning and control stages). Also this PC provides WiFi and Bluetooth connections to the platform.

- Connectivity: the PC provides *WiFi* and *Bluetooth* connections to the platform. Also a RFID reader has been added. In this way, the platform can read beacons located in the environment and share information with other systems

3.3.2.1 Modelling RoboShop in Gazebo simulator

In order to simulate the RoboShop platform in *Gazebo*, an structure and the sensors explained in section 3.3.2 has been added to the PROPINA simulated platform in [Molinos, 2017].

Figure 3.19 shows the simulated model in *Gazebo*, based on the model explain si Section 3.3.1.4 , adding the elements introduced before.

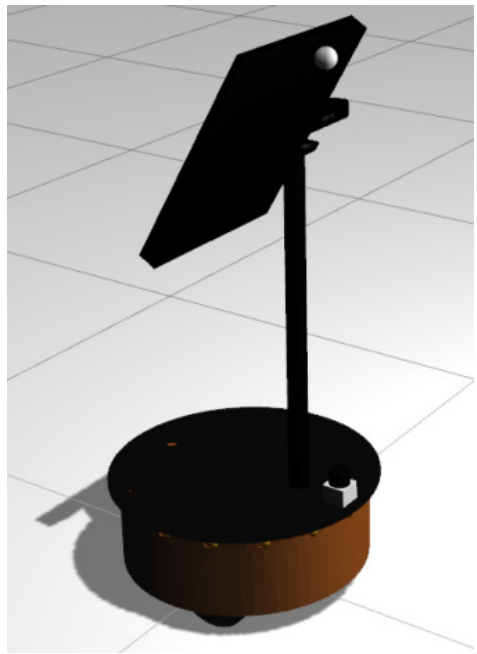


Figure 3.19: *Gazebo* model of RoboShop.

The diagram of the RoboShop model that contains the added components and the PROPINA components are shown in Figure 3.20.

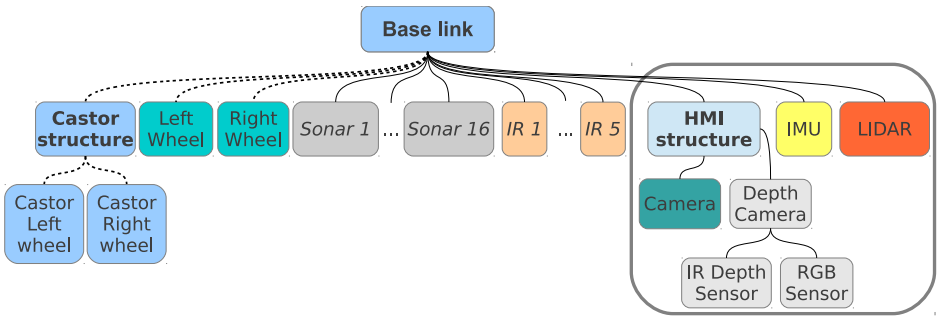


Figure 3.20: *Gazebo* diagram of RoboShop.

3.4 Robotic Platforms Comparison

In order to compare the commercial platforms with the developed one, Table 3.1 shows the main characteristics of them. Some of the remarkable characteristics are the referring to terrain that can be traversable by the platform. *Max Trav. Step*, is the highest step that the robot can overcome. *Max Trav. Gap*, is the longest gap in the ground that the robot can traverse. *The Max Trav. Grade* is the maximum ramp that the platform can ascend. In the Section 3.4 will be compared them it in detail.

Once that the different commercial and developed robotic platforms has been introduced, is necessary to compare them to test the performance of each one. The platforms compared are the three that can be used in indoors, *Pioneer 3-AT*, *Pioneer 3-DX* and our own platform, PROPINA. The characteristics of the *Seekur Jr.* platform are shown in Table 3.1 to introduced all the platforms in the same way, but is out of the comparison due to is a big outdoors robot. Once the characteristics have been described in the previous sections and in Table 3.1, the PROPINA platform has some advantages over the commercial platforms:

- The payload is a 20% higher, then the range of application is wider.
- The top part of the platform (black cover) has different screw holes, this next to the symmetric of the platform, allowing to

Table 3.1: Pioneer 3-DX, 3-AT and PROPINA comparison

	P3-DX	**P3-AT**	**Seekur Jr**	**PROPINA**
Drive	Differential	Skid Steering	Skid Steering	Differential
Tires	19 cm Foam-filled rubber	22,2 cm Reenforced Pneumatic	40,64cm (16") Pneumatic	17,5 cm Rigid rubber
Turn Radius	0 cm	0 cm	0 cm	0 cm
Swing Radius	26.7 cm	34 cm	52 cm	33 cm
Max Lin. Speed	1.2 m/s	0.7 m/s	1.2 m/s	1 m/s (lim. by software)
Max Ang. Speed	300 $^{\circ}/s$	140 $^{\circ}/s$	100 $^{\circ}/s$	100 $^{\circ}/s$ (lim. by software)
Max Trav. Step	2.5 cm	10 cm	12 cm	2 cm
Max Trav. Gap	5cm	15 cm	$\sim$ 20cm	3 cm
Max Trav. Grade	25%	35%	75 %	25%
Traversable Terrain	Indoor, wheelchair accesible	Asphalt, flooring, sand and dirt.	All terrain	Indoor, wheelchair accesible
Dimensions	381 mm width 455 mm length 237 mm height	467 mm width 508 mm length 277 mm height	835 mm width 1198 mm length 494 mm height	560 mm diameter 245 mm height
Robot Weight	9 Kg	12 Kg	77 Kg	19 Kg
Payload	17 Kg	5 Kg (Asphalt) 12 Kg (Tile)	50 Kg	20Kg
Autonomy	8-10 hours (3 batteries, no accessories)	2-4 hours (3 batteries, no accessories)	3-5 hours (no accessories)	4-6 hours (2 batteries)
Encoders	500 CPR	500 CPR	1024 CPR	1000 CPR
Sensors Included	8 Frontal Sonar	8 Frontal Sonar	segmented bumper array, IMU	16 Sonar Ring, 5 IR (floor)
Approx. Price	$\sim$ 4800€	$\sim$ 8900€	$\sim$ 28000€	$\sim$ 3000€

attach easily and in any direction, any kind of sensor, structure or HMI without having to drill the robot, on the contrary that in the Pioneer robots.

- Cheaper platform, almost the half price that the *Pioneer 3-DX* and the third part of *Pioneer 3-AT*.

- Incorporates infrared sensors point to the floor, to avoid the

stairs or steps higher than the platform could overcome.

- It is safer, due to the wheels are located inside the structure, not allowing that the people around the robot can touch the wheels, preventing possible damages, neither the wheels to trip over some parts of the environment. Additionally, the rounded shapes prevent it from tripping with the corners or getting stuck.

The last part to compare, is the odometry performance. To evaluate this parameter, the *University of Michigan Benchmark (UMBmark)* [Borenstein et al., 1996] has been used. UMBmark, is a *Bidirectional Square Path* experiment, with a length of 4x4 m square path, as shown in Figure 3.21, that has to perform five runs each in clock-wise (*cw*) and counter-clockwise (*ccw*) directions, defined as follows:

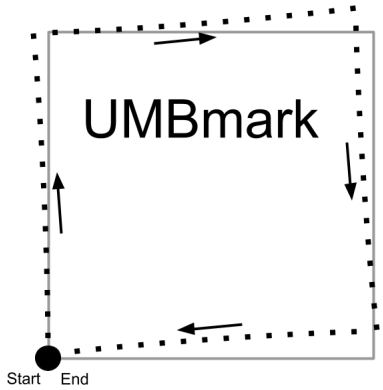


Figure 3.21: UMBmark path.

1. At the beginning of the run, measure the absolute position (and, optionally, orientation) of the vehicle and initialize the onboard odometric starting position to that position.

2. Run the vehicle through a 4x4 m square path in *cw* direction, making sure to:

   - Stop after each 4 m straight leg.

- Make a total of four 90° turns on the spot.
- Run the vehicle slowly to avoid slippage.

3. Upon return to the starting area, measure the absolute position (and, optionally, orientation) of the vehicle, using a fix reference, such as walls.

4. Compare the absolute position to the robot's calculated position, based on odometry. The result is a set of return position errors caused by odometry and denoted εx, εy, $\varepsilon\theta$:

$$\begin{aligned} \varepsilon x &= x_{abs} - x_{calc} \\ \varepsilon y &= y_{abs} - y_{calc} \\ \varepsilon\theta &= \theta_{abs} - \theta_{calc} \end{aligned} \tag{3.6}$$

   Where, εx, εy, $\varepsilon\theta$ are the position and orientation errors due to odometry. $x_{abs}, y_{abs}, \theta_{abs}$ are the absolute position and orientation of the robot. $x_{calc}, y_{calc}, \theta_{calc}$ are the position and orientation of the robot as computed from odometry.

5. Repeat steps 1-4 for four more times.

6. Repeat steps 1-5 in *ccw* direction.

It the experiment a path of 3 x 3m , instead 4 x 4m, have been used due to the lack of precision in some platforms. In addition, only 4 runs could be perform, due to the odometry errors present in some platform, forcing to stop the experiment due to the deviation from the ideal path, leaving the free space, in the fourth run.

After conducting the UMBmark experiment, the authors suggest to consider the center of gravity of each cluster of position and orientation errors due to odometry, obtained by Equation 3.6, as representative for the odometry errors in *cw* and *ccw* directions. These centres of gravity are given by Equation 3.7:

$$x_{c.g.,cw/ccw} = \frac{1}{n}\sum_{i=1}^{n} \varepsilon x_{i,cw/ccw}$$
$$x_{c.g.,cw/ccw} = \frac{1}{n}\sum_{i=1}^{n} \varepsilon x_{i,cw/ccw} \tag{3.7}$$

where $n = 4$ is the number of runs at each. The absolute offsets of the two centres of gravity are defined by Equation 3.8:

$$r_{c.g.,cw} = \sqrt{(x_{c.g.,cw})^2 + (y_{c.g.,cw})^2}$$
$$r_{c.g.,ccw} = \sqrt{(x_{c.g.,ccw})^2 + (y_{c.g.,ccw})^2} \tag{3.8}$$

Finally, the authors define the larger value among $r_{c.g.,cw}$ and $r_{c.g.,ccw}$ as the measure of odometric accuracy for systematic errors:

$$\varepsilon_{max,syst} = max(r_{c.g.,cw}, r_{c.g.,ccw}) \tag{3.9}$$

This thesis focus on real applications and such as authors suggest, the average of the centres of gravity must not use to take into account the largest possible odometry error, that means the worst scenario. In addition, the final orientation ($\varepsilon\theta$) is not explicitly consider to obtain $\varepsilon_{max,syst}$, because the systematic orientation errors are implied by the final position errors.

Figure 3.22 shows the results of UMBmark performed by a *Pioneer 3-DX*, *Pioneer 3-AT* and RoboShop platforms, where the *stars* are the odometry errors at each run, the *circles* are the clusters of the odometry errors and the *crosses* are the centre of gravity of each cluster. These benchmark has been performed with the RoboShop platform, to avoid having to disassemble the additional hardware. This should be taken into account because the weight of the entire platform increases and the mass center rises, which can worsen the results.

Table 3.2 shows the measure of odometric accuracy for systematic errors of each platform, where the better performance of the developed platform RoboShop are shown.

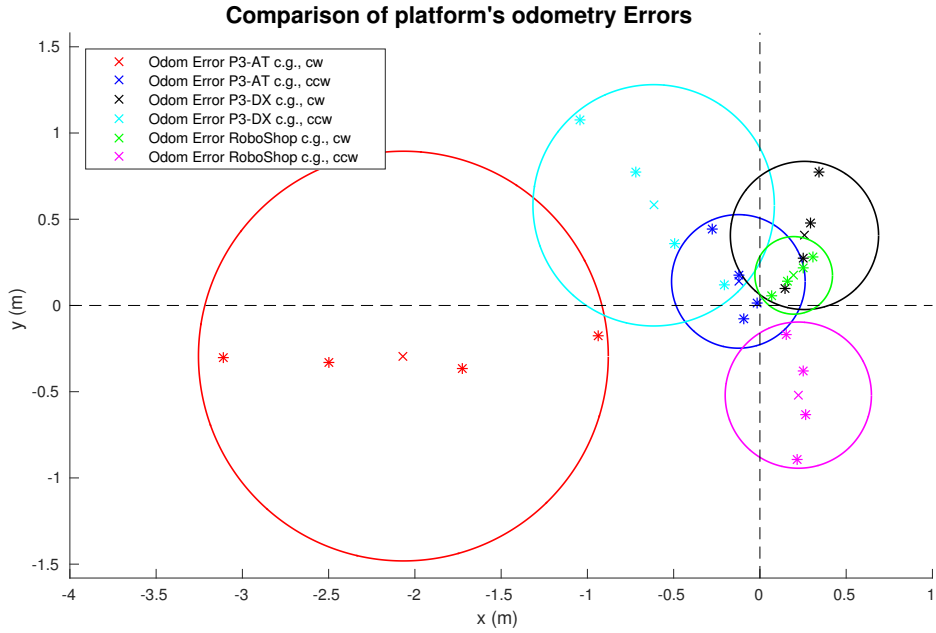


Figure 3.22: Comparison of platform's odometry based on UMBmark.

Table 3.2: UMBMark: Odometric accuracy for systematic errors

	Pioneer 3-DX	**Pioneer 3-AT**	**RoboShop**
$\varepsilon_{max,syst}$	0.8466	2.0875	0.5656

After testing the odometer of the platform through the UMBmark, that means in small travel distance, other approach to compare the platforms odometry is in a long travel loop. This experiment has been performed in an indoor environment around the second floor of the *Polytechnic School of University of Alcalá*. The path travelled is a square of 70 m x 70 m approximately, with a travel distance around 260 m. First, the robots start in a initial position marker. Second, performs the square remotely controlled and close the loop until the initial position marker. Finally, the difference between the real finish pose and the initial position marker has been manually measured to take into account in the comparative results shown in Table 3.3.

Table 3.3: Large Travel Odometry Errors.

	Pioneer 3-DX	**Pioneer 3-AT**	**RoboShop**
ε_x *(m)*	-9.2670	-153.5110	21.4878
ε_y *(m)*	47.7770	165.4110	17.2825
ε_θ *(°)*	159.0000	-264.0000	-33.3389
Euclidean dist. Error (m)	48.6674	225.6688	27.5756

Figure 3.23 shows the path performed based on the odometry info from the different platforms. The red line shows the Pioneer 3-AT path, where is possible to identify that the turns measured by the odometry is less than the real robots turns. On the other hand, in the case of the Pioneer 3-DX odometry (green line), the turns measured by the odometry is bigger than the real robots turns. Finally, the magenta line represent the path performed by RoboShop, which is closer to the real square path, proving the better performance of the developed platform.

The improvement on the odometry performance of the developed platform is due to several factors:

- The better odometer sensors chosen (1000 CPR, instead of 500 CPR of the Pioneer).
- The higher updating frequency of the odometry (25 Hz in RoboShop compared to the 10 Hz in *Pioneer Robots*)
- The narrow rigid rubber wheels chosen that have a good grip, prevents the slipping effect, its diameter not depend on the carrier load neither the pressure. Also, the narrow wheels keep the distance between wheels more constant, especially important in the turns.

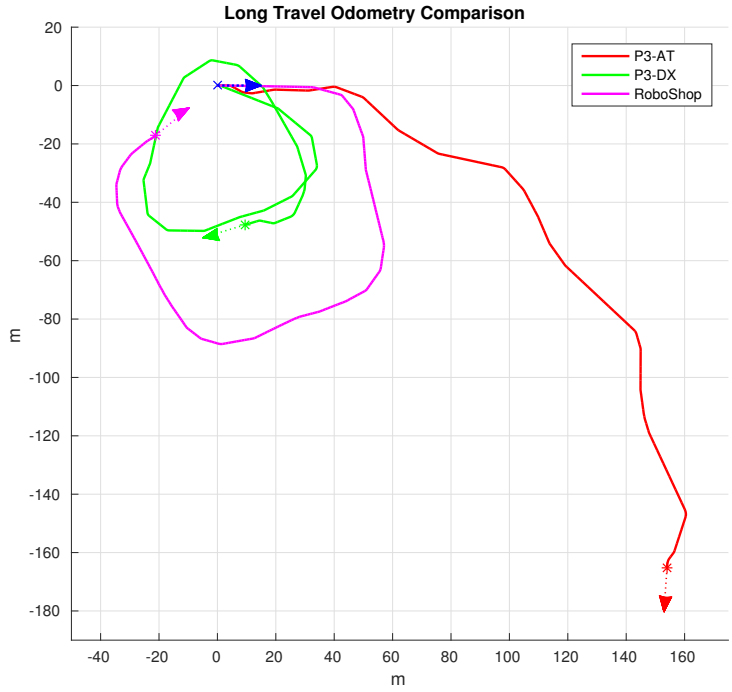


Figure 3.23: Large Travel Odometry Comparative

3.5 Conclusions and contributions

In this chapter, the *Robotics Middleware* (ROS) has been introduced. Also, some of the commercial robotic platforms commonly used in research have been analysed and compared with our developed platform PROPINA in Section 3.4 in order to decide which platform is suitable for each different scenario. Based on their characteristics; *Seekur Jr* is an outdoor platforms with all terrain capabilities, *Pioneer 3-AT* is able to work in indoor (similar to *Pioneer 3-DX*) and outdoors, although it is necessary to improve its odometry as shown in Section 3.4. This odometry improvement will be shown in the next chapter. Several improvements have been introduced in the developed platform RoboShop, that is suitable for indoors, such as the more accurate odometry, higher payload and safer design taking into account that the robotic platforms should share scenario with humans.

Chapter 4

Dynamic Obstacles Mapping Proposal

Modelling the environment is an important skill for mobile robots, even more in changing and dynamic environments. For that reason, and according to the system requirements, it is important to choose the correct hardware and software platforms. The robotic platforms have been explained in the Chapter 3. In the next sections, the algorithms used to deal with the static and dynamic obstacles in the environment, using heterogeneous sensors in the different platforms will be shown.

The remaining of the chapter is organised as follows: First, we will describe an odometry improvement approach proposed to deal with the dead reckoning errors (Section 4.1). Next, in Section 4.2 a laser pose calibration approach will be presented. An application of the previous Section is shown in Section 4.3. Then, Section 4.4 will show our DATMO approach, and its improved version in Section 4.5. Finally, in Section 4.6 a discussion about the results will be provided.

4.1 Odometry improvement

One of the first problem in mobile robots, is answer to the question, *"Where am I?"*, as was introduced in the Section 1.1 that corresponds to the Localization stage. In indoors with UGV this problem is usu-

ally achieve based on *dead-reckoning*. This technique is based on the knowledge of the starting point and the continuous estimation of bearing and speed. These parameters can be estimated from the odometry. Although our developed platform has a better odometry than commercial platforms, due to the accumulative errors that suffer this approach (Section 2.1.1), it is necessary to improve the odometry. For that reason, an IMU has been added to the robotic platform in order to perform an INS, as has been presented in our work [Pintor et al., 2016].

The IMU device behaves as an *Attitude and Heading Reference Systems (AHRS)* reference and orientation system, consisting of a set of *Microelectromechanical Systems (MEMS)* sensors: accelerometers, gyroscopes and magnetometers. The INS has accumulative errors too, due to the bias deviation of the gyroscopes and accelerometers of the IMU. For that reason, some IMU also include magnetometers to fuse the information with the gyroscopes in order to compensate their bias. The IMU usually provides the position and orientation through an EKF to filter the data. Also offers the possibility of using the quaternions, of great relevance both by its simplicity in handling the necessary kinematic equations in the INS, as well as the disappearance of the *gimbal-lock* effect produced by the gyroscopes.

It is important to remark that the magnetometers are affected by nearby magnetic objects or magnetic fields produced by electric devices, such as engines, that provides disturbances in the earth's magnetic field. These elements, such as metallic doors, elevators, metallic structures, are very common in indoors environments. For that reason, regarding the methods used to calculate orientation, it has been considered important to establish a difference between magnetometers and inertial sensors. On the one hand are the results obtained from the IMU, in which the magnetometers are involved, and on the other hand, the results of using only the filtered inertial sensors. This will allow to conclude which of the two methods is the most appropriate.

In addition, it is necessary to calibrate the IMU parameters *intrinsically* (determination of biases of gyroscopes and accelerometers, compensation of ferromagnetic errors, etc.) and *extrinsically* (more

suitable position to install the IMU).

To compare the performance of this proposal, two IMUs will be used over the *Pioneer 3-AT*, due to it has one of the worst odometry, as it has been shown in Section 3.4.

- One designed for general purpose: *Trivisio Colibri*.
- Another specific for use in mobile devices: *Invensense MPU6500* (Accelerometer and Gyroscope) and a *Yamaha YAS537* Magnetometer.

The proposed system is shown in Figure 4.1. First, the IMU is calibrated, in order to compensate the errors of the magnetometers and bias of gyroscopes. Once the different sensors have been calibrated, it is necessary to filter the orientation to obtain an improved version of the IMU orientation. Finally, these data are fused with the robot's odometry using an EKF, thus obtaining an improved version of odometry.

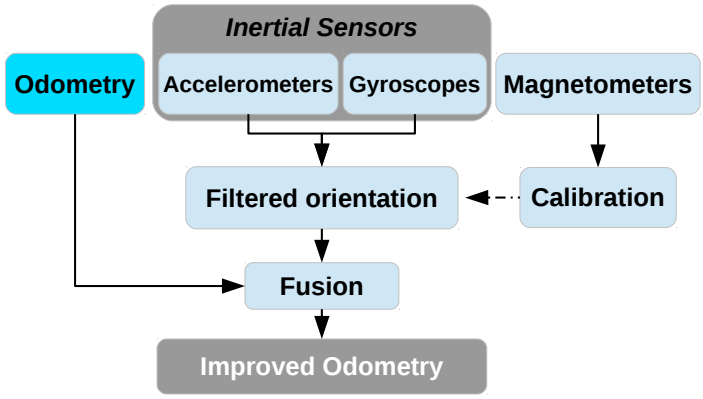


Figure 4.1: Odometry improvement diagram.

4.1.1 IMU Calibration

As mentioned before, inertial navigation suffers the accumulation of errors, so it is important to have an error model to know how it

evolves over time and correct it. The calibration errors belong to the deterministic part. This term collectively refers to errors inherent in scale factors, alignments and linearity of inertial sensors that tend to produce bias errors that are only observed when the sensor is measuring. Such errors lead to the accumulation of additional drift in the integrated signal, whose magnitude is proportional to velocity. In general, it is possible to measure and correct these errors [Woodman, 2007].

There are different methods of calibration: absolute, by comparison, by reciprocity or by inclination for static measurements. Standard methods, usually use gravity as reference for accelerometers and the projection of the speed of earth's rotation with respect to the vertical of a given latitude in the gyroscopes. This type of calibration allows us to calculate the scale factor and bias but it is not possible to estimate the misaligned axes.

Some IMU devices apply internal corrections to calibration errors, the measurements are corrected internally by the sensor firmware by applying a KF. The use of these calibrated measures ensures greater accuracy. In our case, with a general purpose IMU, the accelerometers and gyroscopes are calibrated from factory, that is, the misalignment matrices and bias are fixed taking into account the imperfections of the hardware design. The calibrated data correspond to the following formula:

$$a_{cal} = Ra * ka * (a - ba) \tag{4.1}$$

where Ra is the misalignment matrix, ka is the alignment compensation and ba is the bias. In the gyroscope case, is similar:

$$\omega_{cal} = Rg * kg * (\omega - bg - Kga * a) \tag{4.2}$$

where Rg is the misalignment matrix, kg is the alignment compensation and bg is the bias. Kga is the influence of gravity on gyroscopes. Since this influence is assumed to be null, both equations (4.1 and 4.2) have the same format.

The only sensor that is advisable to calibrate before each use, is the magnetometer, since it depends on the environment. The calibration method consists on rotating the IMU around each of its

axes and it is recommended that there are no ferromagnetic objects near the calibration site so as not to influence the measure. At the end, the parameters Rm (magnetometer misalignment matrix), km (alignment compensation) and bm (magnetometer bias) are modified in Equation 4.3. Calibration solves problems caused by ferromagnetic materials existing in the surrounding environment.

$$H_{cal} = Rm * km * (H - bm) \tag{4.3}$$

It is also necessary to perform an extrinsic calibration of the IMU, determining the most suitable position where to install it on the robot. It is convenient to place the device in the centre of rotation of the robot. In this way, the disturbances introduced by the centripetal and centrifugal accelerations that would be measured if it were displaced with respect to this turning centre are avoided. In addition, it is necessary to locate it as close as possible to the mass centre in order to avoid the noise introduced by the oscillations of the structure that supports the IMU. But in that position there is the disadvantage that the wheels engines cause a magnetic field that interferes to the IMU measures. Therefore, the device is located in the centre of rotation of the mobile robot, raised above its mass centre to the point where the magnetic field formed by the motors has not influence on the measurement.

Once the device has been placed in the platform, in order to obtain the relative pose between the robot and the IMU. The displacement in X, Y, and Z axes is measured manually and the orientation offset in *pitch* and *roll* has been obtained measuring the IMU in a short period of time (without moving the robot), assuming that during this time the orientation obtained is enough accurate.

4.1.2 Orientation filtering

The IMU orientation can be obtained by integrating the rotational speeds provided by the gyroscopes. Due to the drift error from several sources (error in measurement, errors of precision, etc) it grows over time and it is necessary compensate the drift error. To handle this, different filters can be used, to which accelerometers and/or

magnetometers data are added. Some of the most commonly used systems are the complementary filters [Mahony et al., 2005], the Mahony filter [Mahony et al., 2008] or the Madgwick filter [Madgwick et al., 2010].

In this approach the Madgwick filter has been chosen, due to this filter uses quaternions representation avoiding the *gimbal-lock* effect. It is an optimized gradient descent filter that calculates orientation including compensation for magnetic distortion and bias compensation of the gyroscope. Therefore it has the advantage of compensating the gyroscope with the accelerometers and being able to obtain the orientation in *North East Down (NED)* coordinate system, where the X, Y and Z axes point towards *North*, *East* and *Down* respectively, in case of magnetometers.

4.1.3 Odometry and orientation fusion

Following the diagram of Figure 4.1, the next step in this proposal is to merge the IMU filtered orientation with the robot's odometry to improve it. For this purpose, an extension of the KF for nonlinear systems has been used, the EKF, assuming that all noise components have zero mean and follow a Gaussian probability distribution function. The current state of the filter can be calculated by Equation 4.4 where $\hat{x}_t$ is the current state estimate, x_{t-1} is the state at the previous time instant, u_{t-1} is the system input, w_{t-1} is the noise and f is a nonlinear state transition function.

$$\hat{x_t} = f(x_{t-1}, u_{t-1}) + w_{t-1} \tag{4.4}$$

The Equation 4.5 represents the observation model $\hat{o_t}$, where h is the sensor model and v_t is the measurement noise.

$$\hat{o_t} = h(x_t) + v_t \tag{4.5}$$

Assuming a differential robot that moves in a plane environment, its state at any moment can be represented with the following state vector (Equation 4.6), where x_t^{rob} and y_t^{rob} are the robot coordinates in the world, θ_t^{rob} is its orientation (yaw), v_t^{rob} its linear velocity and ω_t^{rob} its angular velocity for a given time t.

$$\hat{x_t} = [x_t^{rob} \ y_t^{rob} \ \theta_t^{rob} \ v_t^{rob} \ \omega_t^{rob}]^T \tag{4.6}$$

The robot kinematics are given by Equation 4.9,

$$\Delta d = v_{t-1}^{rob} * \Delta t \tag{4.7}$$

$$\Delta \theta = \omega_{t-1}^{rob} * \Delta t \tag{4.8}$$

$$\begin{bmatrix} x_t^{rob} \\ y_t^{rob} \\ \theta_t{}^{rob} \end{bmatrix} = \begin{bmatrix} x_{t-1}^{rob} + \Delta d * cos(\theta_{t-1}^{rob}) \\ y_{t-1}^{rob} + \Delta d * sin(\theta_{t-1}^{rob}) \\ \theta_{t-1}^{rob} + \Delta \theta \end{bmatrix} \tag{4.9}$$

where Δd is the distance increment (Equation 4.7) and $\Delta \theta$ is the angle increment (Equation 4.8). This equation is nonlinear (since it is affected by the sine and cosine of the robot orientation) thus preventing a KF from functioning correctly and requiring the implementation of an EKF.

To linearise the system the Jacobian of the state vector is proposed, which is represented by the transition matrix F (Equation 4.10).

$$F = \begin{bmatrix} 1 & 0 & -\Delta d * sin(\theta_{t-1}^{rob}) & \Delta t * cos(\theta_{t-1}^{rob}) & 0 \\ 0 & 1 & \Delta d * cos(\theta_{t-1}^{rob}) & \Delta t * \sin(\theta_{t-1}^{rob}) & 0 \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

In this proposal the EKF implementation available in the ROS within the *robot localization* package [Moore and Stouch, 2016] has been used. This implementation uses as function h an identity matrix in such a way that it allows to realize partial updates of the filter based on the data from different sensors.

Although this filter implementation is prepared to represent the position in a three-dimensional space, in this case only the variables to represent the robot in the two dimensional space are used as indicated in Equation 4.6 neglecting the variables that measure the three

dimensional position (*Cartesian axis* z^{rob}, θ^{rob}_{roll} and θ^{rob}_{pitch} orientations) to avoid introducing errors in the filter. It also has two entries: the robot's odometry and the data obtained from the IMU.

Since the same data types of each sensor are not available, it was decided to measure the linear and angular velocities (v^{rob} and ω^{rob}) provided by the odometry, while in the case of the IMU only its angular velocity (ω^{imu}). In this way, the filter is able to compensate the angular odometry error with the IMU data. Also, since no positions are inserted (this integration is done inside the filter), it is not necessary to align the initial measurements of each sensors due to the speed measurements do not depend on the previous measurements.

4.1.4 Odometry improvement Results

The experiments have been performed in an indoor environment around the second floor of the *Polytechnic School of University of Alcalá* using a *Pioneer 3-AT* robot platform, in order to improve its odometry and the both IMUs introduced previously.

After calibration, the system has been evaluated in two scenarios with velocities limited to 0.5 m/s and 0.3 rad/s and remotely controlled.

- **Short path travel**: around elevators (with some influences in the magnetometers measurements) with several turns in a *20m x 15m* surface and a travel distance of 78 meters approximately.

- **Long path travel**: in a square of *70 m x 70 m* approximately, with a travel distance around 260 m. In this case the robot pass through several metallic doors that can affect to the magnetometer measurements as it has been mentioned before.

To obtain a quality measure of the approaches, the widely used *Adaptive Monte Carlo Localization (AMCL)* localization system has been used as a Ground Truth. This system takes the measures from the onboard robot LIDAR, the CAD map of the building and the initial pose over the map. In order to compare the different combinations of filters and IMUs it is necessary that all of the experiments are

in the same coordinate system. To manage this, all the travels starts with a straight path of 5 seconds, allowing translate and rotate the output position data by the EKF to the same initial position than the AMCL.

At each scenario, the results are compared from both IMUs and with the two different filters algorithms:

- *Od + IMU + EKF:* Fusing the odometry data and all the sensors from the IMU (accelerometers, gyroscopes and magnetometers) using the EKF.
- *Od + IMU (Mad) + EKF:* Fusing the odometry data with the Madgwick filter output (this filter only use accelerometers and gyroscopes from the IMU) using the EKF.

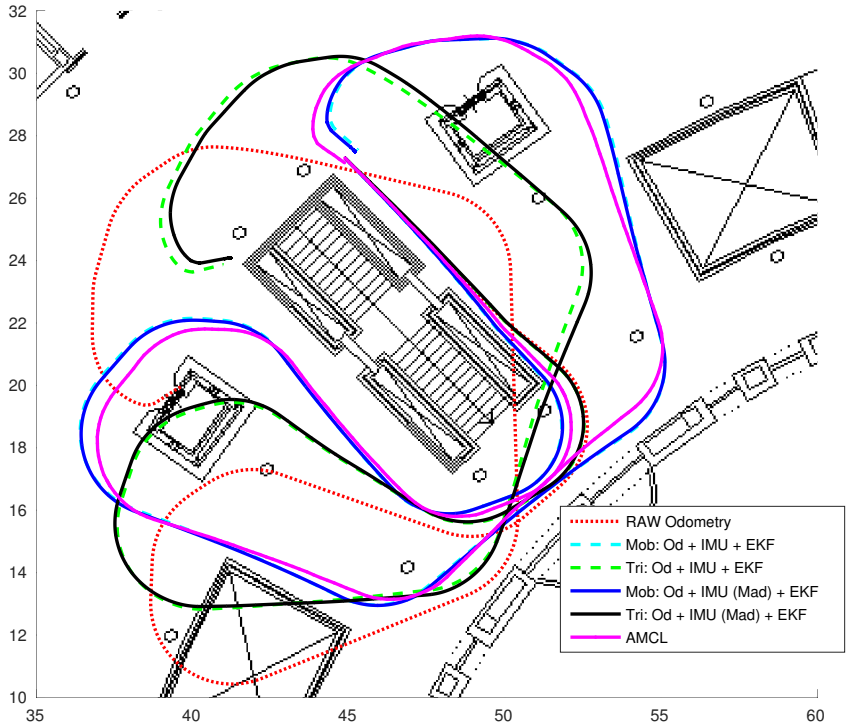


Figure 4.2: Short travel results

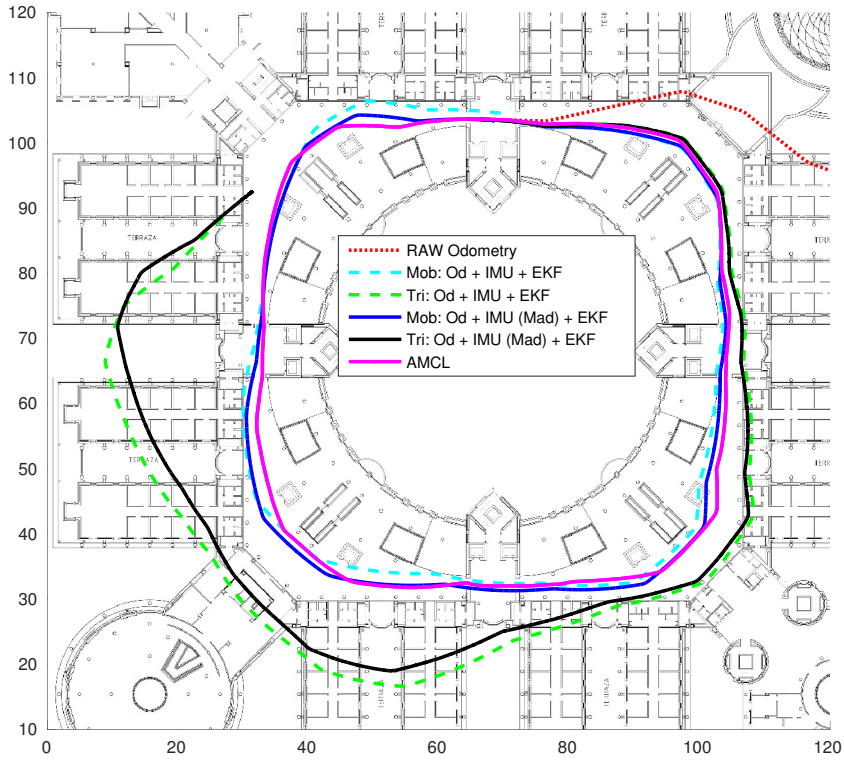


Figure 4.3: Long travel results

Figures 4.2 and 4.3 show the results obtained in both travels. The numerical results are shown in Tables 4.1 and 4.2. The error between the final position of the different approaches and the obtained by the AMCL, are shown in euclidean distance (meters), and orientation (degrees) . Also, the average error with respect to the AMCL is shown during the entire travel, since a better final position does not always means a correct path.

Based on the results, it is proved than the best location is obtained using the IMU of the mobile device, which in principle was not intended for this use. Although, its disadvantage with respect to the *Trivisio* is that it is necessary to have a WiFi network in the environment (if possible dedicated) to communicate the mobile software with the ROS development platform. It is also demonstrated that the inclusion of the magnetometers in the system significantly worsen the

Table 4.1: Results: Errors in short travel

Algorithm	**Dist.(m)**	**Ang.(°)**	**Avg.(m)**
RAW Odometry	9.01	68	5.22
Mob: Od+IMU+EKF	0.49	-7.08	0.611
Tri: Od+IMU+EKF	5.18	52.42	3.06
Mob: Od+IMU(Mad)+EKF	**0.37**	**-7.02**	**0.54**
Tri: Od+IMU(Mad)+EKF	4.86	48.31	2.85

Table 4.2: Results: Errors in long travel

Algorithm	**Dist.(m)**	**Ang.(°)**	**Avg.(m)**
RAW Odometry	222.26	-90.0	111.08
Mob: Od+IMU+EKF	4.87	-8.16	3.95
Tri: Od+IMU+EKF	40.61	43.97	16.08
Mob: Od+IMU(Mad)+EKF	**3.26**	**-4.66**	**2.39**
Tri: Od+IMU(Mad)+EKF	35.61	38.08	13.85

location. Also, it can completely destabilize the filter given the variations of the magnetic field that can be suffered, especially in indoor environments due to the machinery (such as the elevators engines), metallic doors, etc. Therefore, using only inertial sensors together with odometry, a enough precise location system to use by indoor mobile robots is achieved.

A proposal of inertial fusion with odometry of a real mobile robot has been implemented, which includes the calibration of the IMU. Real results are obtained which reduce a maximum distance error from 222m to 3.2m, and the angular error from 90º to 4.6º. In addition, two IMUs, a Trivisio Colibri and one available in a mobile device, have been compared using both heterogeneous devices on the same development platform (ROS).

4.2 Laser Pose Calibration

Laser scanners and range sensors are widely used in mobile robotics applications such as obstacle avoidance, object tracking, map build-

ing, feature extraction or self-localization.

The laser pose is usually not exactly known, for this reason it is necessary a calibration method to improve the accuracy of measurements. The different strategies that exist in the literature could be classified in two categories: some of them require a known movement of the laser and others need additional hardware, such as cameras. In order to simplify this process, and based on the work by [Antone and Friedman, 2007], a calibration method is proposed in our work [Llamazares et al., 2012].

First, the Euclidean transformation (R, T) between the laser system coordinates and the world frame is needed. Where T is the sensor's pose translation matrix and R the rotation matrix from the world frame. The transformation between a reference point M and a laser point Q, is defined by Equation 4.11, and the inverse transformation defined by Equation 4.12:

$$Q = R^T(M - T) \tag{4.11}$$

$$M = RQ + T \tag{4.12}$$

The measures of the laser are contained in a plane. Each ray starts in the origin of the sensor coordinates with an angle α_k and the measure represents a distance d_k along that ray. The x^{laser} axis of the sensor is coincident with the ray at $\alpha = 0$, and the y^{laser} axis with the ray at $\alpha = \pi/2$. Then, the measures that produce the scan are:

$$Q_k^T = (u_k \quad v_k \quad 0) = d_k(\cos\alpha_k \quad \sin\alpha_k \quad 0) \tag{4.13}$$

Once the relationship between coordinates systems is known, it is necessary to find a recognizable pattern with a single laser scan, therefore, it is important that the pattern has singular features. A rectangular prism (Figure 4.4) has 3 edges easily detectable, for this reason, the prism has been chosen. After choosing the best pattern, its position should be a priori known. To solve this, it is fixed to the robot with a metallic structure, carefully manufactured.

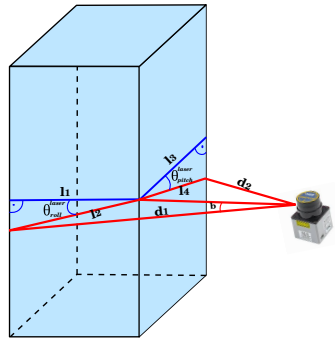


Figure 4.4: Laser Pose Calibration Pattern.

Then, the three edges of the prism are detected. After that, the measured length of each side of the prism is obtained. Finally, comparing the measures with the real lengths, it is possible to calculate the θ_{pitch}^{laser} (Equation 4.14) and θ_{roll}^{laser} (Equation 4.15) of the sensor.

$$\theta_{pitch}^{laser} = \cos^{-1} \frac{l_3}{l_4} \tag{4.14}$$

$$\theta_{roll}^{laser} = \cos^{-1} \frac{l_1}{l_2} \tag{4.15}$$

The three points that are detected form a plane where the laser is placed. Then, it is possible to obtain the coordinates x^{laser} and y^{laser} projecting the detected point to the floor from the known position of the pattern.

Finally, we calculate the z^{laser} coordinate using the distance measured by the laser beam with angle $\alpha = 0$ (Equation 4.16) to the floor (without the pattern).

$$z^{laser} = d^{\alpha=0} \cdot \cos(\theta_{pitch}^{laser}) \tag{4.16}$$

4.2.1 Laser Pose Calibration Results

In order to show the results of the laser pose calibration, the whole system was mounted with the laser angled to the floor. The calibration method starts with the *raw data* points obtained from the laser,

represented in *blue*. Then, with the transformations mentioned before, the points are projected to the floor (*processed data*), that are represented in *red*. The Ground Truth (*black line*) is the known position of the prism. Finally, the *processed data* and the Ground Truth are compared, obtaining a mean quadratic error of 4.8 millimetres. The calibration results are shown in Figure 4.5.

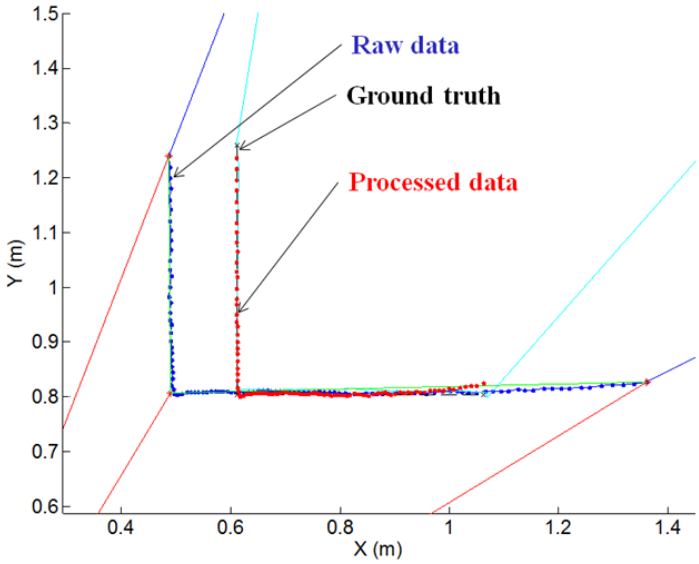


Figure 4.5: Calibration results.

4.3 3D Map Building using a 2D Laser Scanner

A 2D laser scanner provides distances and angles to the surrounding objects by scanning the environment in a plane, usually parallel to the ground. This technique is not enough to detect obstacles like stairs, a land irregularity or floor level variations. These kinds of problems are overcome with the 3D laser scanners. With these sensors, all features are directly extracted in three dimension point cloud. In the other hand, their price is much more expensive than the first ones.

Another solution for these problems is a combination of a 2D laser and a pan-tilt unit but, on the contrary, this system requires to spend more time to move the pan-tilt unit [Iocchi and Pellegrini, 2007].

There is a cheaper solution, presented in our work [Llamazares et al., 2012], by mean of using 2D laser angled down toward the ground in front of the robot, in a well known pose as it has been proposed in the previous section, combined with the robot's action model to extract the 3D features from movement [Singh et al., 1991]. While the sensor is scanning, the mobile robot is moving and then the system assembles each slice into a 3D points cloud. The main problem of this cloud is that it can be affected by the dead reckoning errors [Zhou and Chirikjian, 2003], then it is so important to reduce odometry error as we have done in Section 4.1. Scan matching techniques try to solve these problems [Thrun et al., 2000].

Once the laser pose has been calibrated (Section 4.2) in a known position and angled towards ground, we can assume that it forms a tetrahedron like shown in Figure 4.6. Also, an example of a feature p measured and the similarity of triangles (*red* and *blue* ones) to obtain the 3D pose of the measure are shown in Figure 4.6.

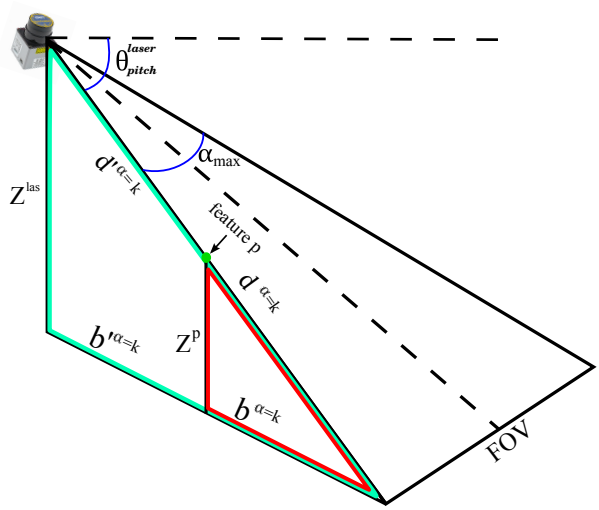


Figure 4.6: Example of tetrahedron formed by laser scanner and the ground.

Where z^{las} is the height of the laser, z^{p} is the height of the measured feature p, $d^{\alpha=k}$ is the distance from laser to ground in an angle $\alpha = k$, $d'^{\alpha=k}$ is the distance measured from laser to the feature, $b^{\alpha=k}$ and $b'^{\alpha=k}$ are the projections of the previous distances, α_{max} is the angle range of measure, θ^{las}_{pitch} is the pitch angle for the radial axis ($\alpha = 0$) and the *Field Of View (FOV)* over the ground.

Taking into account the calibrated pose of the laser over the robot, the measured distance in $\alpha = 0$ shall be a constant ($d^{\alpha=0}$) and it can be obtained by Equation 4.17. Using this constant, the *FOV* can be obtained by Equation 4.18. Concluding that a higher z^{las} or α_{max} and smaller θ^{las}_{pitch} can obtain a higher distance and FOV.

$$d^{\alpha=0} = \frac{z^{las}}{sin(\theta^{las}_{pitch})} \tag{4.17}$$

$$FOV = 2 \cdot d^{\alpha=0} \cdot tg\frac{\alpha_{max}}{2} \tag{4.18}$$

Using this information and the robot's action model, the map can be built assembling all the slices in a 3D points cloud. The method depends strongly on the odometry error. For that reason a improved odometry is necessary (See section 4.1). Also a scan matching technique is advisable to improve the results. In order to perform the scan-matching a method based on [Censi et al., 2005] has been proposed. This method is known like ICP. It is used to minimize the difference between two clouds of points. This method has been used to reconstruct the 3D surfaces from different scans. The algorithm is conceptually simple and is commonly used in real-time. It iteratively revises the transformation (*translation, rotation*) (Equation 4.19) needed to minimize the distance between the points of two raw scans.

$$\begin{bmatrix} cos(\theta^{las}_{yaw})cos(\theta^{las}_{pitch}) & R_{12} & R_{13} & 0 \\ -sin(\theta^{las}_{yaw}) & cos(\theta^{las}_{yaw})cos(\theta^{las}_{roll}) & cos(\theta^{las}_{yaw})sin(\theta^{las}_{roll}) & 0 \\ cos(\theta^{las}_{yaw})sin(\theta^{las}_{pitch}) & R_{32} & R_{33} & 0 \\ x^{las} & y^{las} & z^{las} & 1 \end{bmatrix} \tag{4.19}$$

where,

$$
\begin{aligned}
R_{12} &= sin(\theta^{las}_{yaw})cos(\theta^{las}_{pitch})cos(\theta^{las}_{roll}) + sin(\theta^{las}_{pitch})sin(\theta^{las}_{roll}) \\
R_{13} &= sin(\theta^{las}_{yaw})sin(\theta^{las}_{pitch})sin(\theta^{las}_{roll}) - sin(\theta^{las}_{pitch})cos(\theta^{las}_{roll}) \\
R_{32} &= sin(\theta^{las}_{yaw})sin(\theta^{las}_{pitch})cos(\theta^{las}_{roll}) - cos(\theta^{las}_{pitch})sin(\theta^{las}_{roll})
\end{aligned}
\tag{4.20}
$$

4.3.1 3D Map Building Results

The first Test Bed environment was established in the surroundings of the *Polytechnic School of the University of Alcala*. It has a surface of 60m x 40m. The robot used in the experimentation was a Seekur Jr. by Mobilerobots, with the following configuration: MacBook Pro, angled Hokuyo URG-04LX laser, bumpers and encoders in all wheels. They are shown in Figure 4.7.

Figure 4.7: Test Bed and real prototype used in the experimentation.

Figure 4.8 shows an example of 3D reconstruction of the Test Bed with one trail of the robot. Color is clearer when features are closer to the ground. It has been extracted a reconstruction detail of a bench.

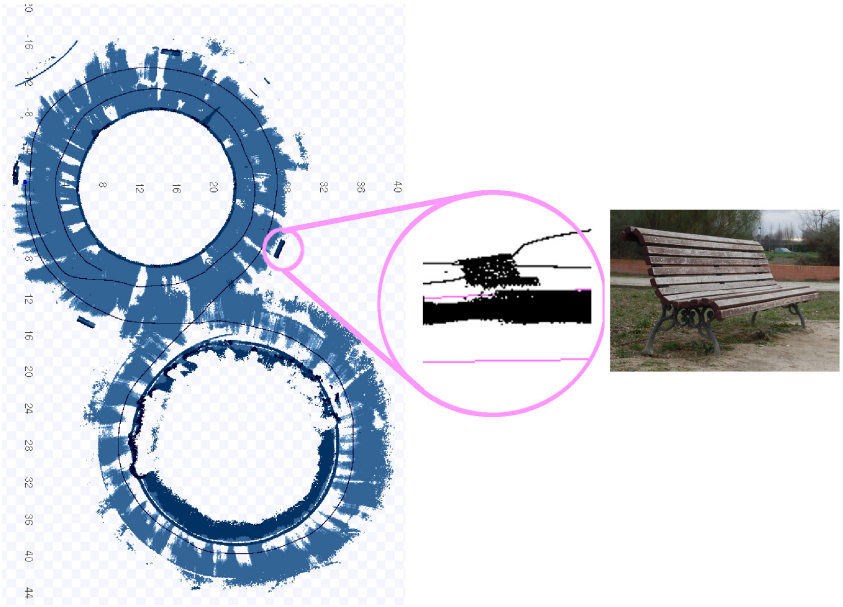


Figure 4.8: 3D map building results.

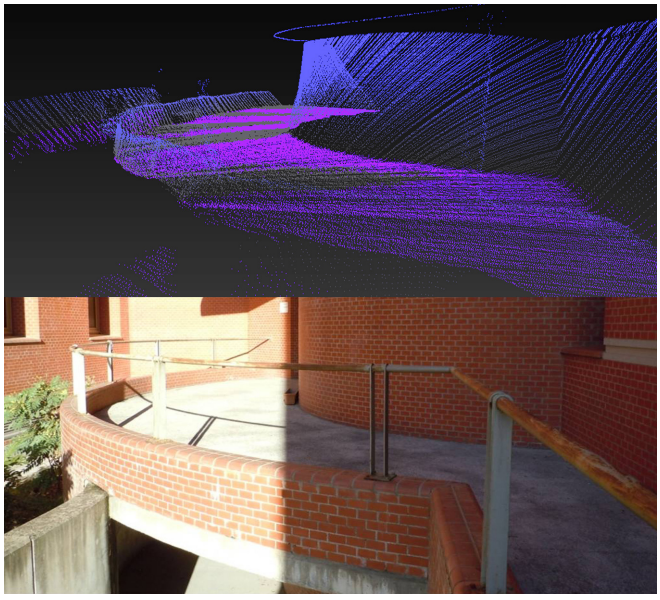


Figure 4.9: 3D School entry.

The second Test Bed was the *Polytechnic School* entry. Figure 4.9, in the bottom shows the real image of the stairs and in the top shows the 3D reconstruction of this environment. In the top image it is possible to identify the wall curvature, the back wall or the curb below the steel barrier rail.

The third Test Bed was an indoor stairs that is shown in the bottom of Figure 4.10. The top of Figure 4.10 shows a reconstruction of the environment where it is possible to recognize the stairs in both, upper and lower angles as well as certain details of the wall.

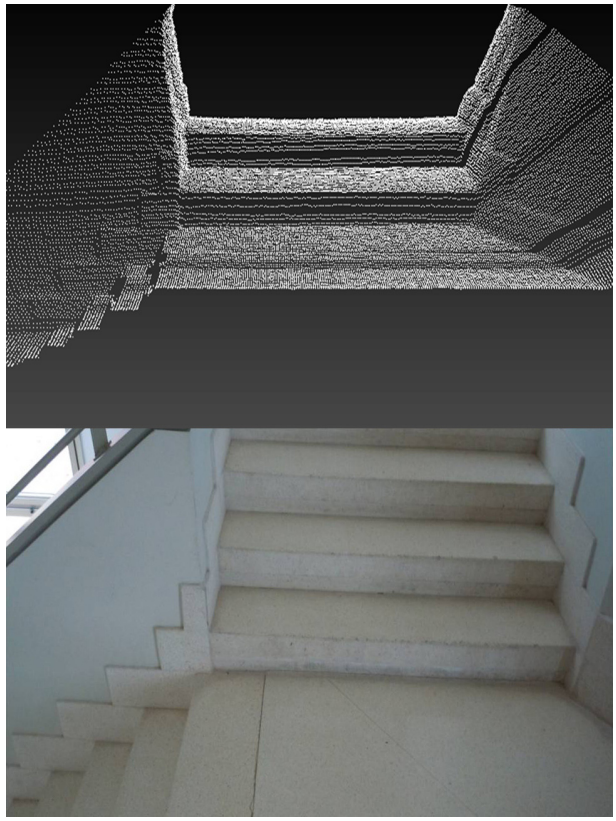


Figure 4.10: Stairs reconstruction.

All of these scenarios are especially relevant when we wan to collaborate in an effective human-robot teams, due to the limited capabilities of the wheeled robots ir order to overcome certain structure

barriers, such as stairs (Figure 4.10) or terrains with big irregularities (shown in Figure 4.8). On the other hand, the maximum detection distance has been limited due to *pitch* the laser to the ground. In order to avoid this reduction of the distance range, one approach would be combine some LIDAR parallel to the ground at different interesting heights where is desirable to detect more far away objects and a LIDAR angled to the ground, keeping the 3D map building capability exposed in this section.

4.4 DOMap: Dynamic Obstacles Map

In this Section, the challenge of implementing a DATMO system in order to achieve a robust navigation of UGV in terms of safety and energy saving, will be faced. This approach has been denoted Dynamic Obstacles Map (DOMap) and it has been published in our previous work [Llamazares et al., 2013].

As explained before, the classical mapping methods in the literature are usually designed to deal with static obstacles only. Also, some Object Based approaches can deal with some dynamic objects if they know the kind of obstacles a priori. In order to generalize the system, and increase the number of different objects that can detect, a *grid based* approach will be devised.

In this approach, due to the distance obtained by the 3D map proposed in Section 4.3 is not enough to detect dynamic obstacles and in order to keep the maximum distance range capabilities of our approach, the system take into account the distance measures provided by a LIDAR located parallel to the ground at the most adequate height based on the scenario.

4.4.1 Probabilistic Model of the Dynamic Environment

The *Grid Based* methods, as mentioned before, improve the perception stage due to be able to deal with the noise in the sensor data. For that reason, the BOF approach has been implemented as first step in the representation of the environment. The 2D Euclidean space is

discretized in a finite number of cells, each represents a position in the plane.

This method can be implemented as a loop of *prediction* and *estimation* steps. The authors of [Coué et al., 2006] suggest that *prediction* (Equation 4.21) and the *estimation* (Equation 4.22) steps can be computed as follows:

$$P(e_n^t|n^t, u^{t-1}) \propto \sum_{n^{t-1}} P(n^t|n^{t-1}, u^{t-1}) P(e_n^{t-1}|n^{t-1}) \tag{4.21}$$

$$P(e_n^t|z^t, n^t) \propto \sum_{m=1}^{S} (\prod_{s=1}^{S} P(o_s^t|e_n^t, m)) \tag{4.22}$$

where e_n^t is the occupancy of the cell n at time t, u^{t-1} is the command issued at time $t-1$, o^t are the observations in t and m is matching between a cell and an observation. $P(n^t|n^{t-1}, u^{t-1})$ is then the transition probability defined by the vehicle dynamics. The standard BOF framework has several issues with the velocity estimation. Firstly, this framework assumes that the velocity of each grid cell is constant [Yguel et al., 2006]. Secondly, the discretisation has to be performed also in the velocity space, meaning that a separate estimate for each pair of velocities (v_x, v_y) is required. This discretisation for a large range of possible velocities together with calculations of static objects result in high computational costs. For these reasons, other authors proposed object detection and clustering techniques to obtain the objects' velocities [Mekhnacha et al., 2008] with an additional constraint that the position of the obstacle has to remain within a bounded neighbourhood.

The BOF approach has been improved in our proposal in terms of efficiency, by adding a stage to detect the relative velocities of the obstacles without assuming discretised velocities. Figure 4.11 shows the flow diagram of the *Dynamic Obstacles Map (DOMap)* proposed model. The probabilistic model of the environment with velocities likelihood is obtained as follow:

- **From laser scan to laser grid**: in order to obtain a frame with the form of a *zenith image* of the environment from the

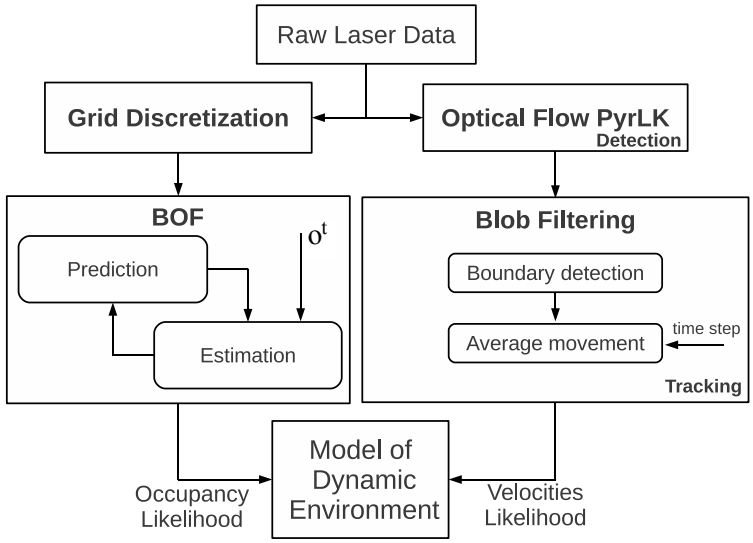
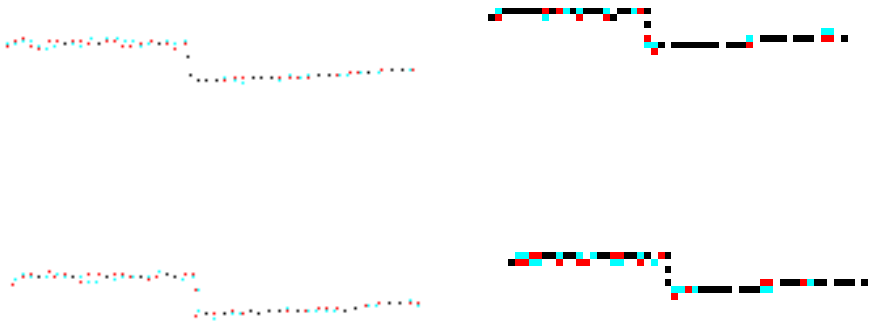


Figure 4.11: DOMap diagram. Probabilistic model for dynamic environment.

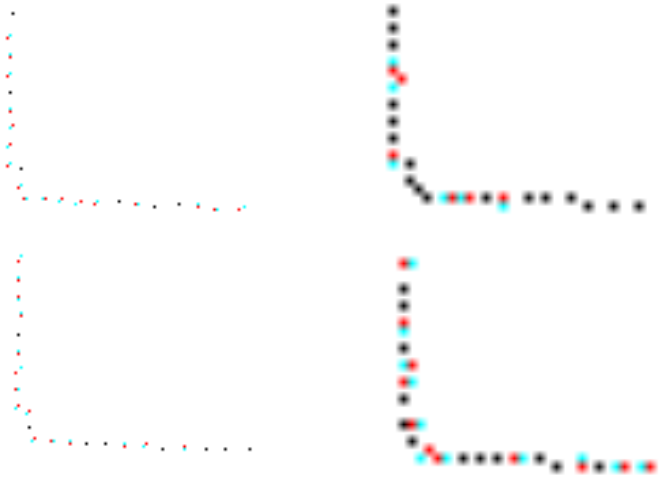
laser raw data, the 2D euclidean space is divided in a grid (called *laser grid*). The laser measures are transferred to this grid. The cells of that grid are smaller than in case of the occupancy obtained by BOF in order to avoid a lost of precision. However, treating the laser measures as a grid allows us to deal with the systematic errors that the LIDAR sensors return in their measures.

Figures 4.12 and 4.13 show real LIDAR measurements of several walls with the robot stopped in two consecutive time steps, where the *red dots* are the LIDAR impacts at the current step, the *cyan dots* are the LIDAR impacts at the previous step, and the *black dots* represent the current impacts match position with the previous ones. Figures 4.12(a) and 4.13(a) show the LIDAR raw measurements oscillations from measurements of a static object in an static environment between time steps. On the other hand, Figures 4.12(b) and 4.13(b) show the same environment with a laser grid of 3cm, where there are much more *black dots*, this means the measurements coincide along



(a) LIDAR raw data. (b) LIDAR data filter in a 3cm grid.

Figure 4.12: LIDAR raw data and filtered in the laser grid.



(a) LIDAR raw data. (b) LIDAR data filter in a 3cm grid.

Figure 4.13: LIDAR raw data and filtered from real experiments.

the time, allowing to filter the systematic errors of the LIDAR measurements. The selection of the laser grid size must be

based on data sheet of the sensor used ($\pm 30mm$ in this case, due the measurements was taken with a *SICK LMS151*).

- **Movement Detection**: once the *laser grid* is obtained, it can be treated as an image and therefore we can use computational vision techniques. The movement of the laser cells between two consecutive frames is estimated by computing the pyramidal implementation of *Lukas Kanade Optical Flow algorithm* [Bouguet, 2000]. Optical Flow methods assume intensity constancy and small motions between consecutive frames. The partial derivatives with respect to the images coordinates (spatial and temporal), are used by Optical Flow methods, for that reason are also called differential methods. In that way, it is possible to find the corresponding points assuming brightness constancy, given by Equation 4.23:

$$I(u, v, t) = I(u + \Delta u, v + \Delta v, t + \Delta t) \tag{4.23}$$

Where $I(u, v, t)$ is the intensity of pixel (u, v) at time t. Assuming small motion and applying a first-order Taylor expansion, Equation 4.23 can be written as:

$$I(u, v, t) = I(u, v, t) + \frac{\partial I}{\partial u}\Delta u + \frac{\partial I}{\partial v}\Delta v + \frac{\partial I}{\partial t}\Delta t \tag{4.24}$$

Obtaining the constrained equation of Optical Flow, given by the Equation 4.25:

$$I_u \Delta u + I_u \Delta v + I_t \Delta t = 0 \tag{4.25}$$

The *Lukas-Kanade* algorithm use the least squares method and a small windows in the image to solve the Equation 4.25, given by Equation 4.26:

$$
\begin{aligned}
&min \sum_i (I_{ui}\Delta u + I_{vi}\Delta v + I_t)^2 \\
&\sum (I_{ui}\Delta u + I_{vi}\Delta v + I_{ti}) I_{ui} = 0 \\
&\sum (I_{ui}\Delta u + I_{vi}\Delta v + I_{ti}) I_{vi} = 0 \\
&\sum I_{ui}^2 \Delta u + \sum I_{ui} I_{vi} \Delta v = -\sum I_{ui} I_{ti} \\
&\sum I_{ui} I_{vi} \Delta u + \sum I_{vi}^2 \Delta v = -\sum I_{vi} I_{ti}
\end{aligned}
\tag{4.26}
$$

In order to avoid the restriction of small motions between time steps, one solution is to apply the Optical Flow in a coarse to fine method. This approach consists on building smaller grids from the original one in a pyramidal way (shown in Figure 4.14), where each cell in successive smaller grids represents a bigger distance that in the original one, allowing to detect larger motions. The output of the Optical Flow in the previous grid (*smaller one*) is used to guess where the motion will be in the next layer of the pyramid (*bigger grid*) in an iterative process, as shown in Figure 4.14.

Also the robot movement between two time steps is taking into account based on the improved odometry explained in Section 4.1. Although the algorithm is made for using in images, it can be applied to the *laser grid* previously generated. The little amount of impacts of each obstacles and the lack of features of the laser impacts make difficult than the Optical Flow returns a lot of matching points as shown in Figure 4.15.

Figure 4.15 shows some examples of movement detections of four persons walking in different directions around to the robot, where the *red dots* are the LIDAR impacts at the current step, the *cyan dots* are the LIDAR impacts at the previous step, and the *green lines* represent the correspondence between points

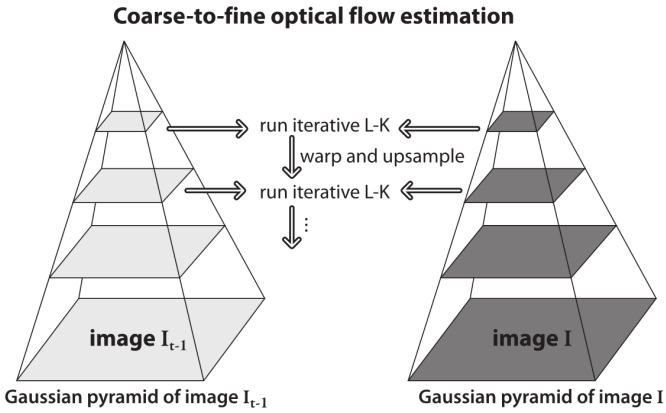
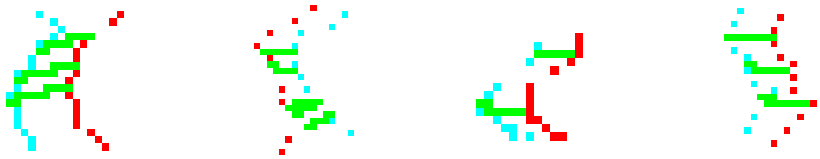


Figure 4.14: Pyramidal Lukas-Kanade Optical Flow [Bradski and Kaehler, 2013].



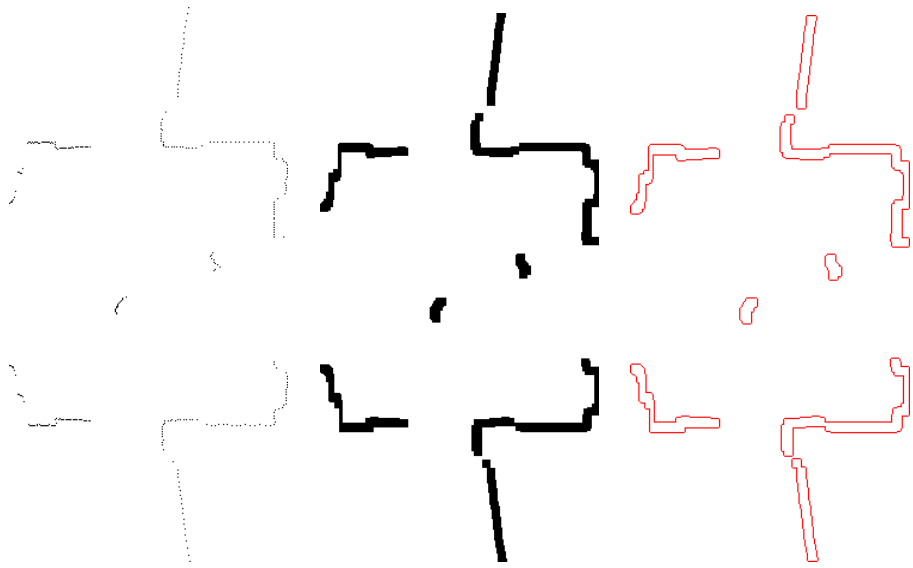
(a) Person 1 (b) Person 2 (c) Person 3 (d) Person 4

Figure 4.15: Optical Flow movement detections in real experiments

obtained by Optical Flow. As it can be seen, only two or three correspondences are detected for each person.

- **Blob Filtering**: in order to filter the output of the Optical Flow, a blob filtering stage is performed in three steps: firstly, an *opening* operation over the laser grid is performed to solve some discontinuities of the object in the laser grid (Figure 4.16(b)), due to the small amount of impacts that the objects receive (Figure 4.16(a)). This amount of impacts depends on the size, the distant and the orientation of the object respect to the LIDAR; secondly, a segmentation of the objects, based on the detected boundaries using the contours detector

by [Suzuki and Abe, 1985] is performed (Figure 4.16(c)); finally, the average motion of all the laser cells inside each boundary is estimated in order to associate the average velocity to the object.



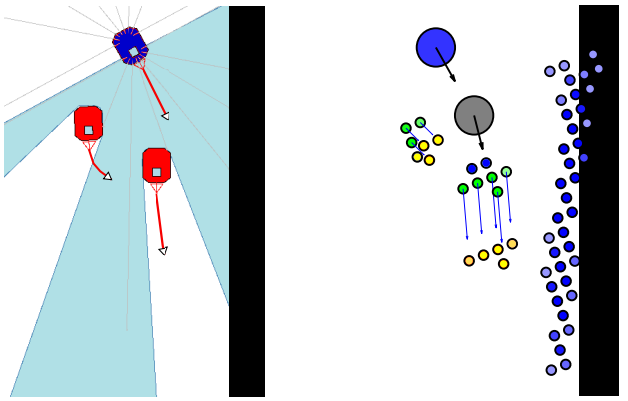
(a) LIDAR grid. (b) Opening. (c) Contours detected.

Figure 4.16: Blob Segmentation.

- **Compute the velocities**: the average velocity value and the time step are used to compute the relative velocities between the obstacles with respect to the robot's reference local frame (along local axis X and Y), while the velocity along the Z axis is not taken into account due to the assumption that our wheeled robots move only on a flat world.

- **Transfer the velocities to the occupancy grid**: Each occupancy cell has to exceed certain occupancy threshold to compute it in order to reduce the noise in the Optical Flow and the computational cost. Once this threshold is exceeded, the velocities from each blob are transferred to the occupancy cells that are occupied by the blob.

The result of the perception stage is a dynamic occupancy grid providing an estimation of velocity and the occupation probability.



(a) Simulated environment.

(b) Sensing results.

Figure 4.17: Simulated environment and sensing results.

The output of the probabilistic model is illustrated in Figure 4.17. Left image shows the simulated environment. The robot is represented in *blue* colour and the moving obstacles are *red*. Right image shows the sensing results. The *blue circle* is the robot at the current time step, the *grey circle* shows the predicted future position of the robot, moving obstacles at current time step are represented with *green dots*, the predicted future positions of moving obstacles are shown as *yellow dots* and static obstacles are shown as *blue dots*. The occupancy probability value is given by the *darkness* value of each dot, the darker the higher is the probability of being occupied.

4.4.2 Obstacle Avoidance Algorithms

In order to test the DOMap performance is appropriate apply the local mapping approach to different obstacle avoidance algorithm, and compare their performance between the two inputs provided to their perception stage: the raw LIDAR data and the DOMap proposed. The obstacle avoidance systems proposed to test our approach are some of the commonly used in robotics:

- **VFH+** [Ulrich and Borenstein, 1998]: is a reactive only algorithm. It was developed to use with range sensors and it divides the environment into angular sectors to choose the best sector to navigate according to a cost function. This function takes into account different parameters such as the angular difference from the actual position or distance to the goal.

- **CVM** [Fox et al., 1997]: it works with by adding physical constraints form the robot and the environment to a velocity space. The velocity space consists of translational and rotational velocities (v, ω). It assumes that the robot navigates describing arcs of curvature. This algorithm builds all the curvature arcs that the robot can follow, taking into account the robot dynamics, and selects the best arc to follow based on a cost function that depends on parameters such as free distance to obstacle or angular difference to the orientation of the robot. There are two extensions of this algorithm that add a previous stage where the environment is divided and select a middle goal based on another cost function that CVM should reach:

  LCM [Ko and Simmons, 1998]: it divides the environment into rectangular lanes.

  BCM [Fernández et al., 2004]: it divides the environment in angular sectors .

One of the main problems of these methods is that most of them do not take into account the dynamic information of the environment, they only consider that all obstacles are static. For that reason an approximation has been computed, changing the perception stage of the different obstacle avoidance algorithms to the map provided by DOMap perception. Instead of using the actual position of the obstacle we can use the prediction of the next position (given a certain time) if the obstacle is moving, so the algorithm can anticipate the avoidance manoeuvre.

4.4.3 Energy Consumption Model

In order to compare the proposed method with classical obstacle avoidance algorithms (Section 4.4.2) two parameters has been taking into account, the smoothness of the path and the energy consumption. In terms of energy, only the power demand of the robot's engines has been taking into account. Also, a constant energy consumption of the rest of the equipment has been assumed and therefore it cannot be improved any further. Assuming also that the power demands of the robot's engines are based on overcoming inertia, road grade, tyre friction and aerodynamic loss. This road-load methodology was mainly introduced by [Sovran and Bohn, 1981]. The power demand (in Watts) is the tractive power as defined by Equation 4.27:

$$P = mv[a(1+\varepsilon) + gR_G + gK_R] + \frac{1}{2}\rho K_D A_F v^3 \tag{4.27}$$

where m is vehicle mass in metric tones (0.077 in our case), v is vehicle speed (assuming no headwind) in m/s, a is vehicle acceleration in m/s^2, ε is a mass factor accounting for the rotational masses and assumed to be 0.1 [Jiménez-Palacios, 1999], g is acceleration due to gravity (9.8 m/s^2), R_G is road grade (0.0 in our case), K_R is rolling resistance, this value for radial tires can range from 0.008 to 0.013 for a majority of the on-road passenger car tires but can be larger depending on tire pressure, temperature, ground surface, and speed [Bosch, 2000, Gillespie, 1992] (a medium value in the range $\approx$ 0.009 is assumed [Sovran and Bohn, 1981]), ρ is air density ($\approx$ 1.2 kg/m^3), K_D is aerodynamic drag coefficient ($\approx$ 0.3 [Sovran and Bohn, 1981]) and A_F is the frontal area in square meters ($\approx$ 1 m^2 in our case). These values are obtained based on the references and the *Seekur Jr* specifications, assuming that the goal of our work is to obtain a comparison of energy saving, and not the exact value at each case.

The robot speed used to obtain the power demand is provided by the kinematic model of the robot based on the angular speed of the wheels for each time step (100 ms in our case). According to this, it is assumed that the robot is moving with constant linear speed between each execution step. On the other hand, the planning proposed and

almost all the classical algorithms do not allow the robot to describe sharp turns or spin.

4.4.4 Optimal path planning using Approximate Inference

In addition to test the DOMap as perception stage of the classical obstacle avoidance algorithms our proposal has been tested with the path optimisation problem within the *Approximate Inference Control (AICO)* framework [Toussaint, 2009, Rawlik et al., 2012]. This approach has been formulated by the *Institute of Perception, Action and Behaviour (IPAB)* taking into account all the information provided by DOMap (not only the prediction of the next obstacle's positions) and the energy consumption model explained in Section 4.4.3. The stochastic optimal control has been successfully used to solve optimisation problems in robotics [Nakanishi et al., 2011, Braun et al., 2012b, Braun et al., 2012a, Rawlik et al., 2010].

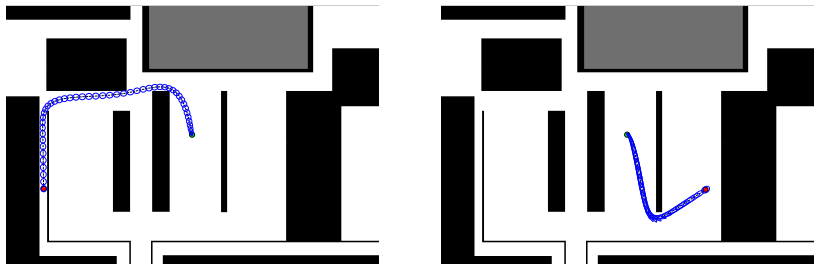


Figure 4.18: Two examples of optimal paths computed using AICO in a static environment (parking of the Polytechnic School).

In order to achieve robustness and safety of our vehicle, the method yields free paths that tend to maximise the clearance between the vehicle and the obstacles based on a *Voronoi graph*. The aim of the global planning is to keep the vehicle at a safe distance from the surrounding obstacles. The initial path is computed using graph search on the *Voronoi graph*. This path is then used as AICO initialisation and it helps to deal with local minima. Figure 4.18

shows two examples of optimised paths computed for static environment. For further details of AICO planning refer to our previous work [Llamazares et al., 2013].

4.4.5 Implementation and Results

In this section, the implementation of the system and the experimental results will be described. The results have been obtained with the Stage simulator in order to obtain comparable results and with *Seekur Jr* hardware platform, due to it is specially indicated to work in outdoors.

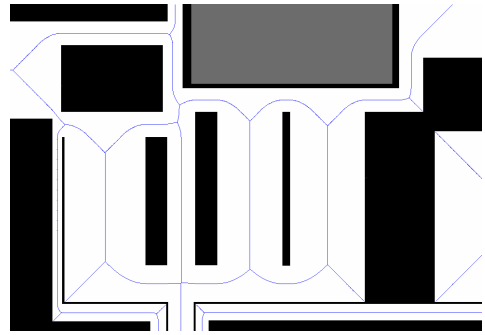
The results have been evaluated in two stages: firstly, the gain of using the probabilistic model of the environment DOMap inside the perception stage of the classical algorithms (VFH+, CVM, LCM and BCM) has been evaluated; secondly, the whole dynamic obstacle avoidance system (DOMap + AICO) with the classical algorithms has been evaluated.

4.4.5.1 Test Bed

The system has been tested in an outdoor environment in the South Parking of the *Polytechnic School* at the *University of Alcalá (UAH)*. The overall area of the environment is approximately 70x70 metres (Figure 4.19(a)). In addition, the surveying route has been marked in *red* colour. The route was 330m long, and the *blue rectangles* represents the scenarios where the system was tested.

The *Seekur Jr.* (Figure 4.20), was equipped with the following configuration: MacBook Pro with Ubuntu 12.04 LTS operating system, *Player/Stage* control software, RTK-GPS Maxor GGDT by JAVAD, an outdoor LIDAR SICK LMS 151, bumpers, encoders in the wheels and a Inertial Measurement Unit (IMU) to reduce the odometry errors in the turns.

(a) Surveying route and test scenarios in real environment.



(b) Voronoi Diagram.

Figure 4.19: Test environment.

Figure 4.20: Seekur Jr. used in the experimentation.

4.4.5.2 Results using the probabilistic model of the dynamic environment (DOMap)

In order to test the effect of using the proposed probabilistic model, the four classical algorithms on the real platform are going to be used, the system has been tested in two different scenarios and it has been

commanded to reach a goal 7 meters away from it start position:

1. **Parallel:** For this test, the robot is located in the middle of a 5 meters wide corridor (Figure 4.19(a): Scenario 1). The obstacle starts moving along the corridor in the same direction as the robot but with a delay and it tries to overtake the robot.
2. **Perpendicular:** For this test the robot is located at the crossroads (Figure 4.19(a): Scenario 2). The obstacle is moving along the main road perpendicular to the robot's direction, blocking its path.

For all of these experiments, the analysed parameters are as follow:

- *Path curvature:* The assumption is that the smoother the path, the lower the energy consumption.
- *Acceleration (a):* the positive acceleration in (m/s^2) ignoring energy regenerated from breaking.
- *Velocity (v):* the absolute velocity of the robot in (m/s).
- *Time (t):* the time needed to reach the goal in (s).
- *Energy (E):* the energy consumption of the robot in (J).

The top part of Figures 4.21 and 4.22 shows the path followed by the robot using the VFH+ algorithm with the raw laser as input (*top left*) and VFH+ with the proposed probabilistic model (*top right*) in the two scenarios. The *yellow diamond* is the target goal, the *blue square* is the dynamic obstacle that follows the path marked as a *dashed cyan line*.

The bottom part of Figures 4.21 and 4.22 shows the path followed by the robot using AICO with DOMap (*bottom left*) and the comparative of the path's curvatures in the three cases (*bottom right*).

Figure 4.21 shows that the path followed by the robot in the parallel scenario with VFH+ and DOMap is smoother than in the case of the VFH+ and raw laser data. Even smoother is the case of AICO with DOMap due to AICO takes into account the obstacle dynamics and computes that the obstacle does not interfere in the

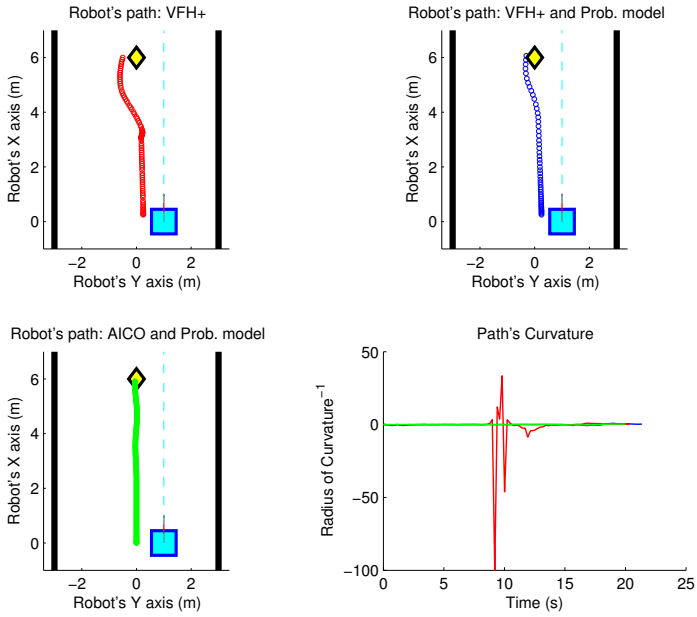


Figure 4.21: Path using VFH+ and AICO: parallel scenario.

robot's path, being almost a straight path, the optimal in term of energy.

Figure 4.22 shows smoother paths when DOMap is used, similar to parallel scenario (Figure 4.21). In this scenario has been avoided not desirable behaviours, such as the loop performed by the VFH+ with laser raw data (*in red*). Although in this scenario the path curvature is similar in the case of VFH+ with DOMap and the case of AICO with DOMap, is safer the path obtained by AICO due to the obstacle is avoided in the rear part of it, avoiding the risky situation of cross in front of it.

Figures 4.23 and 4.24 show the velocities and accelerations of the robot using the LCM with raw laser data (*in red*), LCM with DOMap (*in blue*) and AICO with DOMap (*in green*) in both scenarios. In the case of LCM with laser raw data (*red lines*), the movement has sudden changes in linear and angular speeds, performing erratic movements. Adding the DOMap to the perception stage, the LCM describes a smoother path, even more smooth and efficient (see Table 4.3) if

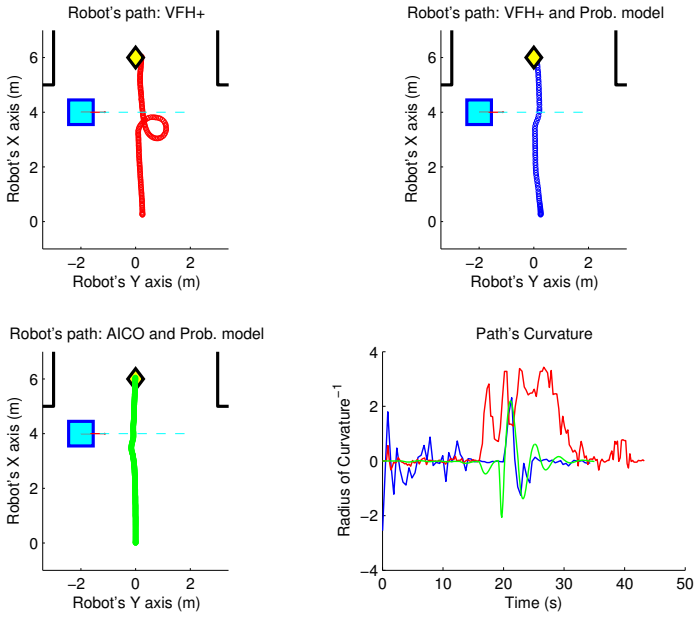


Figure 4.22: Path using VFH+ and AICO: perpendicular scenario.

AICO with DOMap is used.

Figure 4.25 shows the energy consumption of the robot using the VFH+ compared to using AICO algorithm in parallel (Figure4.25(a)) and perpendicular (Figure 4.25(b)) scenarios. The results show that using the probabilistic model DOMap, the energy consumption is reduced by 20% and using AICO, the energy consumption is reduced by 32%, compared with VFH+.

A summary of energy consumption by all the combinations of obstacle avoidance algorithms exposed in Section 4.4.2, with laser raw data and DOMap as perception stages, is shown in Table 4.3. Also the energy consumption of the approach AICO with DOMap is shown.

The results of the path optimization demonstrate that optimising the energy consumption further decrease the curvature of the trajectory (Figures 4.21 and 4.22), increase the path's safety and avoid not desirable manoeuvres. Similarly, Figures 4.23 and 4.24 show that the velocity and acceleration profiles are much smoother. As a result,

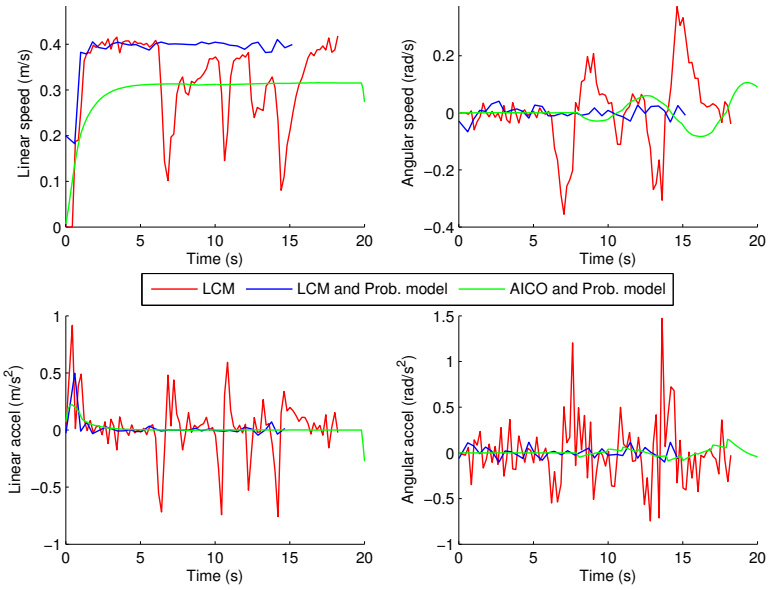


Figure 4.23: Vel. and acc. using LCM and AICO: parallel scenario.

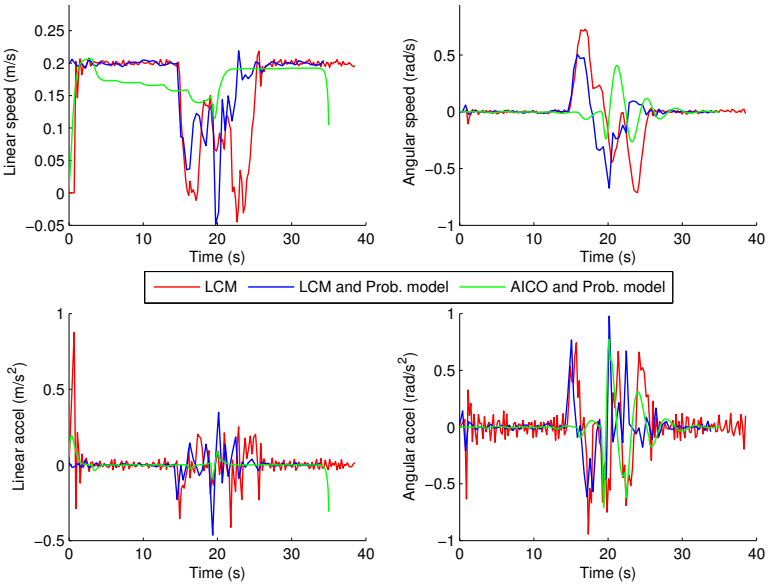
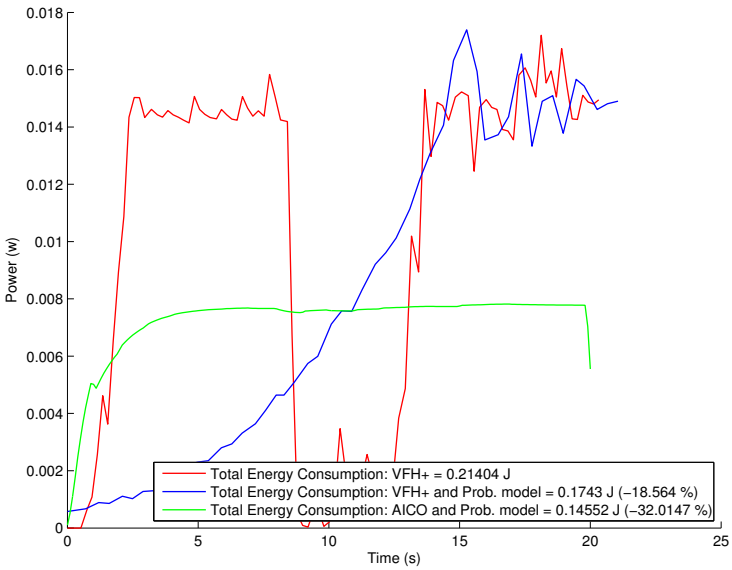
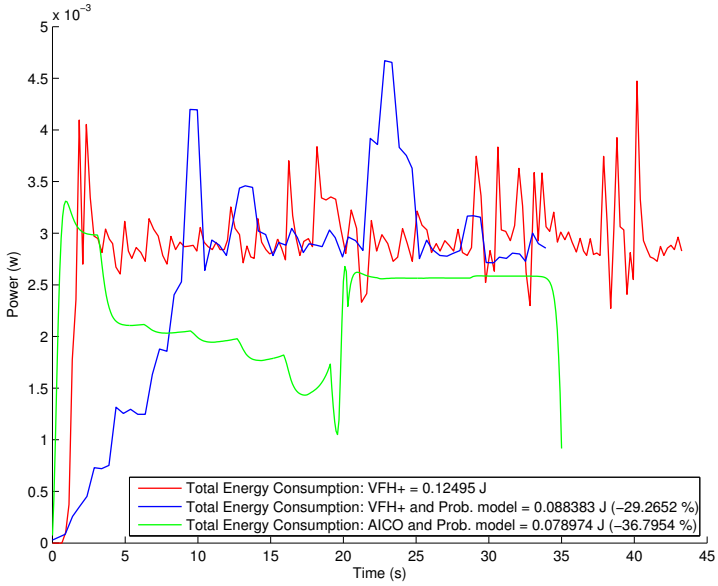


Figure 4.24: Vel. and acc. using LCM and AICO: perpendicular scenario.



(a) Parallel scenario.



(b) Perpendicular scenario.

Figure 4.25: Energy consumption using VFH+ algorithm.

the energy consumption in both scenarios was reduced from 10% and

Table 4.3: Summary of results for energy consumption. *AICO* is being compared with the algorithm with less consumption at each scenario.

	Scenario	Perception stage	Total Energy Consumption (J)	Reduction of Consumption (%)
VFH+	Parallel	Raw laser data	0.21404	-18.56%
		DOMap	**0.1743**	
	Perpendicular	Raw laser data	0.12495	-29.26%
		DOMap	**0.088383**	
CVM	Parallel	Raw laser data	0.22615	-7.52%
		DOMap	**0.20914**	
	Perpendicular	Raw laser data	0.10458	-4.39%
		DOMap	**0.099981**	
LCM	Parallel	Raw laser data	**0.20125**	+2.89%
		DOMap	0.20709	
	Perpendicular	Raw laser data	0.09097	-8.1295%
		DOMap	**0.083575**	
BCM	Parallel	Raw laser data	0.20546	-7.29%
		DOMap	**0.19047**	
	Perpendicular	Raw laser data	0.16599	**-45.5%**
		DOMap	**0.090454**	
AICO	Parallel	DOMap	**0.14552**	-16.51%
	Perpendicular	DOMap	**0.0789**	-10.91%

16% when compared with the best results achieved by the reactive methods (VFH+ with DOMap in the parallel and LCM with DOMap in the perpendicular scenarios respectively)up to 45.5%.

4.5 iDOMap: improved Dynamic Obstacle Map

In this section is presented an improvement of the probabilistic model of the environment explained in Section 4.4. In order to detect a wider range of obstacles. This approach has been denoted *improved*

Dynamic Obstacles Map (iDOMap), as it is shown in Figure 4.26, and it is going to take into account participants of unusual appearance, such as people wearing fully loose clothes, people pushing objects (such as, baby carriage, shopping cart, baggage or trolleys) or even small participants, such as kids or pets. This unlimited number of kinds of participants makes it impossible to model each of them, making necessary a *grid-based model-free* proposal. In the next sections we are going to explain the different parts of the DOMap that have been modified or added in order to improve the whole system.

4.5.1 Improved Movement Detection

In order to improve the movement detection, the accuracy and the richness of the data are crucial, for these reasons some aspects have been evaluated, such as, to increase the features of the LIDAR measurements or to erase false detection points.

4.5.1.1 Mixed pixels

This phenomenon occurs when the laser beam impacts in the edge of an object (foreground object) and behind it there is other object (background object). The measure returned by the LIDAR is a "mix" between the distance of foreground and background objects. For that reason, there are called *"mixed pixels"* or *"mixed points"* [Ye and Borenstein, 2002]. This effect also appears when the surface reflectivity of the objects sharply changes.

During previous real experiments this phenomenon was identified in different situations, usually when a dynamic obstacle goes near a wall. Figure 4.27 shows a sequence with a column (on the left), a corner wall and a person walking (on the bottom) in three time steps where *mixed points* (marked as *red dots*) appear: one between person and the column (Figure 4.27(a)) and other two between the person and the wall (Figures 4.27(b) and 4.27(c)).

On the other hand, in other scenes the *mixed points* can be confused with real measurements, as shown in Figure 4.28 where the *red dots* are *mixed points* and the *green dots* real measurements. In this

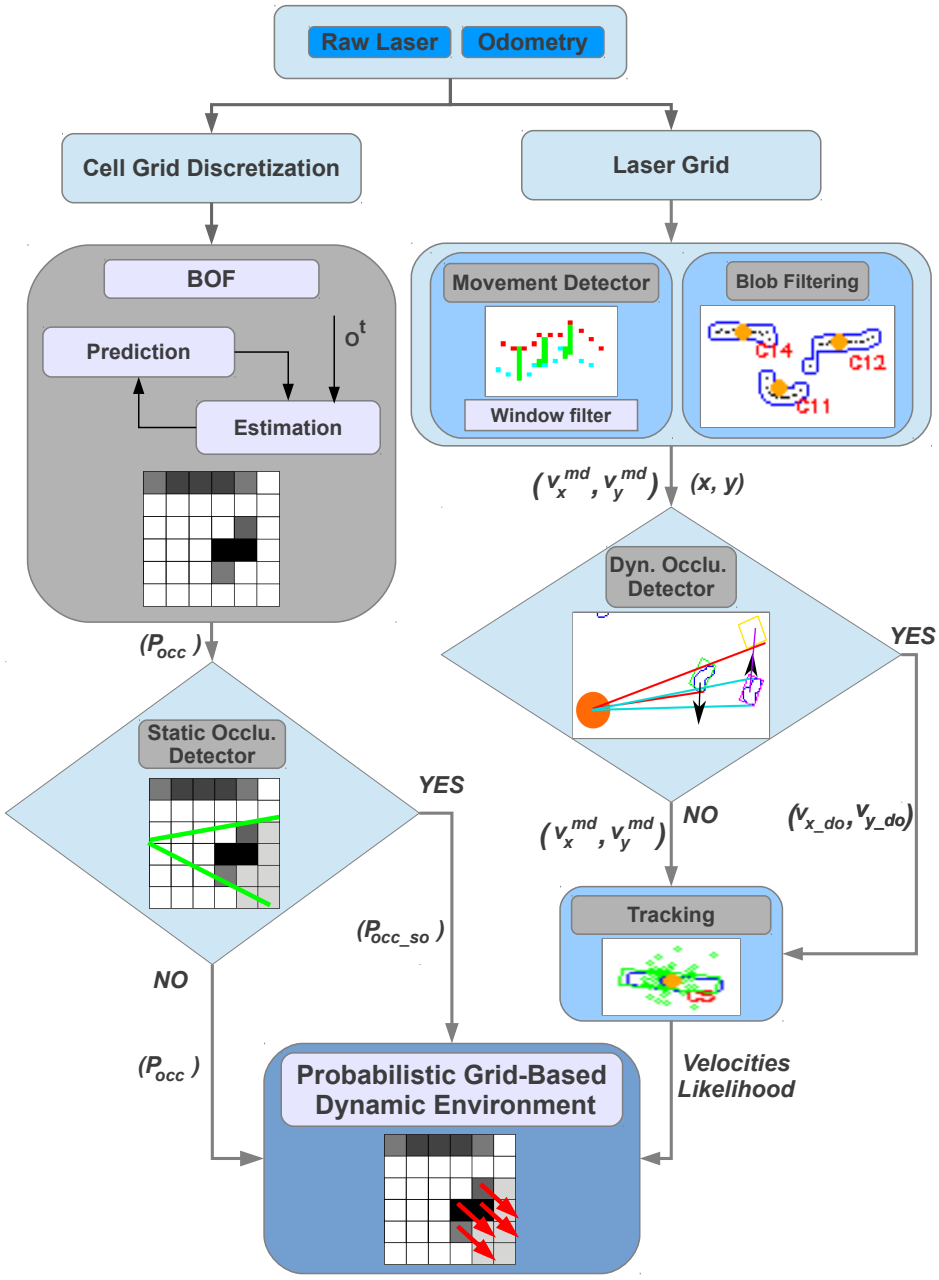
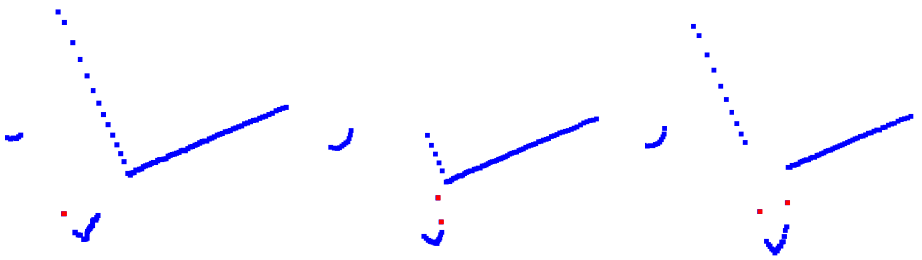


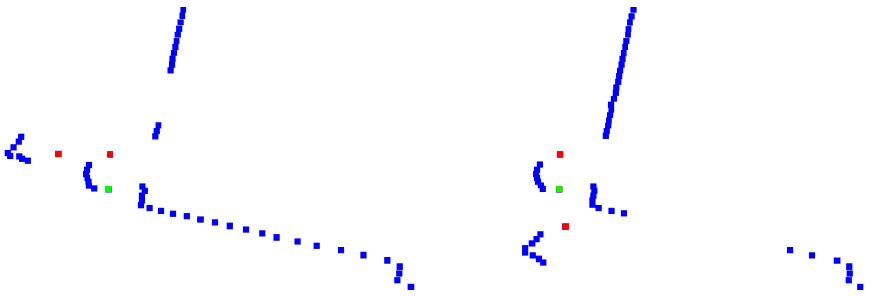
Figure 4.26: iDOMap Diagram.



(a) Step 1 (b) Step 2 (c) Step 3

Figure 4.27: Mixed Points sequence (real experiments).

scene is hard to identify in an automatic way based on appearance, distance or connectivity, what are *mixed* or real points.



(a) Step 1 (b) Step 2

Figure 4.28: *Mixed points* vs real impacts in real experiments sequence.

Trying to filter these *mixed points* without an strong prior knowledge of the elements in the environment, involves the risk of erasing real impacts points that come from narrow elements in the scene. These are the cases of, for example, railings, signpost poles, plant's stalks or the frame of glass walls (due to the laser usually does not

see the glass and would only see the frame). For that reason, and according to one of the main objectives of this thesis (the safety), it has been assumed to keep all the measures, even having *mixed points*, assuming that we are going to have false detections between two real objects but it will be safer than erase possible real impacts from dangerous elements.

4.5.1.2 Laser reflectivity

The movement detection in DOMap is based on Optical Flow, and as it was stated before in Section 4.4.1, it assumes intensity constancy to do the correspondence. In order to improve the performance of this algorithm, it would be appropriate to take into account more features than the ones that system is tracking. For that reason a possible approach could be to take into account the reflectivity, also called *Received Signal Strength Intensity (RSSI)* or remission. Figure 4.29(a) shows laser reflectivity phenomenon, which can be used as intensity in order to identify objects based on this parameter. It is strongly dependent on the beam impact angle on the object (as shown in Figure 4.29(b)) and this angle is influenced by the rugosity of the surface [Kaasalainen et al., 2011]. Therefore, this represents a weakness of this method. In addition, the colour of the object affects to the measure [Kneip et al., 2009]. Some authors tried to identify the lightness of the objects based on distance and reflectivity without success [Kawata et al., 2008].

(a) Laser Reflectivity (b) Impact angle Influence

Figure 4.29: *Laser reflectivity in real experiments.*

Moreover, the manufacturers usually do not provide further information about this parameter. The little information provided about the parameter is not adjusted, that means you can obtain different values from the same object with two different LIDARs. Making it even so more tricky to take into account this parameter.

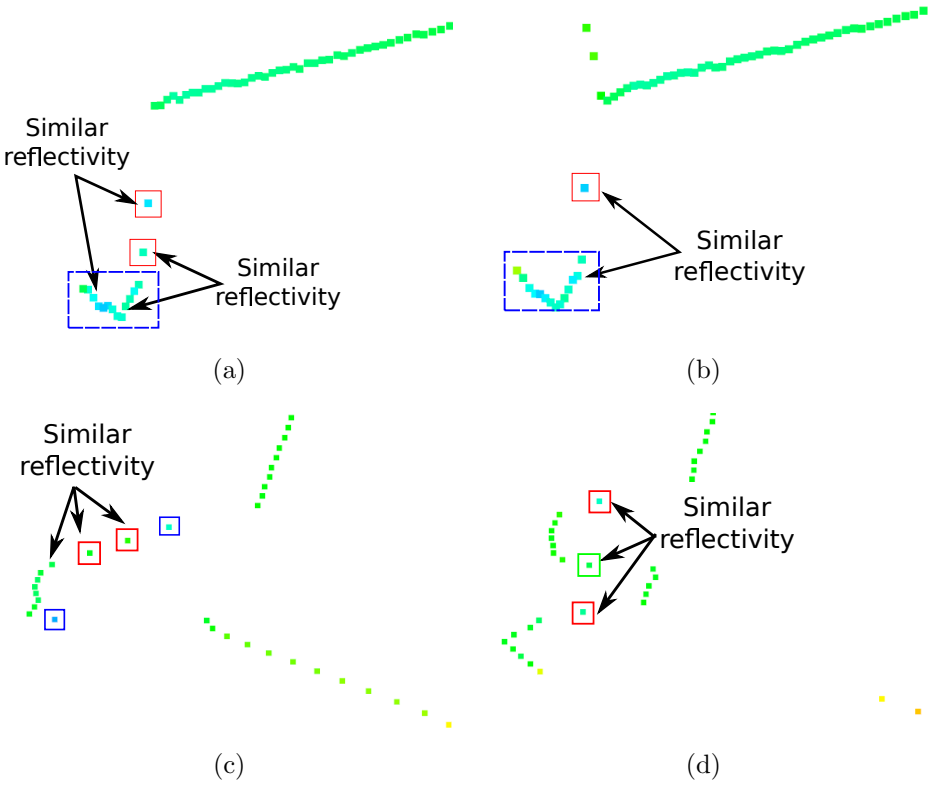


Figure 4.30: *Examples of LIDAR reflectivity in real experiments.*

Above all these difficulties, currently we are working on testing how this parameter improves the motion detection. Tests have been performed in this thesis with some of the commonly used LIDAR in robotics, such as *HOKUYO URG-04LX* and *SICK LMS151*. Several tests have been performed with dynamic obstacles, where the reflectivity variation of impacts over the same object ranges from the 10%

up to 28%.

In spite of this variation, this parameter helps the Optical Flow in the impacts matching, adding a feature to each impact point, assuming constancy in the intensity between consecutive time steps. This assumption is more accurate in the case of the *SICK LMS 151* due to its higher scanning frequency (50 Hz) compared with the *HOKUYO URG-04LX* (10 Hz).

Other possible use of this parameter could be identify the *mixed points* previously explained, in order to filter them. Figure 4.30 shows some scenes where *mixed points* appear, where the dot's colour represents the reflectivity value from *yellow* (low), *green* (medium) to *blue* (high). In the real experiments it has been proved the similar reflectivity between *mixed points* (impacts inside *red boxes*) and nearby real impacts. Only in the case of Figure 4.30(d) the *mixed points* in *blue boxes* can be identified based on the reflectivity with respect to their neighbours impacts. Due to this similarity between mixed and real points, this parameter has been discarded in order to identify *mixed points*.

4.5.2 Occlusion handling

The occlusions between objects, from the LIDAR point of view, is a common situation in dynamic environments that should be solved. This problem appears more often in cluttered environments. There are two situations to take into account:

- When a dynamic obstacle is occluded.
- When a static obstacle is occluded.

In the first case, the iDOMap system handles this situation by mean of its *Dynamic Occlusion Detector* stage (Figure 4.26) following these premises:

- The dynamic obstacle can be occluded, by other objects, that can be static or dynamic.
- The occluded obstacle should be farther from the robot than the object that produces the occlusion.

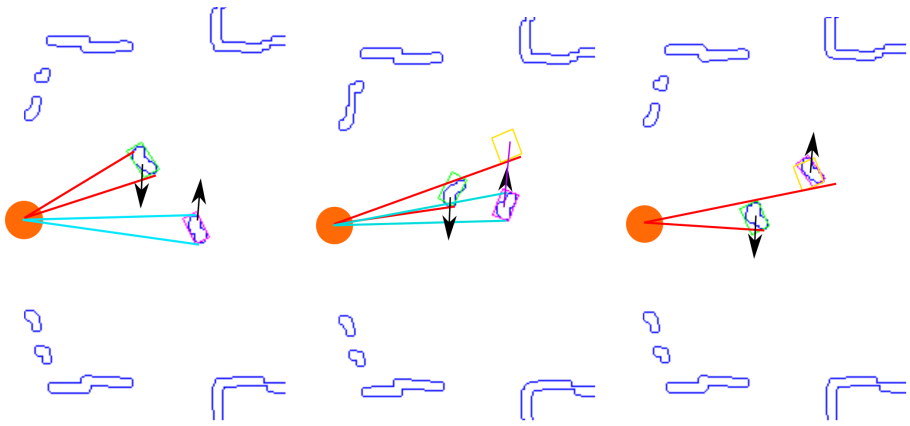
The algorithm stages are:

1. The *max* and *min* angles that form each dynamic obstacle with respect to the robot are obtained (*red* and *light blue lines* in Figure 4.31(a))

2. The occlusion occurs when these two angular sectors begin to overlap themselves (Figure 4.31(b)).

3. Once an occlusion is detected, the *occlusion ends area* where the occluded obstacle will probably appear is computed (*yellow square* in Figure 4.31(c)). This area is obtained in the intersection point between the *current velocity vector* of the occluded obstacle (*black arrow* in Figure 4.31(c)) and the angle of the obstacle that produces the occlusion (*max* or *min* depending on the case) that it is outside of the overlapped angular sector.

4. The *occlusion ends area* is kept in a *temporal memory*. It is computed based on the estimated time that the occluded obstacle would take to appear on the other side of the obstacle that produced the occlusion.



(a) No Occlusion (b) Occlusion Starts (c) Occlusion Ends

Figure 4.31: Dynamic Occlusion Detector: different stages.

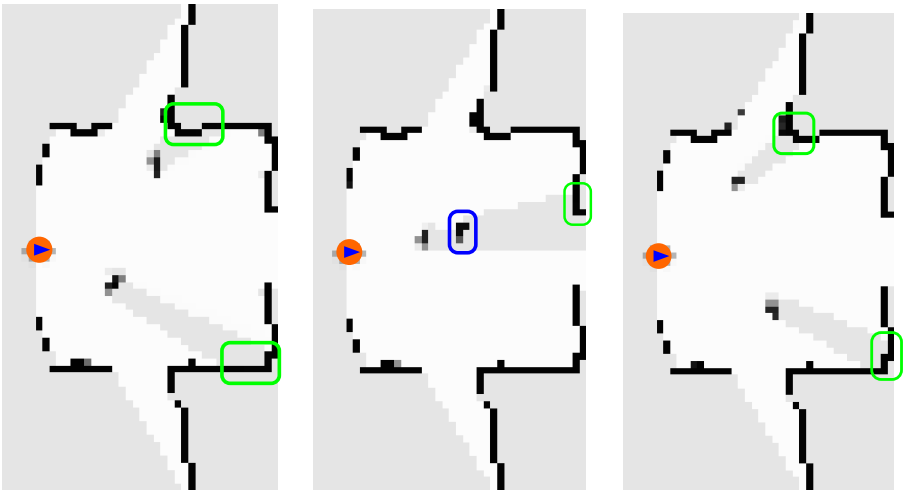
The estimated time that the dynamic occlusion lasts is computed based on the velocities of the previous time when the occlusion starts ($v_{x\_do} = v^{md}_{x(t-1)}, v_{y\_do} = v^{md}_{y(t-1)}$). These velocities ($v_{x\_do}$ and v_{y_do} in Figure 4.26) are sent to the *Tracking Stage* (that will be introduced in Section 4.5.3) in order to keep the obstacle prediction.

The second case is handled in order to avoid losing information of a static obstacle during an occlusion. This is handled by the *Static Occlusion Detector* stage as it is shown in Figure 4.26. When the occupancy probability is assigned to each cell, the occluded cell behind an occupied cell, from the point of view of the laser, has been considered in two ways, depending on its occupation probability:

- The occluded cell was previously free, then an *"occluded space"* probability is assigned ($P_{occ} = 0.5$).

- The occluded cell was previously occupied (occupancy probability above a threshold), therefore the probability of occupancy assigned (P_{occ_so} in Figure 4.26) is a bit higher than the *"occluded space"* in order to maintain it a bit further in time. In this way, it avoids the local planning tries go through obstacles previously detected.

Figure 4.32 shows the occupancy map (where *white* represents $P_{occ} = 0$ and *black* $P_{occ} = 1$) in different time steps of a real experiment where two persons were crossing in front of the robot causing several occlusions. Figures 4.32(a) and 4.32(c) show how the persons occlude parts of the static obstacle (marked with *green boxes*) and the iDOMap keeps their occupancy avoiding that static objects, such as walls, *"disappear"* during the occlusion by dynamic objects (this effect is known as *spatial memory*). Figure 4.32(b) shows both kinds of occlusions, the nearest dynamic object occludes the farther dynamic obstacle (marked with the *blue box*) and, at the same time, some part of the wall (marked with the *green box*). In both cases the occlusions are handled successfully.



(a) Occlusion 1: Static (b) Occlusion 2: Static and Dynamic (c) Occusion 3: Static

Figure 4.32: Occlusion Handling: Static and Dynamic obstacles.

4.5.3 Tracking and Filtering

In the previous versions of DOMap a tracking stage based on a KF and its extended version EKF was implemented. This tracking stage have been used assuming than the dynamic obstacles were differential drive robots which motion model could be linearised between time steps by mean of a Jacobian. In the improved version, we propose to use a system that can deal with a multi hypothesis tracking, avoiding to use a concrete model. Based on the connectivity of the cells, we can detect a *blob* (object) that can be useful to track. Due to the iDOMap's general approach, the tracking stage has been achieves using a PF. This tracking stage can manage the noise in measurement and motion models, in a way that help us to predict the *centroid* of the object during situations where the rest of the system does not provide enough information, such as the occlusions.

The PF is a Bayesian random-sampling algorithm that allows us to use non-linear motion models assuming that the object dynamics form a temporal Markov chain. This means that the state (x_t) is only conditioned by the previous state, independent of the earlier history

$(x_t = x_1, ..., x_t)$, allowing quite general dynamics. The PF uses a set of random samples to represent the posterior belief function $bel(x_t)$. The samples are drawn from the posterior and they are called *particles* and denoted like:

$$x_t^k = \begin{bmatrix} x^k \\ y^k \\ \theta^k \end{bmatrix} \tag{4.28}$$

where the particle x_t^k is a concrete instantiation of the state at time t and it has a importance factor (weight) denoted w_t^k. This weight describes the importance of particle k in the set of N particles, denoted:

$$S_t = x_t^1, x_t^2, ..., x_t^N \tag{4.29}$$

The associated importance factors are denoted:

$$W_t = w_t^1, w_t^2, ..., w_t^N \tag{4.30}$$

The likelihood that the hypothesis state x_t (*centroid* in our case) to be included in the particle set S_t should be proportional to its posterior $bel(x_t)$ of its Bayes filter, in the ideal case:

$$x_t \sim p(x_t | o_{1:t}, u_{1:t}) \tag{4.31}$$

where $o_{1:t}$ are the measurements and $u_{1:t}$ are the control inputs. The denser area of the state space indicates the more likelihood that the true state falls into this area.

The PF algorithm recursively constructs the belief $bel(x_t)$ from the previous belief $bel(x_{t-1})$. As the particles represent a set of beliefs, the particle set S_t is built recursively from the previous set S_{t-1}. Hence, particle filter is a recursive algorithm that operates in two phases: *prediction* and *update*. This means that after each control input, every particle is modified according to the motion model in the *prediction* stage. For our proposal, the control input is defined as $u_t = (v_x^{md}, v_y^{md}, \Delta\theta)$, where v_x^{md} is the velocity in X *axis* and v_y^{md} is the velocity in Y *axis* provided by the *Movement Detector stage*

(shown in Figure 4.26) and $\Delta\theta$ is the incremental orientation change, given by Equation 4.32

$$\Delta\theta = arctan\left(\frac{\Delta v_x^{md}}{\Delta v_y^{md}}\right) \tag{4.32}$$

where, $\Delta v_x^{md} = v_{x_{t-1}}^k - v_{x_t}^k$ and $\Delta v_y^{md} = v_{y_{t-1}}^k - v_{y_t}^k$. The generic motion model is given by Equation 4.33.

$$x_{t+1}^k = \begin{bmatrix} x^k + v_x^{md} \cdot t \\ y^k + v_y^{md} \cdot t \\ \theta^k + \Delta\theta \end{bmatrix} + n_t \tag{4.33}$$

where n_t is the noise vector.

After obtaining information from the *Movement Detector stage*, the particles are weighted based on the euclidean distance to the *blob centroid* obtained by the *Blob Filtering stage*. The PF used is based on *iCONDENSATION* algorithm [Isard and Blake, 1998]. Also, the samples set size is fixed, in a way that guarantees that the algorithm run in successive time steps with the same computational cost. This approach is based on the *CONDENSATION* algorithm that had been extended by the authors adding *"Importance Sampling"* [Ripley, 1987]. This sampling approach improves the efficiency when an auxiliary knowledge (from others sensors) is available, as shown in Figure 4.33, where the motion model predicts the objects remain on the left (the *white samples*) and the *black samples* are positioned based on the auxiliary knowledge that the object has moved to the right. The approximation is more accurate than without this information and conferring robustness to temporary measurements failures. The importance sampling eliminates the low weighted particles and concentrates the filter in the particles with high weights.

In the experiments of iDOMap, that will be shown in the next sections, there are not others sensors to measure the environment than LIDAR. However this feature, in the same way than the BOF, allows to fusion measurements from different sources when they are available, improving, in this case, the iDOMap in terms of robustness and accuracy.

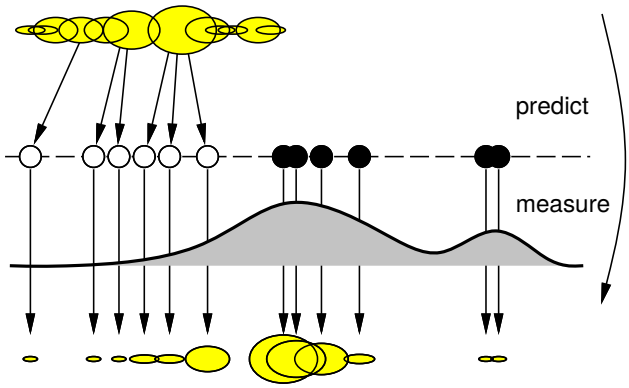


Figure 4.33: iCONDENSATION Algorithm [Isard, 1998].

In the velocity estimation stage a filtering state has been added. A *sliding window median filter* has been chosen in order to minimize the effects of the outliers provided by the *Movement Detector* stage.

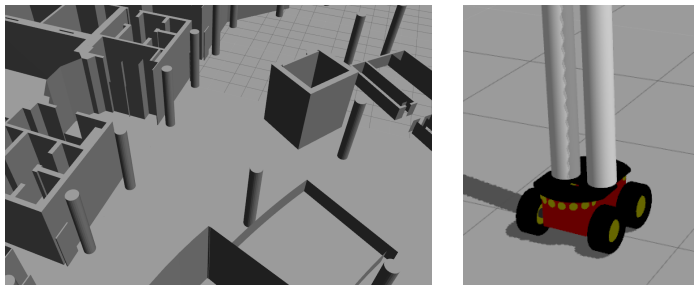
4.5.4 Implementation and Results

Section is organized as follow: firstly Test Bed for simulation and real experiments will be presented; secondly Ground Truth for real experimentation will be shown; simulated experiments will be described in Section 4.5.4.3 and finally real experiments will be described in Section 4.5.4.4 in order to analyse the performance of the iDOMap. The system outputs will be related to the robot coordinates (*X forward, Y left, Z up*), due to our proposal is a local mapping approach. Following the same coordinates system than the used in ROS (*right hand rule*) allows us the direct integration with others systems.

4.5.4.1 Test Bed

The Test Bed in simulated experiments has been performed using ROS connected to *Gazebo* as simulator. The simulated experiments have been performed in a 3D map of the *Polytechnic School* (Figure 4.34(a)). In this environment there will be dynamic obstacles moving around the robot. These dynamic obstacles have been sim-

ulated with other robots, in order to control the velocities and the paths that they follow and to obtain, in this way, the Ground Truth for the simulated results. In addition, in order to simulate the legs of a person, some robots have two vertical cylinders above the platform (Figure 4.34(b)).



(a) Simulated environment (b) Dinamic obstacle

Figure 4.34: Simulated Test Bed and a robot simulating a person.

The real experiments have been performed in a controlled indoor environment in the *Polytechnic School* at the *University of Alcalá (UAH)*. The area of the environment is approximately 7x10 metres, where there are three doors and two lifts (Figure 4.35).

Figure 4.35: Test Bed: Indoor scenario.

Although the proposed method is a *Grid-Based* approach and the output of the system is an occupancy grid with velocities and labels associated to each grid, as it is shown in the bottom part of the

iDOMap diagram (Figure 4.26), the results will show the data at the *"object level"* in order to clarify the explanation.

4.5.4.2 Ground Truth for real experiments

While the Ground Truth in simulation has been stated previously, obtaining the Ground Truth in robotics real experiments is always a big challenge, more over when there are agents in the environment that can not provide information about their own states, such as persons, animals, etc. In DATMO systems there are several approaches; in outdoors environments each dynamic obstacles can carry out a GPS as the authors proposed in [Mekhnacha et al., 2008], other authors considered that in presence of uninstrumented moving objects any system will have a difficult time assessing the accuracy of tracking data [Wang, 2004]. For that reason some authors only report *qualitative* results, such as *"all the dynamic obstacles are well detected with an accurate velocity"* in [Nègre et al., 2014]. On the other hand, other authors proposed an affordable approach for people tracking Ground Truth, based on markers in the ground, where the people are walking and a stop-watch to measure the time between marks [Mertz et al., 2013] (called visual marked Ground Truth).

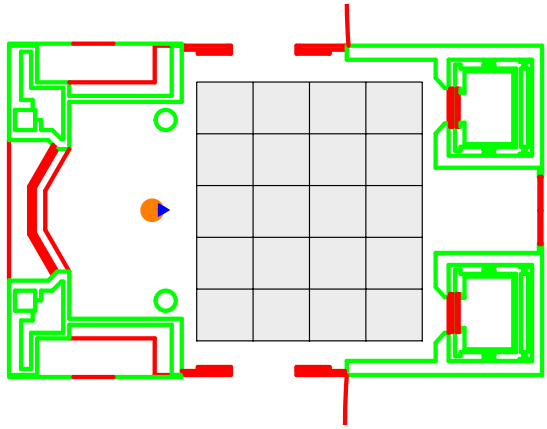


Figure 4.36: Ground Truth: Indoor scenario.

In this thesis, a step forward Ground Truth was performed based on this idea. A video of the scene is recorded synchronized with the

laser measurements of the robot. The dynamic obstacles move in the area that is recorded by the camera where a grid in the floor (*grey grid* in Figure 4.36) is present. Then, hand-made frames are selected when the obstacles are located in key points of the grid. The dimensions of the floor grid are well known and a constant velocity between these points is assumed. A more accurate Ground Truth is beyond the scope of this thesis. The desirable Ground Truth would be an intelligent environment big enough to perform manoeuvres and able to detect the pose and velocities of all participants in the scene. Due to the lack of this kind of Ground Truth system and in order to measure the position and angle of the dynamic obstacles in the most accuracy way, the robot (*orange circle* in Figure 4.36) was placed in a well known and static position respect to the floor grid.

4.5.4.3 iDOMap Simulated Results

The simulated experiments have been performed in order to know when and where exactly our robot and the participants are. This complete knowledge over the scenario allows us to obtain results while the robot is moving in the environment. In this way, several scenarios are going to be explained.

- ***Scenario 1 - Two objects in a perpendicular path:*** In this scenario there are two dynamic obstacles simulated persons crossing in perpendicular paths in front of the robot at 0.25 m/s and 0.3 m/s. The paths followed by the obstacles are shown in Figure 4.37. This scenario shows the performance of the proposal across the $Yaxis$ of the robot at low velocities.

  Figure 4.38 shows the velocities detection at each axis compared with the Ground Truth. Table 4.4 shows the errors in velocities detection where it can be seen that the error is around 0.05 m/s at each axis.

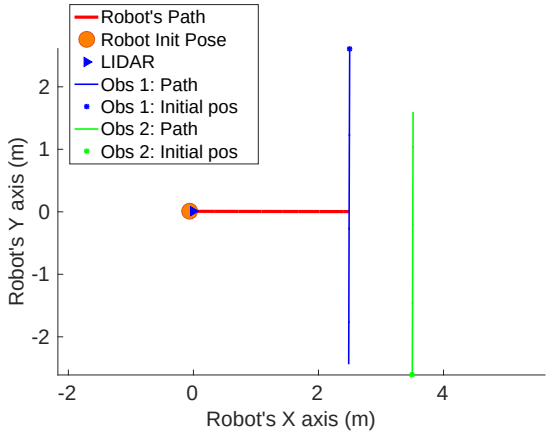


Figure 4.37: Scenario 1 - Paths.

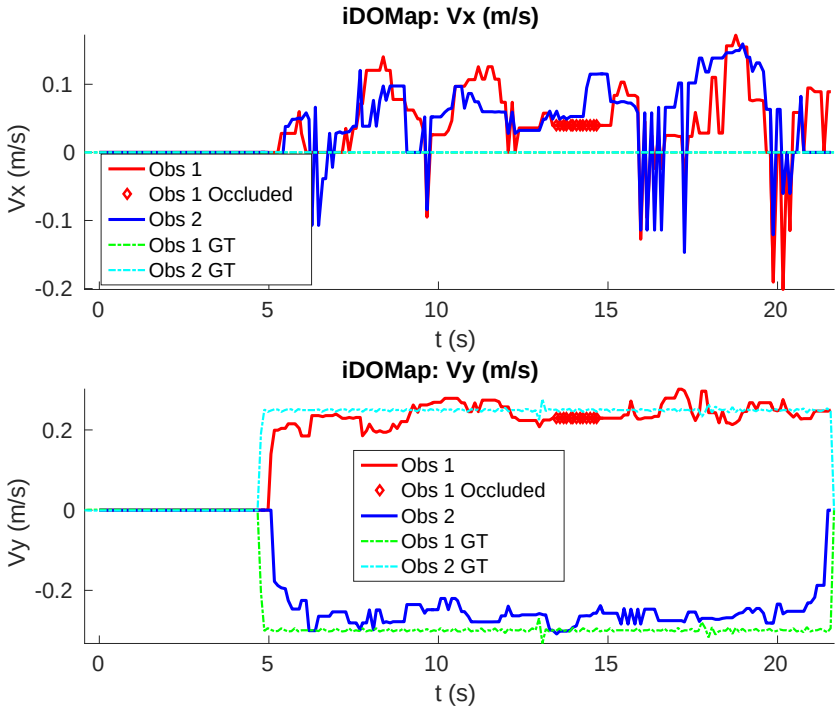


Figure 4.38: Scenario 1 - Results: Velocities detection.

Table 4.4: Scenario 1 - Errors in velocities detection (m/s)

	Obs 1		Obs 2	
	V_x	V_y	V_x	V_y
$\bar{\epsilon}$	0.051	0.038	0.045	0.021
σ_ϵ	0.045	0.049	0.047	0.030

- ***Scenario 2 - Two objects in a diagonal path:*** In this scenario, there are two dynamic obstacles simulating (two persons) crossing in diagonal paths in front of the robot at 0.25 m/s and 0.3 m/s, each one following the paths shown in Figure 4.39.

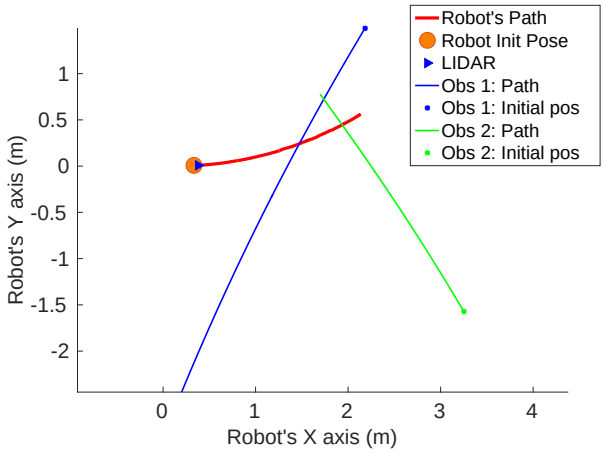


Figure 4.39: Scenario 2 - Paths.

The paths followed by the obstacles produced that the *obstacle 1* (*blue path*) occluded the *obstacle 2* (*green path*). The occlusions detected are marked with *blue diamonds* in Figure 4.40, where it can be seen that the velocities during the occlusion increase a bit in the $Xaxis$ and decrease a bit in $Yaxis$ due to the *Dynamic Occlusion Detector* and the *Tracking* stages detect and predict the occlusion respectively. The errors in velocities detection of each obstacle along each axis are shown in Table 4.5.

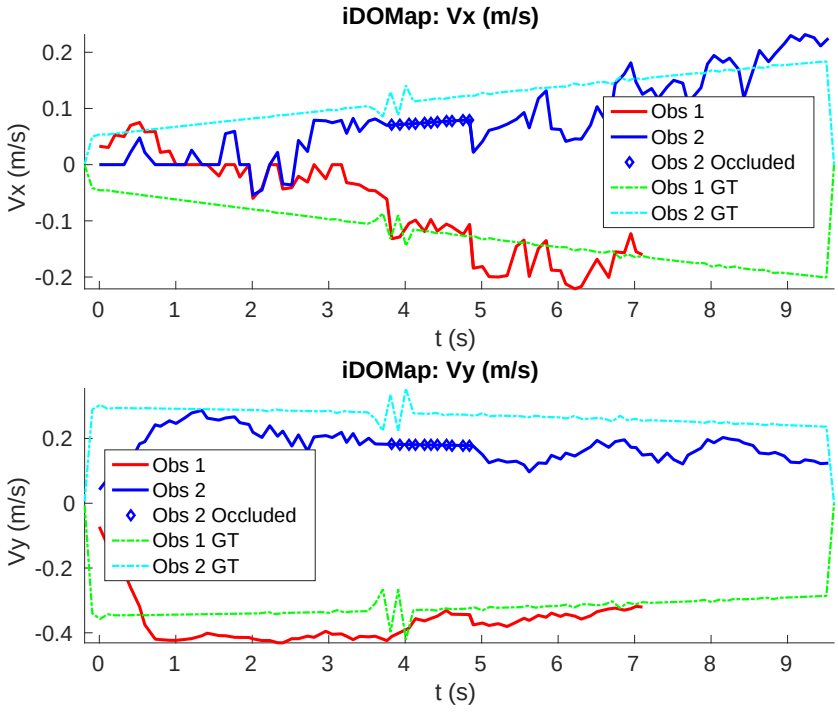


Figure 4.40: Scenario 2 - Results: Velocities detection.

Table 4.5: Scenario 2 - Errors in velocities detection (m/s)

	Obs 1		Obs 2	
	V_x	V_y	V_x	V_y
$\overline{\epsilon}$	0.054	0.062	0.047	0.093
σ_ϵ	0.032	0.045	0.029	0.042

- ***Scenario 3 - Two object in a diagonal path (fast):*** This scenario is similar to the previous one, but in order to simulate the velocities performed by humans, in this case the obstacles are moving with speeds of 0.8 m/s and 1 m/s while the robot is moving in an arc path with a linear velocity of 0.45 m/s and angular velocity of 0.1 rad/s, as shown Figure 4.41.

  Table 4.6 shows the errors in velocity detection at each axis. In this case the error of the *obstacle 1* in Y axis ($\overline{\epsilon}_{V_y}$) is a bit

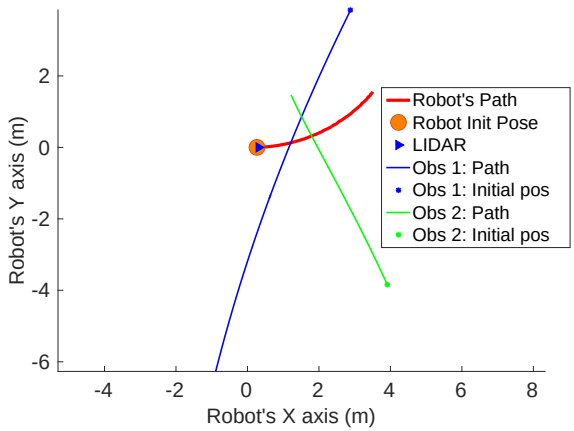


Figure 4.41: Scenario 3 - Paths.

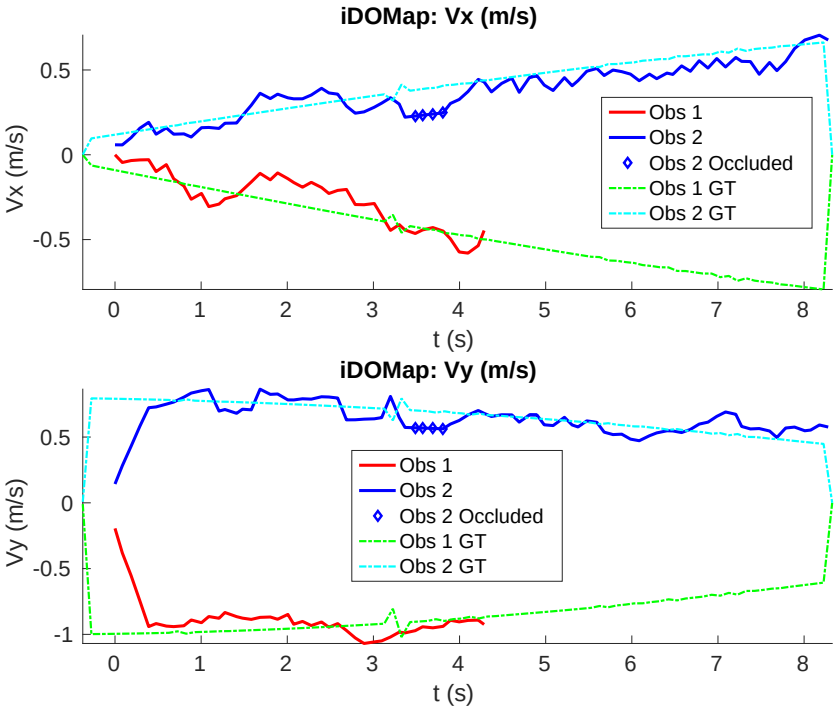


Figure 4.42: Scenario 3 - Results: Velocities detection.

higher due to the occlusion.

Table 4.6: Scenario 3 - Errors in velocities detection (m/s)

	Obs 1		Obs 2	
	V_x	V_y	V_x	V_y
$\overline{\epsilon}$	0.076	0.109	0.067	0.085
σ_ϵ	0.046	0.153	0.044	0.098

- ***Scenario 4 - Manoeuvre:*** This experiment is performed by an object that is moving along the *blue path* in Figure 4.43 while the robot performs the *red path* in Figure 4.43. In this scenario it can be seen how the our proposal performs with simultaneous velocities changes in both axes, due to the approximately circular path that the obstacle follows.

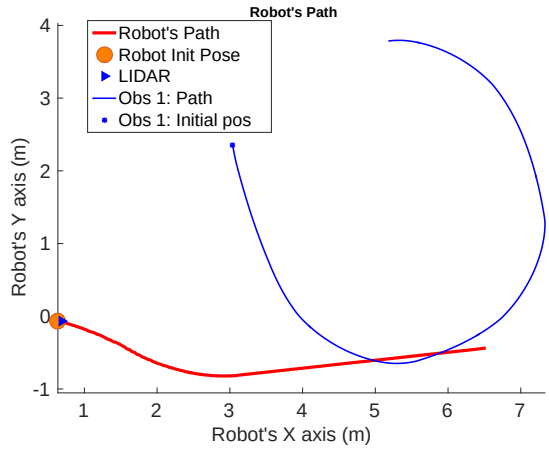


Figure 4.43: Scenario 4 - Paths.

Figure 4.44 shows the results of this scenario, where it can be seen that velocities are detected in both axis (Vx and Vy) up to 0.5 m/s with the errors shown in Table 4.7. The main errors (between seconds 6 to 8) are due to the low number of impacts received from the obstacle when it is in front of the robot in an angled position.

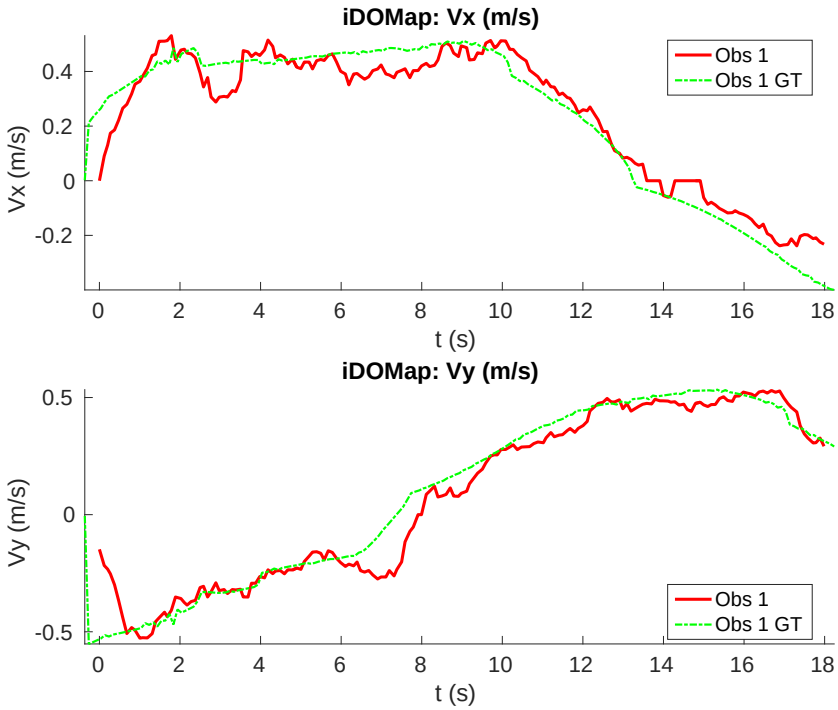


Figure 4.44: Scenario 4 - Results: Velocities detection.

Table 4.7: Scenario 4 - Errors in velocities detection (m/s)

	V_x	V_y
$\overline{\epsilon}$	0.057	0.045
σ_ϵ	0.053	0.062

- ***Scenario 5 - An object overtaking the robot:*** in this scenario a dynamic obstacle overtake the robot with a speed of 0.6 m/s while the robot goes at a speed of 0.3 m/s. This scenario tests the velocities detection along the $Xaxis$. The path followed by the obstacles is shown in Figure 4.45, where it can be seen a little curvature in the paths, due to the little odometry error that the Gazebo simulator introduces.

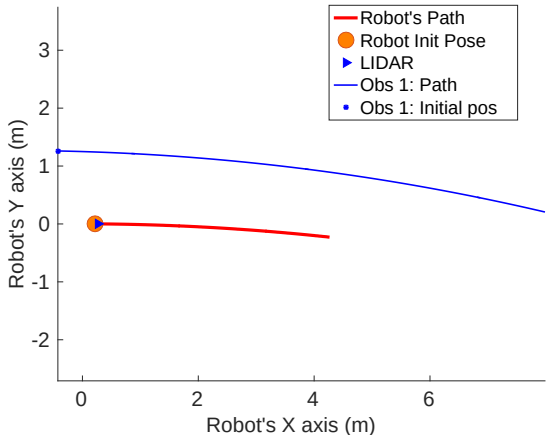


Figure 4.45: Scenario 5 - Paths.

Figure 4.46 shows the results of the perception stage, where it can be seen that velocity is quite close to the real one, except at the beginning of the experiment, due to the robot only see a part of the obstacle.

Table 4.8 show the errors in velocities detection, where the error is higher in the $Xaxis$ due to the obstacles moves in this axis with a very low velocity, that is hard to detect in an accurate way.

Table 4.8: Scenario 5 - Errors in velocities detection (m/s)

	V_x	V_y
$\bar{\epsilon}$	0.089	0.023
σ_ϵ	0.093	0.017

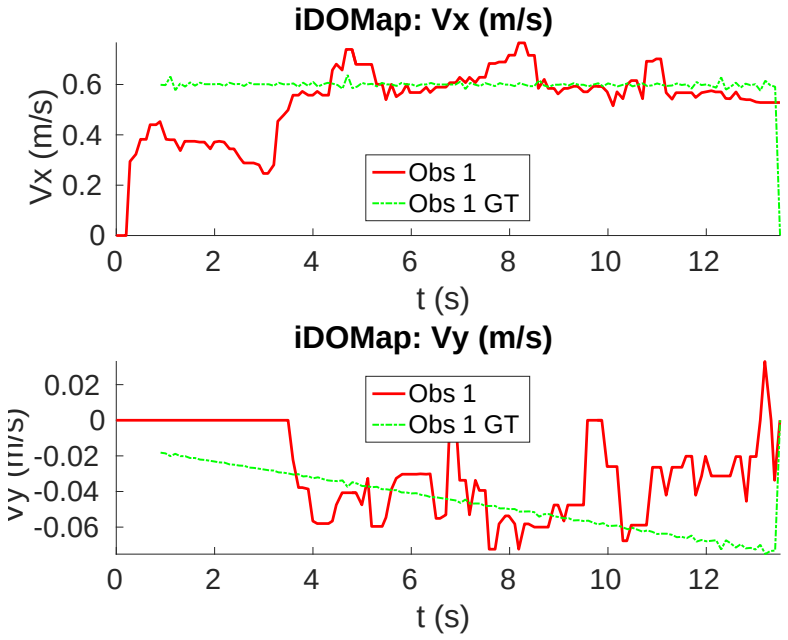


Figure 4.46: Scenario 5 - Results: Velocities detection.

4.5.4.4 iDOMap Real Results

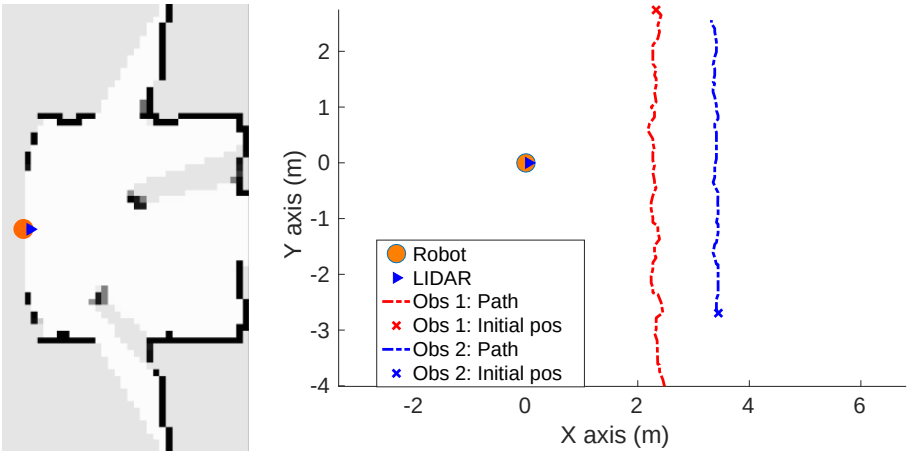
Due to one of the objectives of this thesis is to get a safe interaction between humans and robots, it is interesting to test the iDOMap in a real scenario with dynamic obstacles. As the Ground Truth (Section 4.5.4.2) is limited to the camera FOV, in some experiments, the Ground Truth ends before than the output of proposal system, due to the laser FOV is wider than the camera FOV. As mentioned before, in those experiments the robot has been placed in a static position while the participants are moving in front of the robot. The experiments with the robot in movement will be shown in the Section 5.1. Although at the beginning of each scenario, the *persons* could be enumerated to introduce the scenarios, in the next paragraphs will be denoted as *obstacles* due to the proposal do not take into account which kind of dynamic obstacle they are.

- ***Scenario 1 - Two persons crossing:*** In this scenario two persons cross perpendicularly to the robot and the *obstacle 1* produces an occlusion over the *obstacle 2*, as can be shown in Figure 4.47(b). This is a comparable scenario with the *Scenario 1* in simulated experiments (Section 4.5.4.3) and shows a real situation when obstacles appear from other room or corridor. Figure 4.47 shows images during the experiments while the obstacles follow the detected paths shown in Figure 4.48(b). In addition, an occupancy grid map of this scenario during the experiment is shown in Figure 4.48(a). In this occupancy grid, it can be seen as the cells occupied by the static and dynamic obstacles have high occupancy probability (the darker the higher is the probability)

(a) Crossing t_1 (b) Crossing t_2 (c) Crossing t_3

Figure 4.47: Scenario 1 - Images sequences.

Figure 4.49 shows the velocities detected at each axis of both obstacles, where it can be seen that the velocities of the *obstacle 2* are maintained during the occlusion. Table 4.9 shows the errors in the velocities detection for this scenario, where $\bar{\epsilon}$ is the average error and σ_ϵ the standard deviation of the following parameters: Vx is the obstacle velocity along X axis, Vy the obstacle velocity along Y axis, $||V||$ is the vector velocity magnitude and θ the vector angle.



(a) Occupancy Grid (b) Detected Paths

Figure 4.48: Scenario 1 - Occupancy Grid and Paths detected.

Table 4.9: Scenario 1 - Errors in velocities detection

		V_x (m/s)	V_y (m/s)	$\|\|V\|\|$ (m/s)	θ (°)
Obs 1	$\overline{\epsilon}$	0.011	0.153	0.152 (13.12%)	0.71
	σ_ϵ	0.036	0.127	0.127 (10.70%)	2.19
Obs 2	$\overline{\epsilon}$	0.0137	0.092	0.092 (9.75%)	0.79
	σ_ϵ	0.035	0.075	0.075 (8.12%)	2.05

- ***Scenario 2 - Two persons in diagonal paths:*** This scenario shows two persons crossing in several diagonals paths. Each path crosses the other as it is shown in Figure 4.51(b). This scenario is similar to the *Scenario 2 and 3* in simulated scenarios (Section 4.5.4.3), in order to compare with it. Figure 4.50(a) shows an example of the occupancy grid computed during this experiment.

  The detection of the *obstacle 2* worsens between seconds 5 and 6 (Figure 4.52) due to the low amount of laser impacts. This situation appears when the obstacle (a person in this scenario) is located laterally to the laser. Although the vector velocity magnitude decreases in this situation (top part of Figure 4.53),

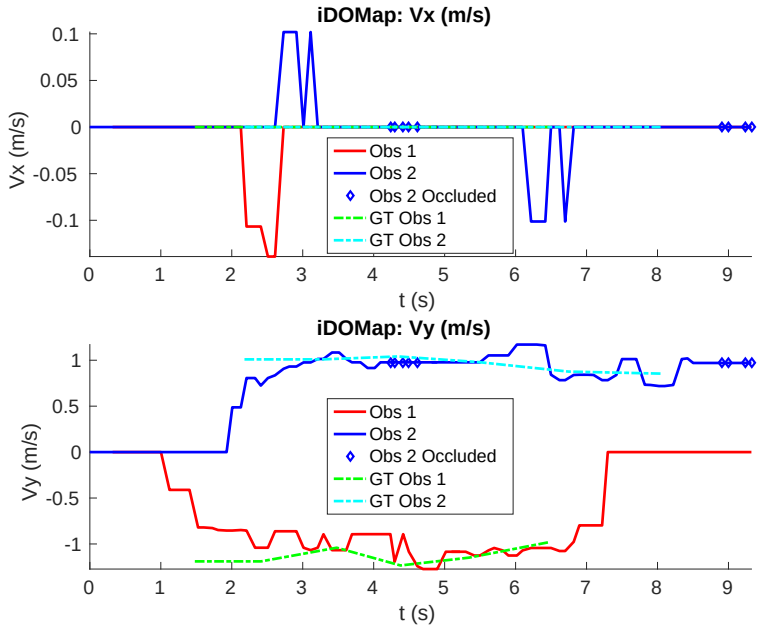


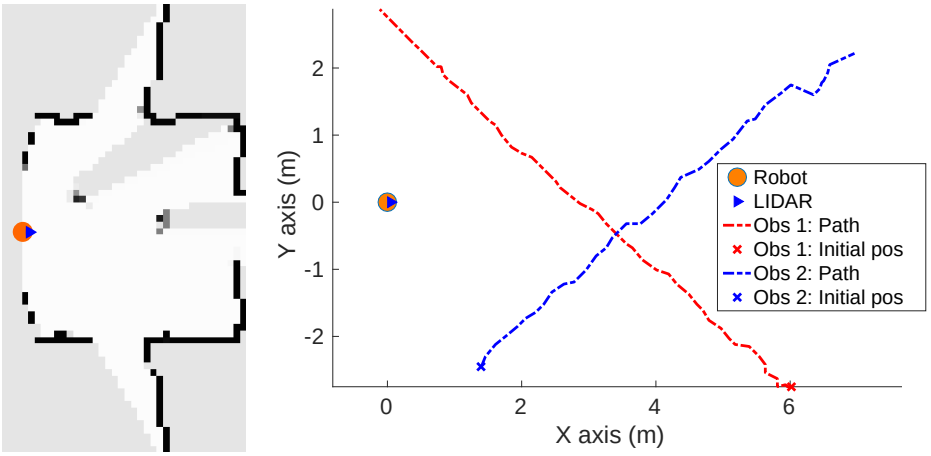
Figure 4.49: Scenario 1 - Results: Velocities detection.

the vector angle of the obstacle (bottom part of Figure 4.53) keeps almost the true direction. This is an important fact to maintain the safety for the obstacle avoidance system.

(a) Diagonal t_1 (b) Diagonal t_2 (c) Diagonal t_3

Figure 4.50: Scenario 2 - Images sequence.

Table 4.10 shows the averages velocity errors ($\bar{\epsilon}$) and their standard deviation (σ) for each axis.



(a) Occupancy Grid (b) Detected Paths

Figure 4.51: Scenario 2 - Occupancy Grid and Paths detected.

Table 4.10: Scenario 2 - Errors in velocities detection

		V_x (m/s)	V_y (m/s)	$\|\|V\|\|$ (m/s)	θ (º)
Obs 1	$\overline{\epsilon}$	0.056	0.073	0.078 (12.19%)	7.44
	σ_ϵ	0.041	0.045	0.063 (17.88%)	10.57
Obs 2	$\overline{\epsilon}$	0.144	0.098	0.170 (17.86%)	5.10
	σ_ϵ	0.130	0.072	0.141 (16.21%)	5.87

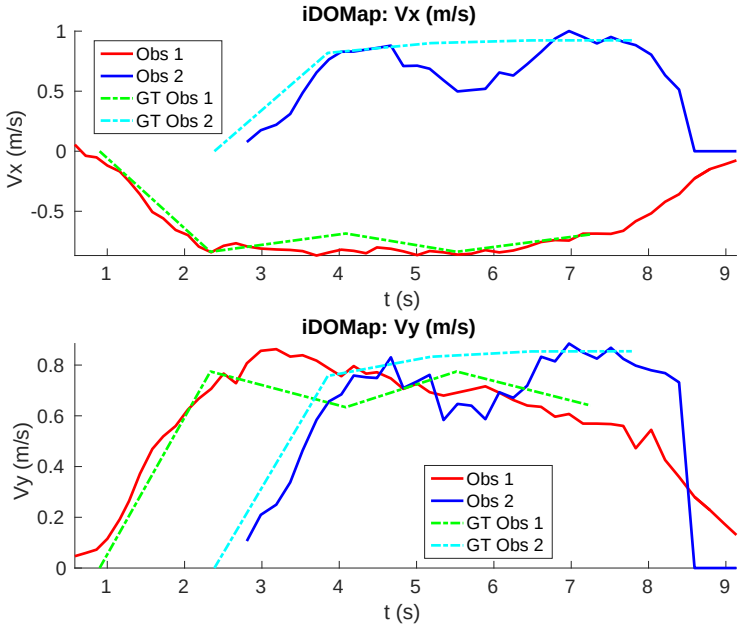


Figure 4.52: Scenario 2 - Results: Velocities detection.

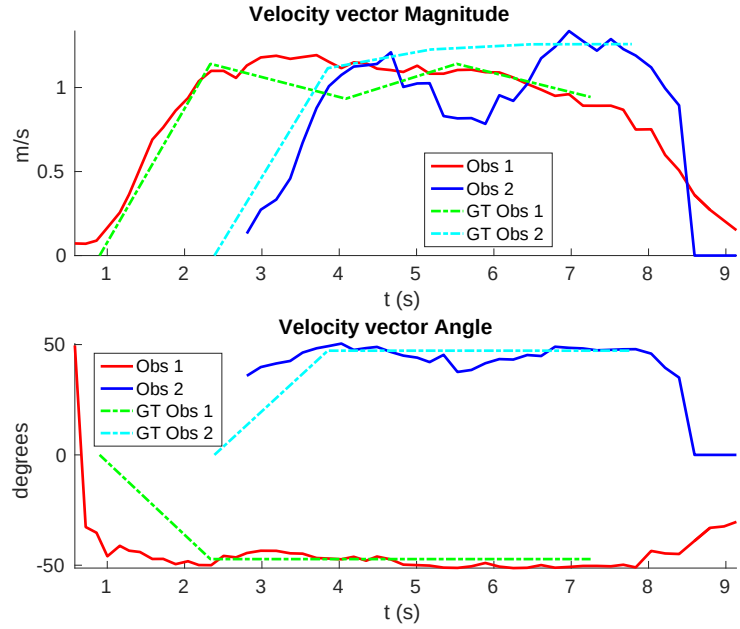


Figure 4.53: Scenario 2 - Results: Angle and Magnitude vectors.

- ***Scenario 3 - Two persons in opposite diagonal paths:*** This scenario is similar to the previous one, but the participants walk in the opposite direction following diagonal paths. Figure 4.54 shows an images sequence of the scenario and the velocities detection results are shown in Figure 4.55.

(a) Diagonal t_1 (b) Diagonal t_2 (c) Diagonal t_3

Figure 4.54: Scenario 3 - Images sequence.

Figure 4.55 shows the velocities detection, where it can be seen that around the second 5 the obstacles velocities detection decreases in both axis. This effect is due to the low amount of laser beams that impact in the obstacles when is located laterally in front of the robot, similar to the previous scenario.

Table 4.11 shows the errors in the velocities detection.

Table 4.11: Scenario 3 - Errors in velocities detection

		V_x (m/s)	V_y (m/s)	$\|\|V\|\|$ (m/s)	θ (°)
Obs 1	$\overline{\epsilon}$	0.116	0.075	0.126 (13.32%)	5.90
	σ_ϵ	0.089	0.056	0.159 (16.86%)	6.48
Obs 2	$\overline{\epsilon}$	0.193	0.105	0.190 (17.09%)	5.81
	σ_ϵ	0.202	0.202	0.220 (19.29%)	8.65

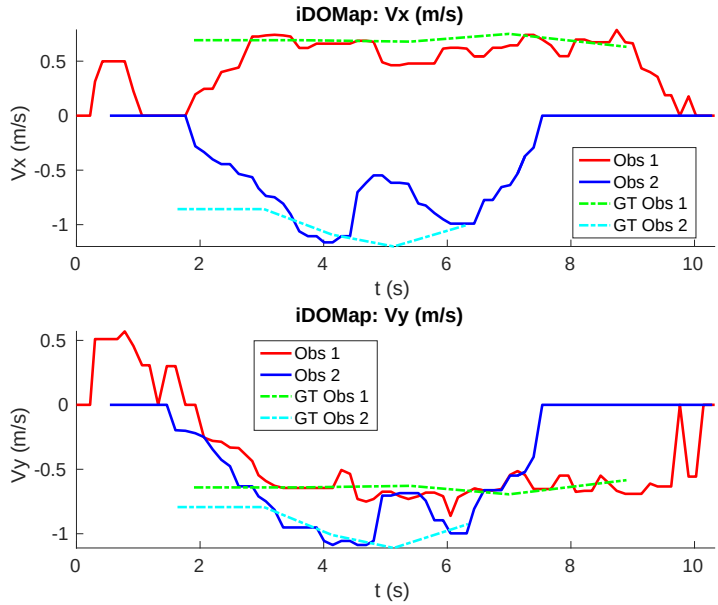


Figure 4.55: Scenario 3 - Results: Velocities detection.

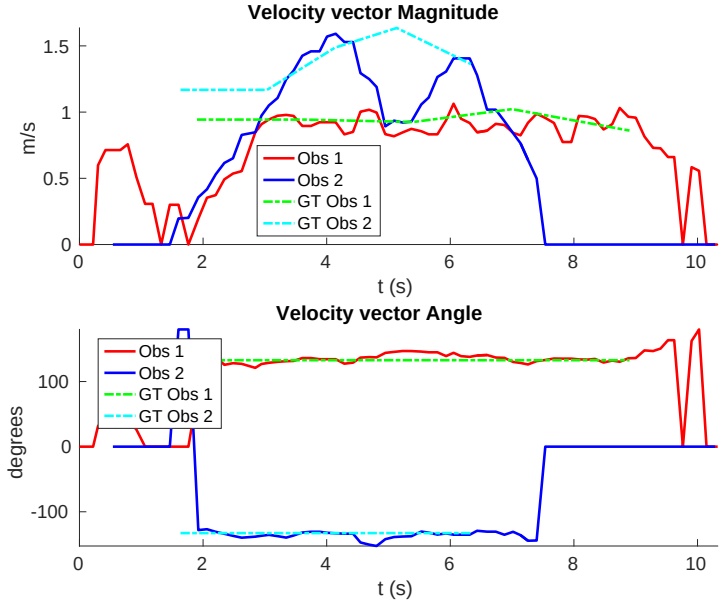


Figure 4.56: Scenario 3 - Results: Angle and Magnitude vectors.

- ***Scenario 4 - Person walking randomly:*** In this scenario a participant is walking in different directions inside the interesting area, where the Ground Truth is available, with several direction changes, as it can be seen in Figure 4.57. This scenario has been selected to test the performance of the proposal with several direction changes, also in a short period of time.

(a) Random walk t_1 (b) Random walk t_2 (c) Random walk t_3

Figure 4.57: Scenario 4 - Images sequence.

Figure 4.58 shows the path detected by our proposal, where it can be seen that despite the frequent and rapid changes of direction that the obstacle describes, the system can detect the velocity at each axis with an error less than 0.1 m/s, as it is shown in Table 4.12. Also the position tracked (shown in *red* in Figure 4.58) follows in a good manner the Ground Truth (shown in *blue*).

Table 4.12: Scenario 4 - Errors in velocities detection

		V_x (m/s)	V_y (m/s)	$\|\|V\|\|$ (m/s)	θ (º)
Obs 1	$\overline{\epsilon}$	0.111	0.100	0.144 (14.31%)	11.58
	σ_ϵ	0.090	0.095	0.117 (13.34%)	21.98

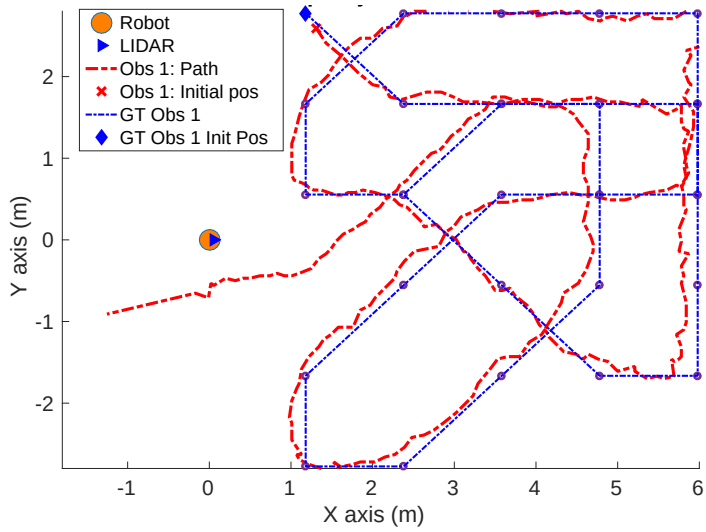


Figure 4.58: Scenario 4 - Results: Path described by the person.

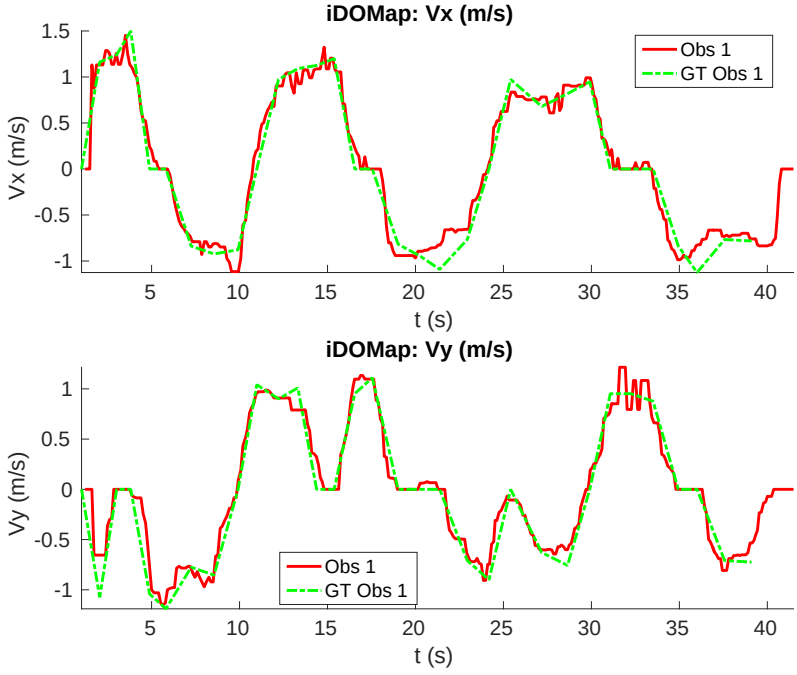
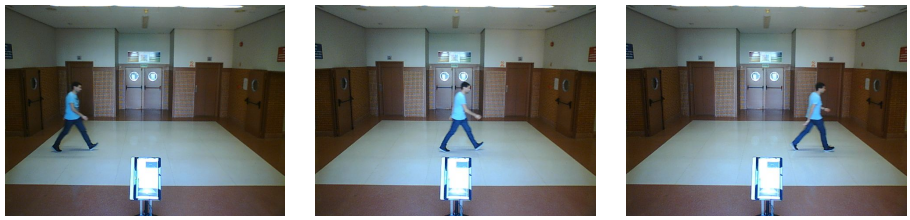


Figure 4.59: Scenario 4 - Results: Velocities detection.

- ***Scenario 5 - A person walking fast***: In this scenario (Figure 4.60) a person walking fast (about 2 m/s) crosses assuming a perpendicular path respects to the robot.



(a) Fast walk t_1 (b) Fast walk t_2 (c) Fast walk t_3

Figure 4.60: Scenario 5 - Images sequence.

The results of the velocities detection of the obstacle are shown in Figure 4.62, where it can be seen the path of the dynamic obstacle has been detected and its velocities estimated with an error about 10% shown in Table 4.13.

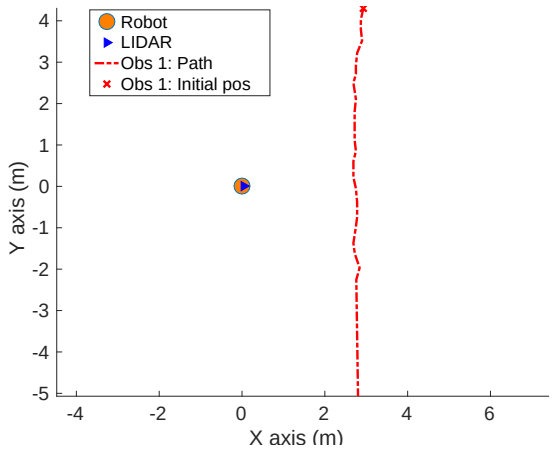


Figure 4.61: Scenario 5 - Results: Path detected.

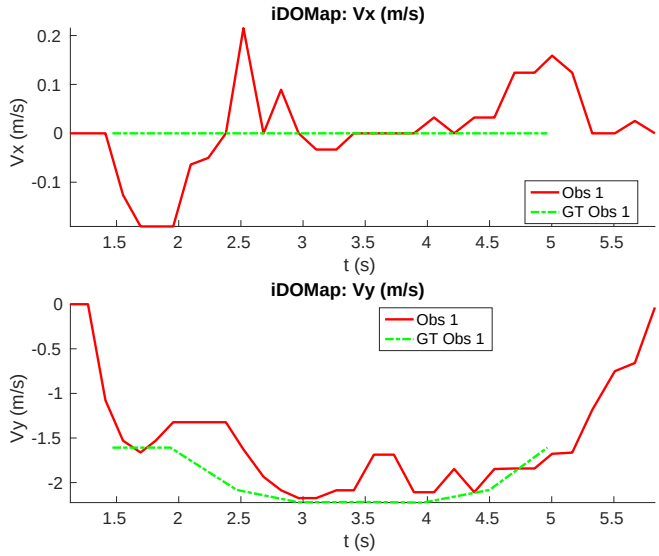


Figure 4.62: Scenario 5 - Results: Velocities detection.

Table 4.13: Scenario 5 - Errors in velocities detection

		$V_x\ (m/s)$	$V_y\ (m/s)$	$\|\|V\|\|\ (m/s)$	$\theta\ (^o)$
Obs 1	$\overline{\epsilon}$	0.066	0.22	0.226 (11.35%)	2.33
	σ_ϵ	0.073	0.204	0.203 (10.11%)	2.74

- ***Scenario 6 - A person running***: In this scenario a person running (near to 3.5 m/s) crosses assuming a perpendicular path in order to test our proposal in higher speeds.

(a) Running t_1 (b) Running t_2 (c) Running t_3

Figure 4.63: Scenario 6 - Images sequence.

The results of the velocities detection of the obstacle are shown in Figure 4.65, where it can be seen the path of the dynamic obstacle has been detected and its velocities estimated with an error about 12% shown in Table 4.14. The velocities detection accuracy decreases about the second 4 due to the obstacle starts to be occluded by the left wall. For that reason, Vy of the obstacle has been maintained in the end of the experiment.

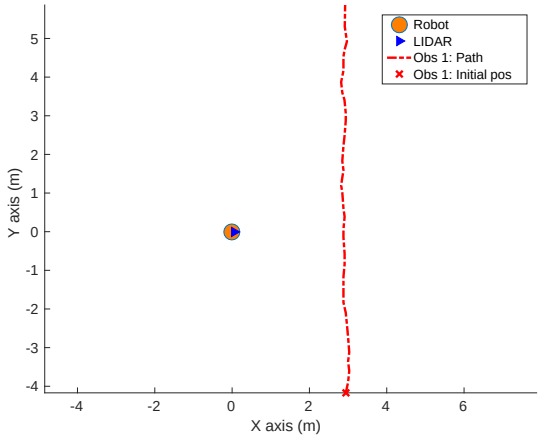


Figure 4.64: Scenario 6 - Results: Path detected.

Table 4.14: Scenario 6 - Errors in velocities detection

		V_x (m/s)	V_y (m/s)	$\|\|V\|\|$ (m/s)	θ (º)
Obs 1	$\overline{\epsilon}$	0.142	0.362	0.360 (11.97%)	2.79
	σ_ϵ	0.109	0.237	0.236 (7.87%)	2.19

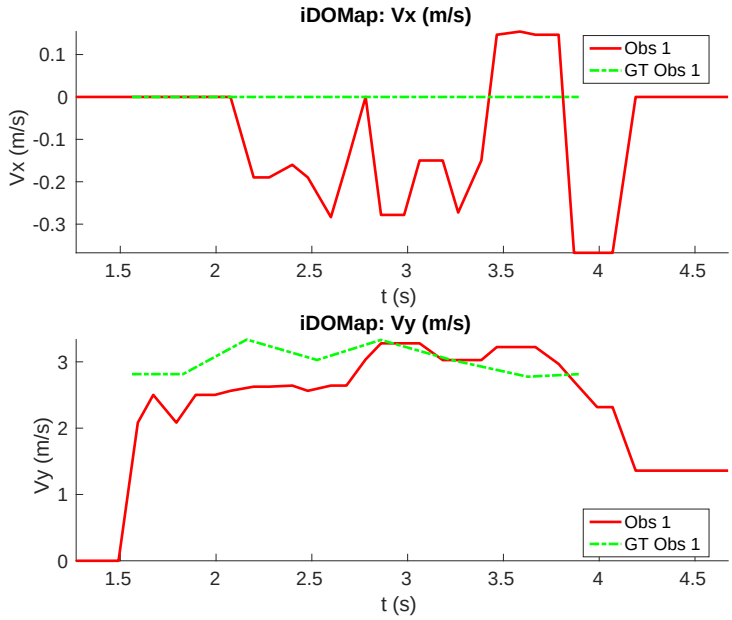


Figure 4.65: Scenario 6 - Results: Velocities detection.

- ***Scenario 7 - A person crossing wearing loose clothes:*** In order to test our proposal with different participants, in this scenario the obstacle has been a person wearing loose clothes, as can be seen in Figure 4.66. The person crosses in a perpendicular path performing a *"lane change"* manoeuvre, as can be seen in Figure 4.67.

(a) Loose chothes t_1 (b) Loose chothes t_2 (c) Loose chothes t_3

Figure 4.66: Scenario 7 - Images sequence.

Figure 4.68 shows the results in velocities detection, where it can be seen that the system detects this unusual appearance participant and his changes in velocities due to the *"lane change"* manoeuvre. Errors in velocities detection can be seen in Table 4.15, where our proposal detect the dynamic obstacle with an error about 7%.

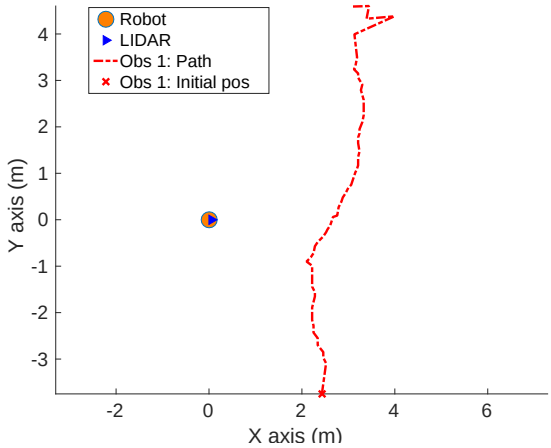


Figure 4.67: Scenario 7 - Results: Path detected.

Table 4.15: Scenario 7 - Errors in velocities detection

		V_x (m/s)	V_y (m/s)	$\|\|V\|\|$ (m/s)	θ (º)
Obs 1	$\overline{\epsilon}$	0.112	0.096	0.090 (7%)	5.02
	σ_ϵ	0.118	0.102	0.094 (7.1%)	5.54

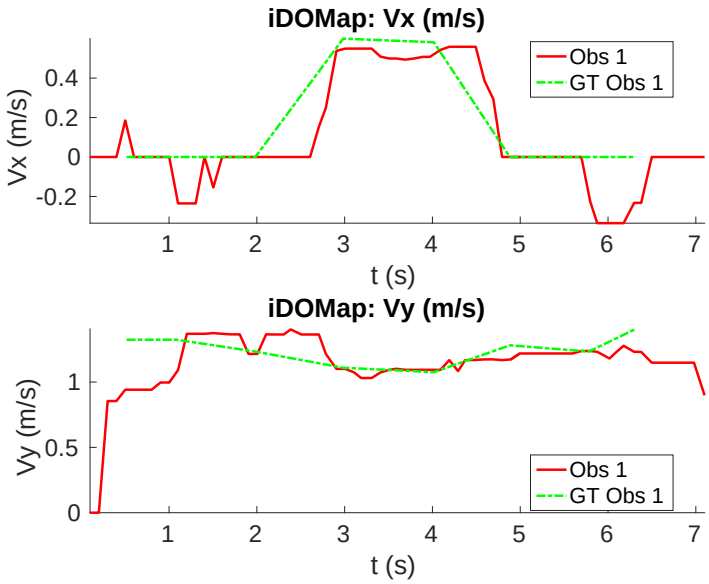


Figure 4.68: Scenario 7 - Results: Velocities detection.

- ***Scenario 8 - A robot crossing in a diagonal path:*** Continuing with the idea of test our proposal with different participants, in this scenario, a robot crosses in a diagonal path with a constant linear velocity of 0.5 m/s and in an angle of 18º respect to the robot with iDOMap. This scenario is bigger than the previous one, in order to see the obstacles more time. The Ground Truth in this case has been obtained taking into account the initial and final static positions, due to the obstacles maintain this position for 5 seconds. Then, hand-made centroid has been computed from the measures of LIDAR during the static period of time of the obstacle.

  Figure 4.69 shows the vector magnitude $||V||$ and angle (θ) of the velocity detection in order to clarifying the comparison with the Ground Truth. Table 4.16 shows the error in velocities detections, where it can be seen that the error velocity magnitude vector is lower that 0.04 m/s and the error in orientation is about 6º.

  The vector velocity magnitude percentage error $\bar{\epsilon}_{||V||}$ and σ_ϵ

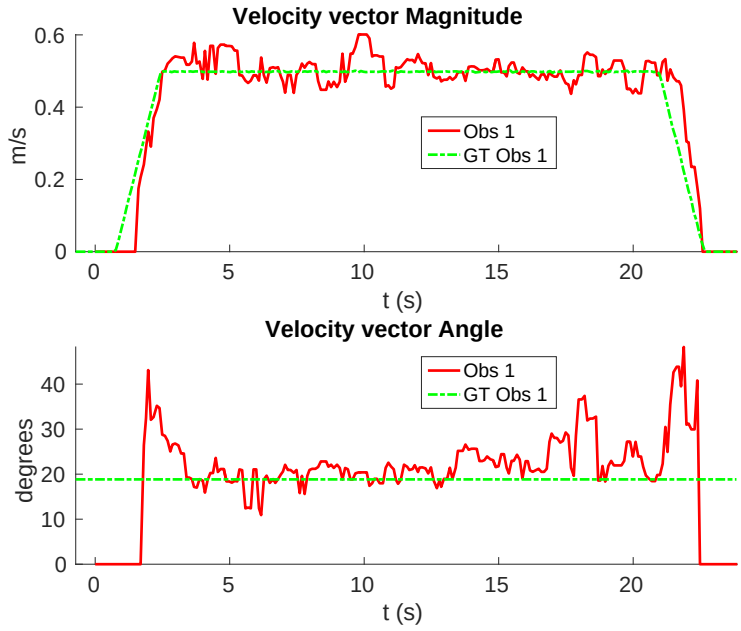


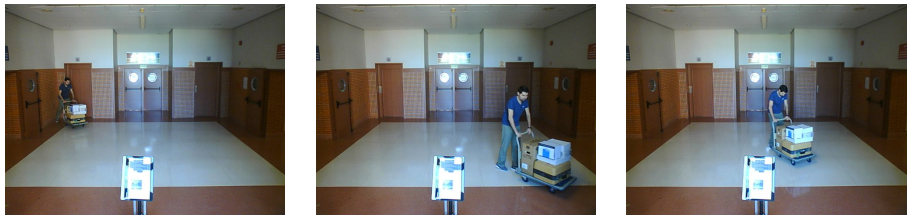
Figure 4.69: Scenario 8 - Results: Angle and Magnitude vectors.

Table 4.16: Scenario 8 - Errors in velocities detection

		$V_x\ (m/s)$	$V_y\ (m/s)$	$\|\|V\|\|\ (m/s)$	$\theta\ (^o)$
Obs 1	$\bar{\epsilon}$	0.037	0.033	0.031 (30.65%)	6.10
	σ_ϵ	0.039	0.033	0.038 (229.40%)	5.81

are the highest experimental errors due to when the Ground Truth velocities are very low (when the obstacle starts and ends move), the error detected (in percentage) is very high. In this case, if only dynamic obstacles are taking into account when their velocities are upper a threshold (for example 0.15 m/s) the error would be $\bar{\epsilon}_{||V||} = 7.67\%$ and $\sigma_\epsilon = 15.20\%$.

- ***Scenario 9 - Person pushing a trolley:*** Other participants with whom robots must coexist are people pushing objects, such as baby carriage or shopping cart. In order to test our proposal with this kind of obstacles, a person pushing a trolley has performed two diagonal paths with an stop between them. The scenario can be seen in Figure 4.70.



(a) Person starts (b) Person Stops (c) Backward path

Figure 4.70: Scenario 9 - Images sequence.

Figure 4.71 shows the detected paths followed by the person, in this case, the *red path* shows the forward path and the *yellow* one the backward path.

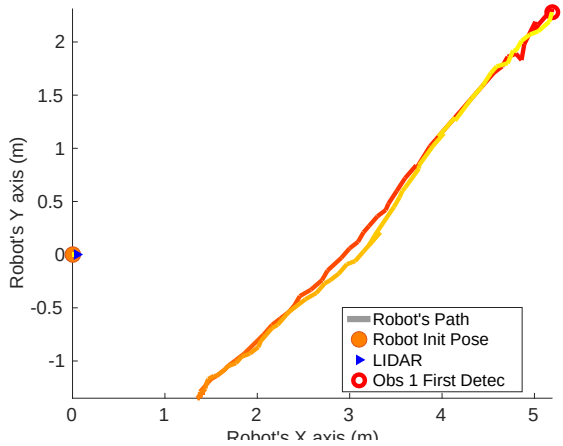


Figure 4.71: Scenario 9 - Results: Path detected.

Figures 4.72 and 4.73 show the velocities detection at each axis and in magnitude and angle vector respectively. It can be seen

that both paths (forwards and backwards) have been detected with an errors around 0.1 m/s, as it can be shown in Table 4.17.

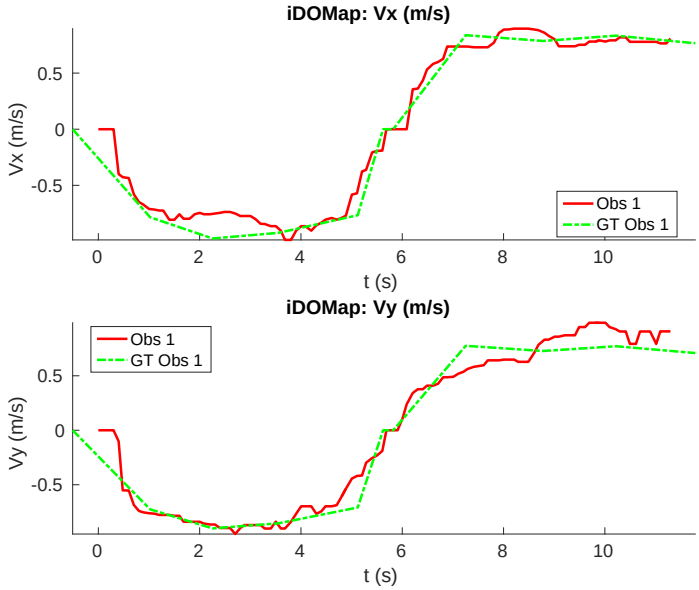


Figure 4.72: Scenario 9 - Results: Velocities detection.

Table 4.17: Scenario 9 - Errors in velocities detection

		V_x (m/s)	V_y (m/s)	$\|\|V\|\|$ (m/s)	θ (º)
Obs 1	$\bar{\epsilon}$	0.090	0.102	0.104 (21.01%)	13.26
	σ_ϵ	0.077	0.083	0.097 (75.3%)	20.82

Similar to the previous scenario, if only dynamic obstacles are taking into account when their velocities are upper a threshold (0.15 m/s) the magnitude velocity error would be $\bar{\epsilon}_{||V||} = 13.28\%$ and $\sigma_\epsilon = 19.42\%$.

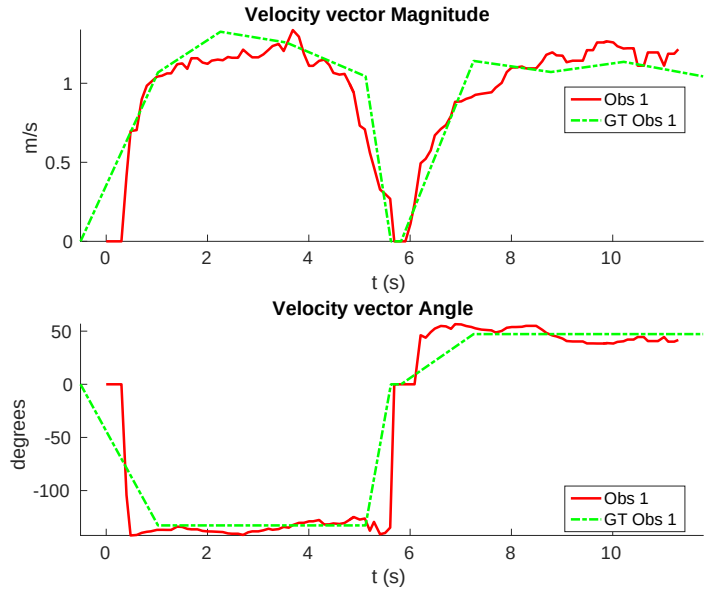


Figure 4.73: Scenario 9 - Results: Angle and Magnitude vectors.

After tests the proposal in different simulated and real scenarios, with different kinds of participants and velocities up to 3.5 m/s, the error in the magnitude vector of velocity ($\bar{\epsilon}_{||V||}$)has remained lower than 14%, only in some cases when the obstacles are thin and located laterally to the LIDAR the errors increase up to 21%. In addition, the error in orientation has remained lower than 13º.

4.6 Conclusions and contributions

In this chapter, an odometry improvement system has been presented in order to deal with odometry errors, that is especially important in indoors where other localisation systems are not always available, such as GPS.

Also a laser pose calibration system has been explained, making it possible to know the real pose of the laser when this kind of sensor are placed on-board the platform. This is an important fact to obtain a more accurate measures from these kinds of sensors when they should be translated and rotated based on the robot movements. In addition

a low cost 3D mapping system has been presented using a 2D LIDAR. Taking into account 3D environments increase the safety needed of the system, especially when the robot has to deal with structure barriers such as stairs or ramps in indoors or irregular ground in outdoors.

The BOF approach has been improved in our DOMap proposal in terms of efficiency, by adding a stage to detect the relative velocities of the obstacles without assuming discretized velocities. This extension overcome the restriction of detecting the velocities only when an obstacle changes from one cell to another, neither maintain a occupancy map for each discretized velocity, decreasing the computational costs. Also, the improved version iDOMap can handle occlusions between static and dynamic obstacles. The *"mixed points"* effect has been analysed. In addition, iDOMap is able to fuse environment information from different range sensors, if there are available, due to an improved stage that compute the occupancy grid and tracking stage are ready for this fusion. A results of the DOMap perception stage and its improved version iDOMap have been shown in order to detect dynamic obstacles in the surroundings of the robot, increasing the safety and reducing the energy consumption when the robot performs obstacle avoidance manoeuvres. Different participants have been tested in order to show the wider kind of obstacles than our proposal can handle, where others model-based approaches would need being trained or could fail.

Chapter 5

Application Results

This chapter presents the results of the application of previous chapters attending to the objectives of this thesis. As mentioned before, this thesis is part of ABSYNTHE and RoboShop projects. Application results of both projects in real scenarios, where the systems described in the previous chapters are involved, will be exposed in next sections.

In addition, this chapter also demonstrates the improvement in obstacle avoidance systems using the perception stage presented in this thesis.

5.1 Improvement of Obstacle Avoidance Algorithms

In this section, in order to show the improvement of the obstacles avoidance systems when the dynamic obstacles in the scene are taken into account by the proposed perception stage, some simulated and real experiments have been performed in different scenarios with different number of participants. At each scenario the system has been tested including the perception stage proposed and without it, that means taking the raw laser data as perception stage.

5.1.1 Obstacle Avoidance Algorithms

Classical obstacle avoidance algorithms introduced in Section 4.4.2 are able to deal with dynamic obstacles if the movements of the robot are significantly faster than the dynamic obstacles. However, avoiding dynamic obstacles like static ones can result in non optimal trajectories and risky situations, especially if the measurements of the environment is not accurate enough or the movement of the obstacles drastically changes. The obstacle avoidance algorithms that have been used in order to test the improvement in the avoidance manoeuvres, are the *Dynamic Curvature-Velocity Method (DCVM)* and *Dynamic Window For Dynamic Obstacles (DW4DO)*, proposed by [Molinos, 2017], that take into account the dynamic obstacles as follow:

- DCVM: is an extension of the static obstacle avoidance algorithm CVM. The idea of this algorithm is that a differential or non holonomical robot moves in circles, applying linear and angular velocities. This algorithm sectorises the environment into curvature sectors of the same distance free of obstacles in order to reduce the search space of possible paths to follow. Then, curvature arc is chosen based on a cost function that approaches the robot to the goal, rapidly and keeping it away from obstacles. DCVM extension checks the collisions with moving obstacles thanks to the modified local stage that is used. Also, the cost function is modified to avoid, not only the collision with moving obstacles, in addition crossing their predicted paths, increasing the safety of the robot.

- DW4DO: is based on the *Dynamic Window Approach (DWA)*. At each iteration only a set of velocities are reachable by the robot, so the dynamic and kinematic restrictions are taken into account. The collision checking is made in a similar way than DCVM does, and the algorithm is able to avoid dynamic obstacles and also avoid crossing their trajectories. To optimise the path, a two step *dynamic window* is executed, where the best path to the goal is calculated even if it is not reachable in the next iteration, and the algorithm tries to follow reachable

paths near the best one. Also, a term in the cost function prizes following similar velocities reducing the drastically changes of velocities and improving the stability and energy consumption of the robot.

5.1.2 Simulated Experiments

In these simulated experiments RoboShop has been used as robotic platform. The range sensor used is *SICK LMS 151* with 20 meters coverage and a FOV of 180º in simulation. The *Gazebo* 3D as simulator have been used and the moving obstacles has been simulated with other robot platforms in order to know their velocities and pose. Each scenario has been performed with the two obstacles avoidance algorithms introduced in Section 5.1.1 with raw LIDAR data and with the perception stage proposed. The system has been simulated in the scenarios as follow:

- ***Scenario 1 - Two obstacles crossing:*** in this scenario, the final goal is in front of the robot and two objects cross in perpendicular paths to the robot at 0.4 m/s. Figure 5.1 shows the starting position and orientation of the obstacles. Also it shows the path followed by the robot at each case. This scenario tries to simulate a risky environment where the robot could cross the trajectories of the moving obstacles.

  The paths followed (Figure 5.1) show that without knowledge of the obstacles velocities, both algorithms try to overcome the obstacles crossing in front of them, and the obstacles block the possible paths to the goal making the robot travels parallel to the first obstacle until this obstacle is far enough away to go to the goal. On the other hand, knowing the estimated velocities of the obstacles, both avoidance algorithms try to go to the goal overcoming the obstacle behind them in a safer manoeuvre. Also these two manoeuvres are more efficiency, due they are shorter in distance and time, as it is shown in Table 5.1.

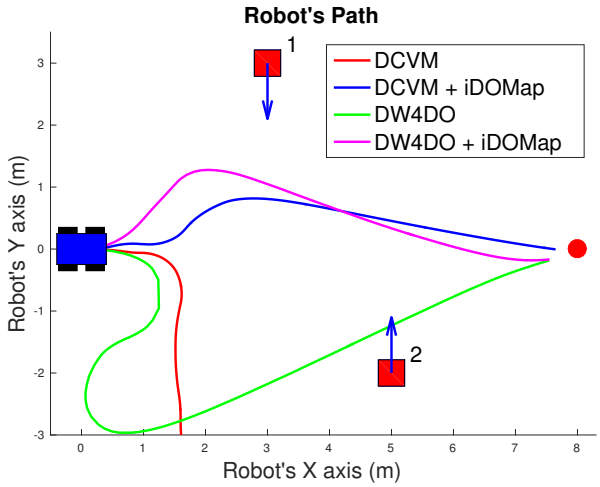


Figure 5.1: Scenario 1: Paths Results.

- ***Scenario 2 - Moving obstacle approaching the robot:*** this scenario is a more complex environment that mixes static and dynamic obstacles (Figure 5.2). Two static obstacles (*blue squares*) have been placed: a corridor has been form between the big obstacle (on the left of the robot) and the obstacle in front of it. Also another obstacle has been placed in the corner that forms another wider corridor. In addition there is one moving obstacle (*red square*) approaching to the robot in collision path to it through the first corridor. Therefore, a corridor that will be blocked by a moving obstacle has been simulated. This scenario has been selected to deal with this risky situation when the robot could be locked in a small space due to an obstacle is moving to the robot (*local minimum*).

  Figure 5.2 shows the static obstacles (*blue squares*) and the starting position and orientation of the dynamic obstacle (*red square*). Also, Figure 5.2 shows how without knowledge about the obstacle velocity, the DCVM tries to enter to the narrow corridor to later change the direction when the obstacles block its path. Even worst is the case of the DW4DO that starts loops trajectories. On the other hand, with our proposal, both

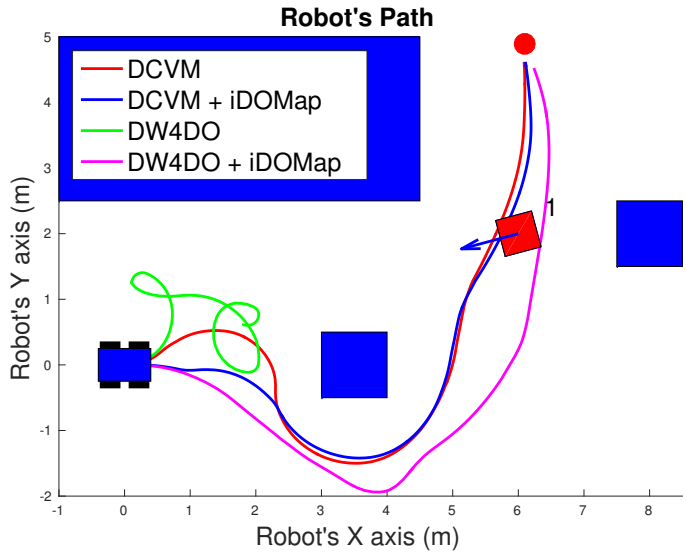


Figure 5.2: Scenario 2: Paths Results.

algorithms are able to reach the goal knowing that the narrow corridor will be blocked and avoiding this path. The paths followed with iDOMap are safer and more efficient than without it due to are shorter in time and distance as can be seen in Table 5.1.

- ***Scenario 3 - Lane changing:*** In this scenario moving obstacles travelling in the same direction as the robot, so it simulates a lane changing in a road. The obstacle nearer the robot travels at 0.75 m/s and the further one at $0.4m/s$.

  This scenario is similar to the previous one in term of riskiness, due to the robot could cross the dynamic obstacles trajectories. In the case of only take into account the raw LIDAR measurements, the DCVM spends a lot of time trying to overcome the obstacle travelling parallel to them. This situation occurs when the velocities of the obstacle and robot are similar, until the obstacle is far enough to avoid it. Similar is the case of the DW4DO without our proposed iDOMap. However, if the algorithm has the velocities estimation, both of them begin to

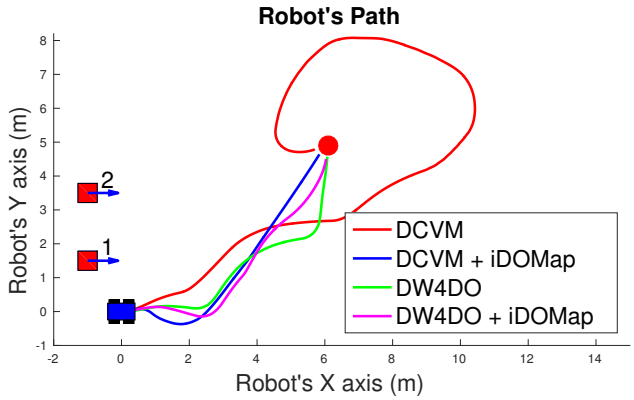


Figure 5.3: Scenario 3: Paths Results.

separate from the nearer obstacle (because it is safer) and go to goal crossing behind both obstacles in a safer and smoother paths, as show the time and distance travelled data for these cases in Table 5.1.

Table 5.1: Simulated experiments: Results.

Experiment	**Algorithm**	**d**	**t**	$\overline{v}$	σ_v^2	$\overline{\lvert\omega\rvert}$	σ_ω^2
Scenario 1: Two Obstacles Crossing	DCVM	–	–	–	–	–	–
	DCVM + iDOMap	**7.90**	**20.28**	**0.38**	**0.009**	7.93	**12.88**
	DW4DO	12.18	47.91	0.25	0.16	15.09	20.27
	DW4DO + iDOMap	**8.18**	**21.85**	**0.37**	**0.04**	6.93	**11.00**
Scenario 2: Approaching obstacle	DCVM	11.83	30.22	**0.39**	**0.001**	14.15	16.75
	DCVM + iDOMap	**11.06**	**28.30**	**0.39**	0.002	0.59	**12.18**
	DW4DO	–	–	–	–	–	–
	DW4DO + iDOMap	**11.84**	**32.47**	0.35	0.081	**7.41**	**10.79**
Scenario 3: Lane changing	DCVM	22.549	57.699	**0.39**	**0.001**	16.11	19.38
	DCVM + iDOMap	**8.43**	**22.90**	0.36	0.084	**8.97**	**15.00**
	DW4DO	8.76	22.94	**0.37**	**0.049**	**9.41**	13.54
	DW4DO + iDOMap	**8.42**	**22.787**	0.36	0.056	10.70	**13.52**

A summary of paths followed by all the combinations of both obstacle avoidance algorithms exposed in Section 5.1.1, with LIDAR raw data and iDOMap as perception stages, it is shown in Table 5.1.

The parameters shown in Table 5.1 measure velocity of the manoeuvre: distance travelled $d(m)$ and time spent $t(s)$ to reach the goal, also the medium linear velocity $\overline{v}(m/s)$, while others measure the smoothness of the path: medium absolute angular velocity $|\overline{\omega}|(^{o}/s)$ and the velocities standard deviation: σ_v^2 and σ_ω^2. Taking into account these parameters, it can be shown how the distance travelled and the time spent had been reduced in the three scenarios with our iDOMap. In addition, when our perception stage is available, the σ^2 is lower, meaning that smoother paths are followed. Also, in some risky situations, without velocities estimations, the goal can not be reach or the path followed is huge, as the case of DCVM in the Scenario 1 and the DW4DO in the Scenario 2. For that reason, taking into account the velocities estimation provided by our proposal reduces the time spent and the distance travelled to reach the goal, moreover the performed path is smoother and safer.

5.1.3 Real Experiments

In these real scenarios it has been used the best combinations obtained in simulation. Due to in this case the dynamic obstacles are persons, the exacts positions and velocities can not be controlled in a accurate way, increasing the difficulty to make repeatable and comparable real scenarios. The real experiments have been performed in a controlled indoor environment in the *Polytechnic School* at the *University of Alcalá (UAH)*. The area of the environment is 6x12 metres approx, where there are several doors, corridors and columns (Figure 5.4).

Taking these premises into account, some real scenarios have been performed:

- ***Scenario 1 - A person overtaking the robot:*** In this scenario the robot has to reach a goal in $8m$ forward from its initial position (*red dot* in Figure 5.6), while a person (marked as *red square* in Figure 5.6) overtakes the robot on the left.

  The paths followed by the robot at each combination of perception and obstacles avoidance algorithm are shown in Figure 5.6. Figure 5.7 shows the detected dynamic obstacle at each case,

Figure 5.4: Test Bed: Indoor scenario.

(a) Test Bed t_2 (b) Test Bed t_2

Figure 5.5: Scenario 1 - Images sequence.

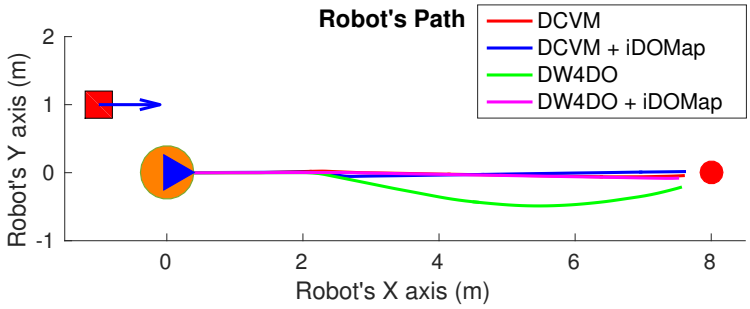
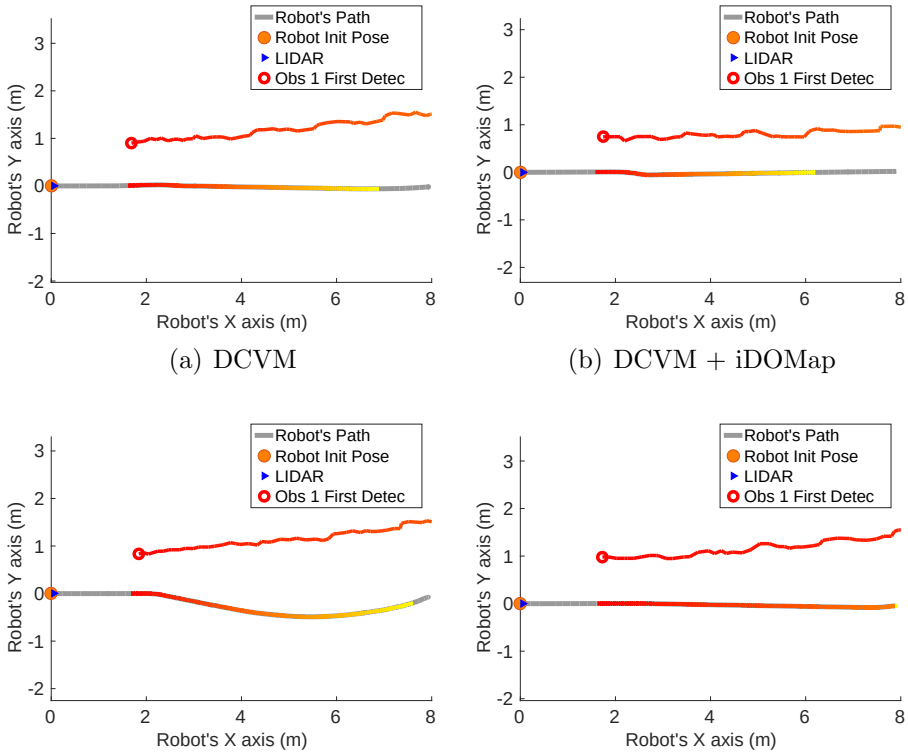


Figure 5.6: Scenario 1 - Robot's Paths.

where the *grey* line represents the robot path. The colour of the obstacle's path and the robot's path match (from *red* to *yellow*) during the period of time that our proposal detect the obstacle, in order to show them in a synchronized way, the pose



(a) DCVM (b) DCVM + iDOMap

(c) DW4DO (d) DW4DO + iDOMap

Figure 5.7: Scenario 1 - Robot's Paths and iDOMap Detections

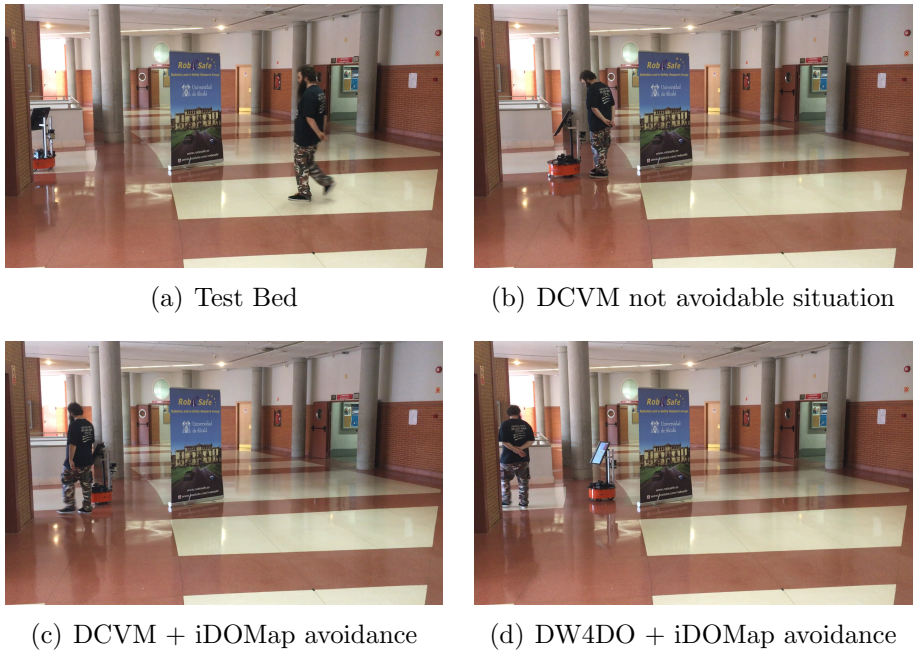
of the robot where the obstacle is detected at each step.

Figure 5.13 has been limited to 8m, in order to show it in a clarifying way, although the obstacle has continue been detected.

Evaluating four cases in the same way, assuming that they can be comparable, Figure 5.7(b) shows the bit difference when an estimated velocities are present in the case of the DCVM, the algorithm moves away a bit (to the right) respects to the case without velocities estimations (Figure 5.7(a), increasing the safety. Figures 5.7(c) and 5.7(d) shows other behaviour performed by DW4DO; without velocities estimation (Figure 5.7(c)) when the obstacle appears in the robot's map

near to the robot, suddenly it tries to move away of the obstacle, on the contrary, if iDOMap is available (Figure 5.7(d)), the velocities of the obstacle are estimated, then the algorithm estimated that the overtaking of the dynamic obstacle do not influence in its trajectory, keeping the direction, increasing the energy efficiency of the path (due to it do not changes its velocities and the path is shorter).

- ***Scenario 2 - Moving obstacle approaching the robot:*** This scenario (Figure 5.8(a)) is similar to the Scenario 2 in Section 5.1.2. This scenario has been selected in order to test a risky situation when the robot could be locked in a small space, between the wall in the left part of Figure 5.8(a) and the *auxiliary wall* located in the middle of Figure 5.8(a), due to an obstacle is moving to the robot (*local minimum*).



(a) Test Bed (b) DCVM not avoidable situation

(c) DCVM + iDOMap avoidance (d) DW4DO + iDOMap avoidance

Figure 5.8: Scenario 2 - Images sequence.

Figure 5.10 shows the detected dynamic obstacle at each case,

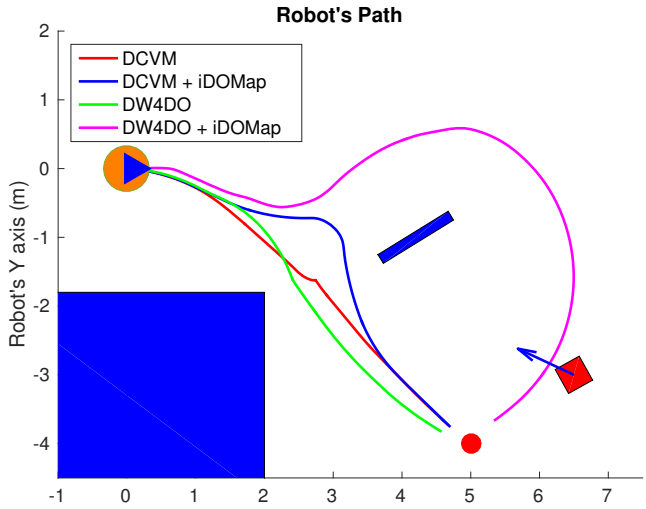


Figure 5.9: Scenario 2 - Robot's Paths.

where the *grey* line represents the robot path. The colour of the obstacle's path and the robot's path match (from *red* to *yellow*) during the period of time that our proposal detects the obstacle, in order to show in a synchronized way, the pose of the robot where the obstacle is detected at each step. Figure 5.10 has been limited to manoeuvre area, in order to show it in a clarifying way, for that reason the obstacle first detection is outside of the figure.

Figure 5.9 shows the behaviour of both algorithm without velocities estimation and with our proposal. In the case of the DCVM when the velocities are not available, the robot tries to pass trough the corridor and when the obstacles *"appear"* (Figure 5.8(b)) is too late to avoid it, then the robot stops and spins (position 3, -1.5m approx) being a risky situation. That spins manoeuvre and that the obstacle is too close to the robot, make the detection inaccurate at this point (Figure 5.10(a)). When iDOMap is combined with the DCVM (*blue path*) the robot avoid it to the left (Figure 5.8(c)), reducing its velocity and then continue to the goal. In the case of the DW4DO without our proposal is similar to the DCVM due to is a *"not avoidable*

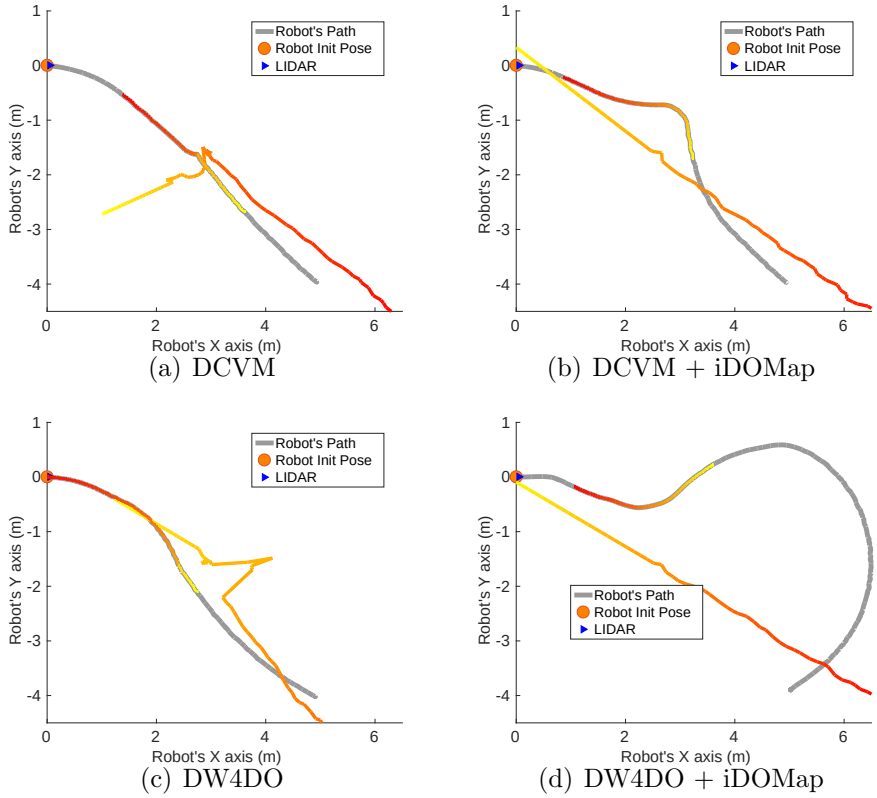


Figure 5.10: Scenario 2 - Robot's Paths and iDOMap Detections.

situation" in both cases. On the contrary, with DW4DO and iDOMap the algorithm is able to predict that the obstacle will be blocking the pass trough the corridor, avoiding this situation (Figure 5.8(d)) an reaching the goal in a longer but safer path (*pink path*).

- ***Scenario 3 - Lane changing:*** In this scenario the robot has to reach a goal in $7m$ forward and $4m$ to right from its initial position (*red dot* in Figure 5.12), during the manoeuvre a person (marked as *red square* in Figure 5.12) overtakes the robot in a parallel path. This scenario is comparable with the Scenario 3 in Section 5.1.2. The paths followed by the robot at each combination of perception and obstacles avoidance algorithm are

shown in Figure 5.12. Figure 5.13 shows the detected dynamic obstacle at each case, where the *grey* line represents the robot path. The colour of the obstacle's path and the robot's path match (from *red* to *yellow*) during the period of time that our proposal detects the obstacle, in order to show in a synchronized way, the pose of the robot where the obstacle is detected at each step.

(a) Test Bed t_1 (b) Test Bed t_2

Figure 5.11: Scenario 3 - Images sequence.

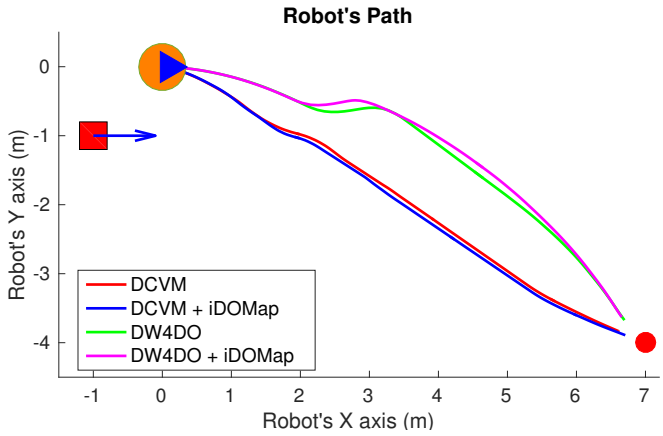
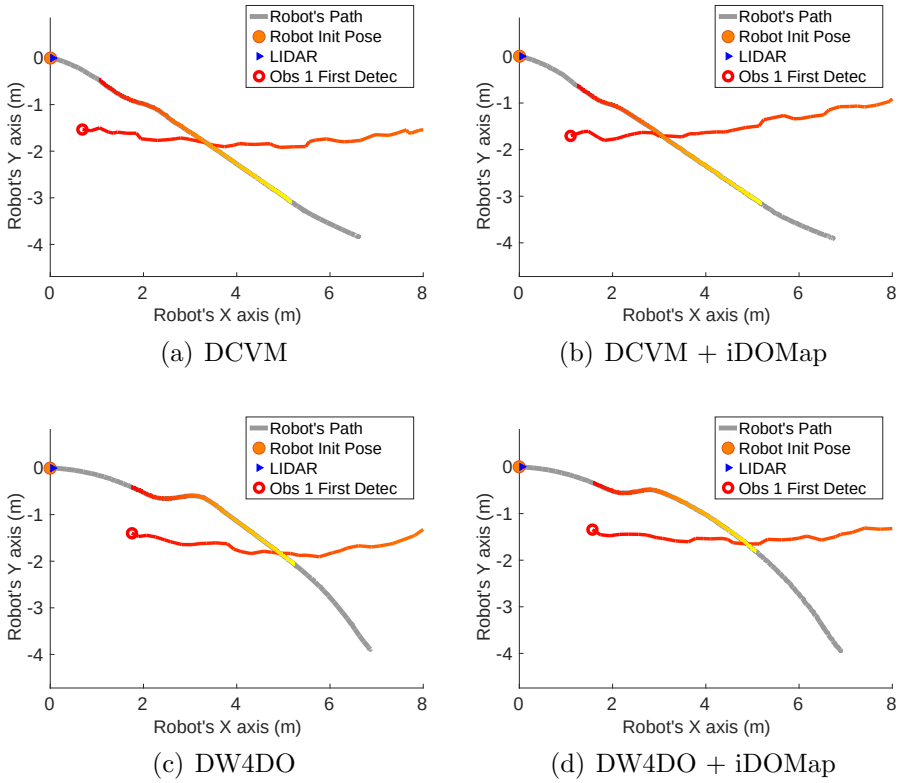


Figure 5.12: Scenario 3 - Robot's Paths.

Figure 5.13 has been limited to 8m, in order to show it in a clarifying way, although the obstacle has continue been detected.

Evaluating four cases in the same way, assuming that they can be comparable, Figure 5.12 shows that both obstacle avoidance



(a) DCVM (b) DCVM + iDOMap

(c) DW4DO (d) DW4DO + iDOMap

Figure 5.13: Scenario 3 - Robot's Paths and iDOMap Detections.

algorithms move away from the obstacle, DCVM to the right and DW4DO to the left, being in both safer manoeuvres.

Although the results are not comparable at the same level, due to they are not completely replicable, Table 5.2 shows the numerical results, in the same way that in simulated results (Section 5.1.2) in order to give an idea of the performance. The parameters shown in Table 5.2 measure velocity of the manoeuvre: distance travelled $d(m)$ and time spent $t(s)$ to reach the goal, also the medium linear velocity $\overline{v}(m/s)$, while others measure the smoothness of the path: medium absolute angular velocity $|\overline{\omega}|(^{o}/s)$ and the velocities standard deviation: σ_v^2 and σ_ω^2. In the Scenario 1 it can be seen that the case of DW4DO with iDOMap spends less time and travels less distance due

Table 5.2: Real experiments: Results

Experiment	Algorithm	d	t	$\overline{v}$	σ_v^2	$\overline{\lvert\omega\rvert}$	σ_ω^2
Scenario 1: Person overtaking	DCVM	7.61	19.30	0.39	0.043	0.97	2.25
	DCVM + iDOMap	7.63	19.34	0.39	0.045	1.67	4.55
	DW4DO	7.62	19.64	0.38	0.046	2.47	3.92
	DW4DO + iDOMap	7.52	18.96	0.39	0.043	0.71	1.53
Scenario 2: Approaching person	DCVM	6.15	18.52	0.33	0.134	9.06	14.33
	DCVM + iDOMap	6.63	17.40	0.38	0.062	9.75	14.12
	DW4DO	6.18	18.29	0.33	0.127	7.29	8.30
	DW4DO + iDOMap	10.71	28.20	0.37	0.054	10.95	11.90
Scenario 3: Lane changing	DCVM	7.68	21.55	0.35	0.120	4.99	7.98
	DCVM + iDOMap	7.80	21.99	0.34	0.122	5.10	8.56
	DW4DO	7.99	20.96	0.37	0.053	6.81	10.73
	DW4DO + iDOMap	7.96	20.89	0.37	0.057	6.22	9.18

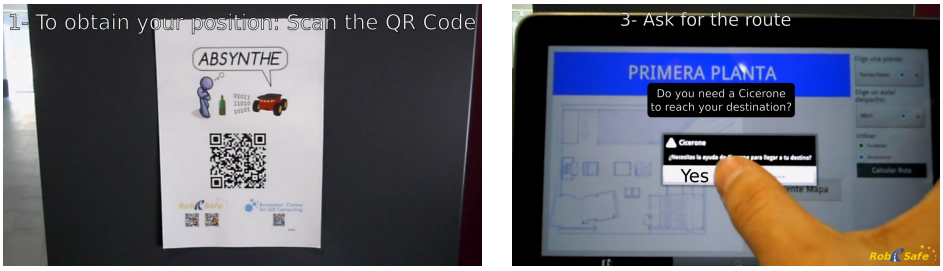
it keeps its trajectory as has been explained before. In the Scenario 2 both obstacles avoidance algorithms travel longer paths, but in a safer way, when iDOMap is present. Finally, in the Scenario 3, although the numerical results are very similar, in Figure 5.12 it can be seen that using our proposal, the paths are safer, as has been explained before.

These real experiments and the simulated ones exposed in Section 5.1.2 show that including the proposed perception stage increases the performance of the obstacle avoidance algorithms, helping to overcome *"not avoidable situations"*. These situation are more often when do not take into account the dynamics of the obstacles. This performance increasing, as it has been explained before, in some cases means to improve the safety of the paths and in other cases reduces the energy consumption.

5.2 ABSYNTHE Project Application

This project, where this thesis has been developed, aims to develop tools that can be used for human-robot teams in order to promote collaboration among them in indoor and outdoor environments. Collaboration between individuals is usually based on a specific task regarding the position of humans and robots. The interaction of

robots with their surroundings requires the acquisition and representation of objects contained within their immediate neighbourhood. Also is crucial that the robots are able to deal with these dynamic environments where are different kinds of participants with different dynamics. For that reason is necessary that the robots detect this dynamics no matter what type of object it is. In order to test this project a *Cicerone* application has been proposed through this thesis and the thesis of Eduardo Molinos [Molinos, 2017]. In this application several agents are involved: human, data stored in remote machines and heterogeneous robots.

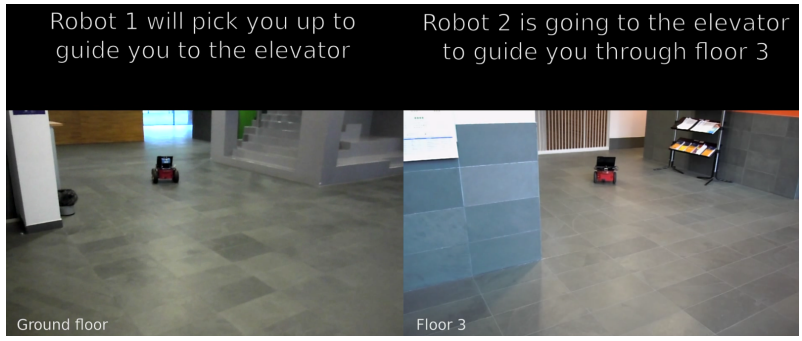


(a) ABSYNTHE QR Codes (b) ABSYNTHE App

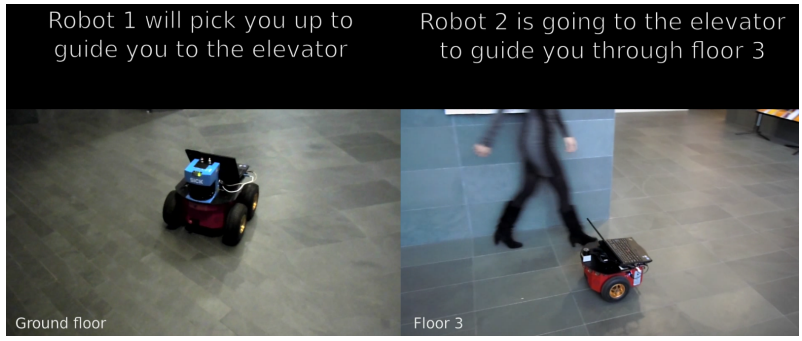
Figure 5.14: ABSYNTHE *Cicerone* Application.

The *Cicerone* application communicates with the human users via *QR codes* (Figure 5.14(a)) that are set in interest points inside a building, and it can be read by a mobile device application (Figure 5.14(b)). Each code is linked to a *webpage* where the current position of the user is stored. In this application the user can search how to go to an interesting point in the building, selecting the goal (for example, a room into the building) and some settings (for example, use the elevators or the stairs). The path to goal is calculated in a remote computer, where discrete positions of the building representing the topological relevant information are stored, and a Dijkstra algorithm is executed to obtain the shortest path that satisfies the restrictions given by the user.

Once the path has been received by the user application, there is a possibility to call a *Cicerone* to guide him to the goal. That *Cicerone* is an autonomous robot team. Each robot in the environment is



(a) ABSYNTHE App: Robots goes towards the user



(b) ABSYNTHE App: Robots continues towadrs the user

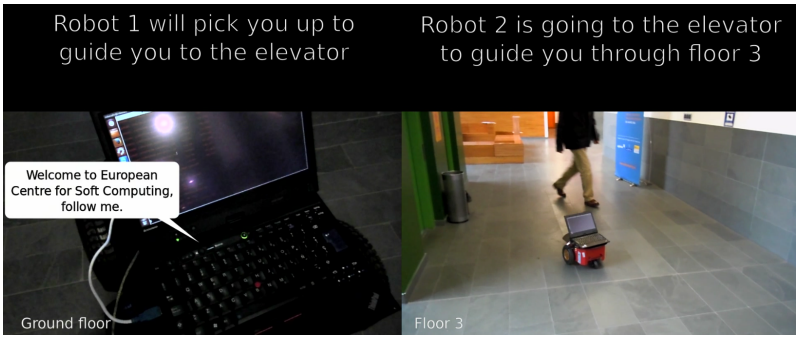
Figure 5.15: ABSYNTHE *Cicerone* Application.

connected to a central database, and a task assignment algorithm decides which robot needs to move to help user, based in different criteria like distance to the user, capabilities of the robot (not all robots can reach all points in the building) and if the robot is now free or occupied.

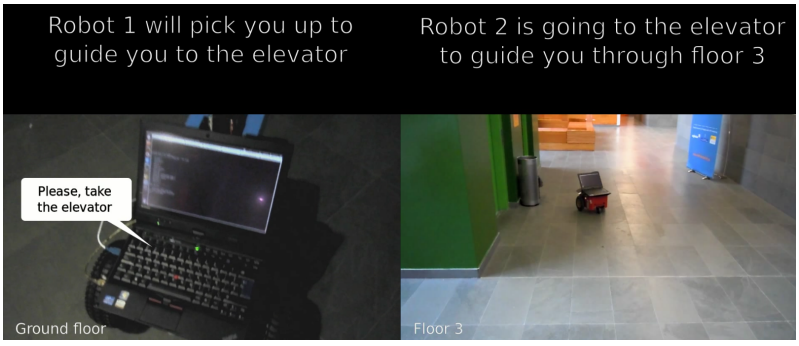
This system works with heterogeneous robots, each one with different capacities. In the demonstration two different robots are used: a Pioneer 3-AT with off-road capabilities and a *Sick LMS200* LIDAR with 18 m coverage (can be seen in the left part of Figure 5.15(b)) placed on the basement floor of the building and a Pioneer 3-DX without off road capabilities, with a *Hokuyo URG-04LX* LIDAR (can be seen in the right part of Figure 5.15(b)) with 5 m coverage placed on the third floor. Both robots execute the same localisation and

navigation algorithms adapted to their capabilities.

As the user has been set the goal in a room in the third floor, the path is calculated using the elevator and a new *Cicerone* has been called. As there are one robot on each floor, the task dispatch system orders the basement robot (on the left part of Figure 5.15(a)) to move towards the user and the third floor robot (on the right part of Figure 5.15(a)) moves to the elevator waiting for the user to reach that point. When the basement robot reaches the user position, it plays an audio message welcoming the user (On the left part of Figure 5.16(a)), and goes to the elevator slowly as the user can follows it.



(a) ABSYNTHE App: Robot 1 welcoming the user, Robot 2 waiting



(b) ABSYNTHE App: Robot 1 giving instructions, Robot 2 waiting

Figure 5.16: ABSYNTHE *Cicerone* Application.

When the robot reaches the elevator another audio message tells the user the destiny floor (on the left part of Figure 5.16(b)). The

messages and the situations when each message must be executed are also stored in the remote database. Once the user reaches the third floor, the robot is waiting in front of the elevator with a message in the tactile screen (on the right part of Figure 5.16(b)). When the user touches the screen the robot goes to the final destination. Finally, when the room is reached the robot plays another message thanking the user (Figure 5.17).

Figure 5.17: ABSYNTHE App: Robot 2 reaches the goal.

As the project finalised before the end of this thesis, the demonstration has not been made with the final proposed system. The localisation system is based on AMCL filter, navigation and perception proposal are achieved using preliminary versions of DCVM and DOMap algorithms, allowing the system to deal with dynamic participants in the scene (can be seen the right part of Figure 5.15(b) and Figure 5.16(a)) and global path planning has been achieve using a Dijkstra algorithm. In order to limit the movement of each robot on the building, the maps that robots use for path planning are different and they are stored in the remote server. Even if more than one robot shares the same environment, its maps can be different in order to adapt to its capacities (for example, going through sand parts of the environment if the robot has off-road capabilities or not). In the demonstration, each robot is limited to movement in one floor (basement and third floor).

The video of this demonstration can be watched in the RobeSafe Research Group youtube channel [1] and was perfomed in the *European*

[1] https://www.youtube.com/watch?v=lD44C1WyR58&t=9s

Centre For Soft Computing in Mieres (Asturias). This demonstration shows the possibilities of the collaboration between human and robots, and how a full navigation framework, that takes into account the movement of the dynamic obstacles (like the user following the robot) is needed.

5.3 RoboShop Project Application

This thesis was also developed inside the RoboShop project. In this project, our proposed platform PROPINA was improved to work as a *shop assistant*. To demonstrate the feasibility of the proposal, tests under real conditions in the *Juguetrónica* shop in Madrid were performed.

The *Juguetónica* scenario is a real challenging one for so many reasons: it has very few empty space for moving with the robot, it is usually crowded of people, the objects in the environment have very irregular forms, there are bright lights and shelves and doors made of glass are present.

In order to communicate with the users, the robot is equipped with a tactile interface where an HMI helps interacting with the user (shown in Figure 5.18), including audio messages. As a future work it is proposed to be also responsive for audio commands. In that screen a database of the sections and products of the store is present. When the user selects a product to buy, the robot will go along the user in order to show where the product is and giving additional information about it. All the positions within the map where the products are and its information are stored in a database. The navigation of the robot was fully autonomous.

As the system can be deployed in different scenarios, the first step to achieve is building a map for localisation and path planning. This process can be seen in Figure 5.19(a). The localisation is achieved using a fusion with AMCL (built using only the LIDAR and not the other sensors), dead reckoning odometry and IMU measurements, fused using an EKF filter as has been explained before. Even with this localisation system, in that challenging, small and dynamic scenario the localisation can fail. In order to correct this possible failures *QR*

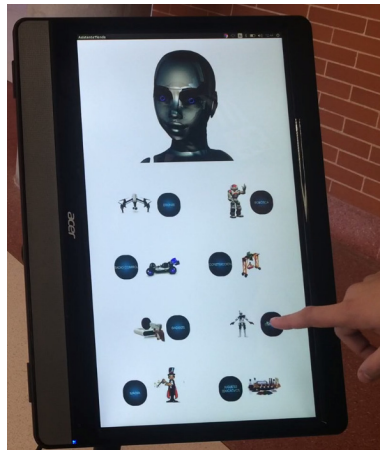
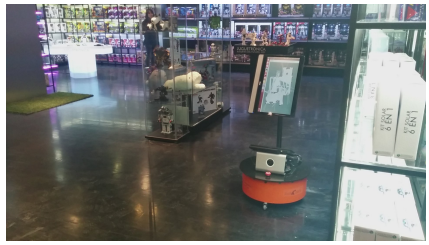


Figure 5.18: RoboShop HMI.

codes are situated in key points of the environment in order to be detected by the two cameras of the robot (one pointed forward and one pointed to the ceiling). When an *QR code* is detected the robot can correct its position if it was erroneous.



(a) Mapping

(b) Navigation

Figure 5.19: RoboShop Project demonstration.

As this project ended before the ending of this thesis, the navigation framework used is not the final one. The perception stage was performed by previous version of iDOMap taking into account sonar and LIDAR measurements, due to the presence of glass in the environment Also a previous version of DW4DO algorithm by [Molinos, 2017] along a Dijkstra global planner is used.

The maps used for localisation and global path planning are different. There are obstacles where the robot can collide but are not detected by the LIDAR and sonar sensors due to its range and height limitations. These obstacles are inserted in the planning map avoiding the robot goes near them. Also, areas where the robot must not go are neglected in the map.

These tests demonstrate the feasibility of this autonomous shop assistant platform in a real challenging scenario (Figure 5.19(b)). Full video of this tests can be watched on the RobeSafe Research Group youtube channel [2].

[2]https://www.youtube.com/watch?v=H2ZsfAPaDr0

Chapter 6

Conclusions and Future Work

The goal of this thesis is to develop a improved perception stage for UGVs in order to achieve a robust navigation in terms of safety and energy saving. The LIDAR sensor has been used as main sensor in this proposal due to its high accurate in range measurement and that it can be use in indoors and outdoors, depends on the sensor version. In indoors scenarios the main drawback is not knowing the exact pose of the robot, when localisation systems are not available. For that reason an odometry improvement has been presented using the information provided by an IMU.

A developed robotic platform has been presented and compared with some of the usual commercial platforms used in research. This developed platform has been designed taking into account aspects as safety and modularity and it has been reached improvements, such as, higher payload, improved odometry and increase in safety in scenarios where the platform has to collaborate with humans and robots teams.

The proposed *Grid-Based* perception stage is based on BOF and extended to not discretized velocities. This extension overcome the restriction of detecting the velocities only when an obstacle changes from one cell to another, neither maintain a occupancy map for each discretized velocity. A spatial memory has been added to handle the occlusions when the dynamic obstacles occlude static part of the environment. These features help to the planning algorithm to avoid

computing paths through static obstacles when LIDAR does not see them leading an obstacle to *"disappearing"*. A laser grid discretisation has been proposed in order to filter the measurement error in the LIDAR data. The movement detection stage is based on Optical Flow applied to the laser measurements to detect the obstacle velocities. A dynamic occlusion detector has been added to the system to deal with these situations. In addition, a PF algorithm has been used to track multiple hypothesis and to deal with the noise in the measurements. The number of particles has been fixed to limit the computational complexity.

The proposed perception stage has been tested in several simulated and real scenarios with different obstacles avoidance algorithms, even though they were not designed for dynamic environments. The focus of these tests have been designed to show how the velocities detection of the dynamic obstacles improves the performance of the obstacle avoidance algorithm, especially in risky situations.

The remaining of the chapter presents the main contributions introduced and developed along this thesis. Finally, the future lines of research left open by this thesis will be drawn.

6.1 Main Contributions

From the results obtained in the previous chapters, the main contributions of this thesis are as follows:

1. **Commercial Robotic Platforms Comparison:** Some of the commercial robotic platforms commonly used in research have been analysed and compared with our developed platform PROPINA. Their capabilities, strengths and weaknesses have been shown in order to know which platform is the best election for each task, useful when there are several platforms to solve different challenges to efficiently cooperate.

2. **Development of a Robotic Platform:** Several improvements have been introduced in the developed platform PROPINA and its extended version RoboShop, that are suitable for indoors, such as the more accurate odometry, higher

payload and modular and safer design taking into account that the robotic platforms should share scenario with humans.

3. **Odometry improvement proposal:** It has been shown an odometry improvement of a robotic platform in real indoors scenarios using a general purpose IMU and one embedded in a mobile device. This is especially important in indoors scenarios where GPS is not available and localization is the base of higher level algorithms.

4. **Laser pose calibration proposal:** A pose calibration method for laser sensors has been proposed. The task to know the exact sensor pose is crucial to translate the measures from the sensor to other coordinates system in the most accurate way. An application of this calibration method in 3D map building with a 2D laser has been presented.

5. **Perception Stage Improvement proposal:** The proposed DOMap has improved the BOF approach in terms of efficiency, by adding a stage to detect the relative velocities of the obstacles without assuming discretized velocities. The velocities detection has been obtained using Optical Flow over a discretized grid of the LIDAR measurements, even though the Optical Flow was not originally designed for laser measurements. In addition, an improved version of our algorithm has been proposed iDOMap. It is able to fuse environment information from different sensors, if there are available, due to the stage that computed the occupancy grid and tracking stage are ready for this fusion. Also, the proposal can deal with the occlusions. Taking into account obstacles dynamics increases the safety and reduce the energy consumption when the robot performs obstacle avoidance manoeuvres.

6. **Application:** In order to test the projects where this thesis is framed two applications were developed. In the case of ABSYNTHE project, a *ciceron* application was proposed and in RoboShop a robotic shop assistant application was developed.

Both cases have been tested in real environments, demonstrating that uses an improved perception stage with velocities estimations improves the routes provided by several local obstacle algorithms, even when they have not been designed for dealing with dynamic obstacles.

6.2 Future work

From the results and conclusions of the present work, several research lines can be faced:

1. **Extend the system to 3D:** In order to deal with possibly available 3D features in the environment, an extension to 3D mapping will be addressed. The first step will be the extension to the system in several layers in *Z axis* where the 2D grid becomes a 3D grid of VOXELs. This approach can be done taking the data from several 2D LIDAR located in different heights or with a 3D LIDAR discretizing their measures in VOXELs.

2. **Filter *mixed pixels* in LIDAR measurements**: Detecting and filtering this phenomenon would be interesting due to help to improve the detection of obstacles when they are close to other elements of the environment. Also, it would help to avoid to close narrow spaces. On the other hand, it is a challenging and risky task due to erase real laser impacts would decrease the safety of the avoidance.

3. **Adaptative grids:** Taking into account the scenario, to adapt the cell size in occupancy and laser grid could be improve the performance:

   Occupancy cells: could be vary their size based on the environment state, such as the global occupancy, the number of dynamic obstacle, etc.

   Laser grid: based on the LIDAR measure error, due to this one increases with the distance, and in order to filter it, the size of the laser grid should increase with the distance according to this error.

4. **Scan Matching alternatives**: There are several scan matching alternatives, such as ICP, than can be applied in environments with enough translational and rotational invariant features. For that reason, the matching accuracy and the spend time of the algorithm depend of this amount of features. In the case of new environments, crowed, without lineal neither orthogonal features; it makes difficult that these methods work properly or the computational cost rises. Other approaches take the risk of shorten the linear spaces, such as corridors. Also, scan matching approaches usually reject points that were not in the previous instant. Due to the dynamic changing environments that are the focus of this thesis make this approaches inappropriate, a priori, for dynamic obstacles mapping. Nonetheless, an interesting future work would be to evaluate and improve these techniques in order to including the laser reflectivity in them, since these approaches originally do not usually take into account this feature.

5. **Ground Truth improvement:** To develop a system that tracks the UGV and the objects in the dynamic surrounding based on an external system, such as an intelligent environment, allowing to obtain accuracy numeric results of the DATMO systems. One possible approach could be based on high frame rate cameras, placed on the ceiling perpendicular to the floor. With enough height camera location and FOV, could be possibly identify and track the objects from the zenith image.

6. **Test the proposal in different environments:** Although the proposal has been tested in different simulated and real environments with different platforms, it would be interesting to test it in others environments and other robotic platforms. For that reason, we are trying to test the proposal with the *Pepper* robot by *SoftBank Robotics* [1] in the *Juguetronica* environment (with whom we have previously collaborated), due to this platform is available at this company.

[1] www.ald.softbankrobotics.com

7. **iDOMap Parameters Selection:** In order to identify the best configuration parameters of the iDOMap based on the scenario where the robotic platform has to move. This selection could be based in scenario features such as its dimensions, the *"cluttered level"* or the *"crowded level"*, and the robotic platform equipment available. This parameters selection would help to further generalize the proposal.

8. **Publish iDOMap as ROS package:** In order to test and improve our proposal, we will publish it as ROS package, to be available for the research community.

Appendices

Appendix A

Publications Derived from this PhD Dissertation

A.1 Journal Publications

2014 **Perception and Navigation in Unknown Environments: The DARPA Robotics Challenge**, *E. Molinos, Á. Llamazares, N. Hernández, R. Arroyo, A. Cela, J.J. Yebes, M. Ocaña, L.M. Bergasa*, Advances in Intelligent Systems and Computing (ISSN: 2194-5357), Vol. 253, Pages 321-329.

2014 **Competing in the DARPA Virtual Robotics Challenge as the SARBOT Team**, *E. Garcia, M. Ocaña, L.M. Bergasa, M. Ferre, M. Abderrahim, J. Arevalo, D. San-Merodio, E. Molinos, N. Hernández, Á. Llamazares, F. Suarez, S. Rodriguez*, Advances in Intelligent Systems and Computing (ISSN: 2194-5357), Vol. 253, Pages 381-396.

2013 **Dynamic Obstacle Avoidance Using Bayesian Occupancy Filter and Approximate Inference**, *Á. Llamazares, V. Ivan, E. Molinos, M. Ocaña, S. vijayakumar*, Sensors (ISSN: 1424-8220), Vol. 13(3), pages 2929-2944.

2013 **Comparison of Local Obstacle Avoidance Algorithms**, *E. Molinos, J. Pozuelo, Á. Llamazares, M. Ocaña, J. López*, Lecture Notes in Computer Science (ISSN: 0302-9743), Vo. 8112.

2011 **3D Map Building using a 2D Laser Scanner**, *Á. Llamazares, E. Molinos, M. Ocaña, L. M. Bergasa, N. Hernández and F. Herranz*, Lecture Notes in Computer Science (ISSN:0302-9743), vol 6928, p. 413-420.

A.2 Conference Publications

2016 **Mejora de la Odometría de un robot móvil aplicando medidas inerciales**, *R. Pintor, Á. Llamazares, E. Molinos, M. Ocaña*, Seminario anual de automática, electrónica industrial e instrumentación (SAAEI 16), Elche (Spain).

2015 **Cooperative Adaptive Cruise Control for a Convoy of Three Pioneer Robots**, *F. Martin, V. Milanés, M. Ocaña, E. Molinos, Á. Llamazares*, ROBOT'2015: Second Iberian Robotics Conference, Advances in Robotics. Springer ed. 2015, pp. 217-230.

2014 **Dynamic obstacle avoidance based on curvature arcs**, *E. Molinos, Á. Llamazares, M. Ocaña, F. Herranz*, IEEE/SICE International Symposium on System Integration (IEEE/SICE SII 2014), Tokyo (Japan).

2014 **Development of a Navigation System for a Robotic Shop Guide**, *M. Ocaña, Á. Llamazares, E. Molinos, N. Hernández, F. Herranz, P. Revenga, E. López*, XV Workshop of Physical Agents (WAF2014), Leon (Spain).

2014 **Integrating ABSYNTHE autonomous navigation system into ROS**, *Á. Llamazares, E. Molinos, M. Ocaña, F. Herranz*, IEEE ICRA 2014. Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots, Hong Kong (China).

2012 **Dynamic obstacle avoidance minimizing energy consumption**, *Á. Llamazares, V. Ivan, M. Ocaña and S. Vijayakumar*, IEEE Intelligent Vehicles Symposium Workshops, Alcalá de Henares (Spain).

A.3 Publications Partially related with this PhD Dissertation

A.3.1 Journal Publications

2014 **WiFi SLAM algorithms: an experimental comparison**, *F. Herranz, Á. Llamazares, E. Molinos, M. Ocaña, M. Sotelo*, Robotica (ISSN: 1469-8668).

2012 **Extended Floating Car Data System: Experimental Results and Application for a Hybrid Route Level of Service**, *J. J. Vinagre Díaz and D. Fernandez Llorca and A. B. Rodríguez González and R. Quintero Minguez and Á. Llamazares Llamazares and M. Á. Sotelo*, IEEE Transactions on Intelligent Transportation Systems (ISSN:1524-9050), vol 13, núm 1, p. 25-35.

2011 **Enhanced WiFi Localization System Based on Soft Computing Techniques to deal with Small-scale Variations in Wireless Sensors**, *J.M. Alonso, M. Ocaña, N. Hernández, F. Herranz, A. Llamazares, M.A. Sotelo, L.M. Bergasa, L. Magdalena*, Applied Soft Computing (ISSN: 1568-4946), Vol. 11(8), pages 4677-4691.

A.3.2 Conference Publications

2011 **Extended Floating Car Data system - experimental study**, *R. Quintero and A. Llamazares and D. F. Llorca and M. A. Sotelo and L. E. Bellot and O. Marcos and I. G. Daza and C. Fernández*, 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, (Germany).

2010 **Studying of WiFi Range-only Sensor and its Aplication to Localization and Mapping Systems**, *F. Herranz, M. Ocaña, L.M. Bergasa, M.A. Sotelo, D.F. Llorca, N. Hernández, A. Llamazares, C. Fernández*, IEEE International Conference on Robotics and Automation (IEEE ICRA 2010), pages 115-120, Anchorange (USA).

Bibliography

[Adarve et al., 2012] J. D. Adarve, M. Perrollaz, A. Makris and C. Laugier, "Computing Occupancy Grids from Multiple Sensors using Linear Opinion Pools". In *IEEE International Conference on Robotics and Automation*, St Paul, Minnesota, United States (2012).
https://hal.inria.fr/hal-00671211

[Alonso et al., 2012] J. M. Alonso, K. LeBlanc, M. Ocaña and E. Ruspini, "ABSYNTHE: Abstraction, Synthesis, and Integration of Information for Human-Robot Teams". In *International Workshop on Perception in Robotics, IEEE Intelligent Vehicles Symposium*, pages P14.1–P14.6 (2012).

[Andrieu et al., 2003] C. Andrieu, N. de Freitas, A. Doucet and M. I. Jordan, "An introduction to mcmc for machine learning". *Machine Learning*, volume 50(1), pages 5–43 (2003).
http://dx.doi.org/10.1023/A:1020281327116

[Antone and Friedman, 2007] M. Antone and Y. Friedman, "Fully automated laser range calibration". In *Proc. BMVC*, pages 66.1–66.10 (2007), doi:10.5244/C.21.66.

[Arras et al., 2008] K. O. Arras, S. Grzonka, M. Luber and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities." In *ICRA*, pages 1710–1715, IEEE (2008).

[Baig et al., 2014] Q. Baig, M. Perrollaz and C. Laugier, "A robust motion detection technique for dynamic environment moni-

toring: A framework for grid-based monitoring of the dynamic environment". *IEEE Robotics Automation Magazine*, volume 21(1), pages 40–48 (2014).

[Bekris et al., 2006] K. E. Bekris, M. Click and E. E. Kavraki, "Evaluation of algorithms for bearing-only slam". In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'06)*, pages 1937–1943 (2006).

[Biswas et al., 2002] R. Biswas, B. Limketkai, S. Sanner and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots". In *2002. IEEE/RSJ International Conference*, page 1014–1019 (2002).

[Blom and Bar-Shalom, 1988] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients". *IEEE Transactions on Automatic Control*, volume 33(8), pages 780–783 (1988).

[Borenstein, 1994] J. Borenstein, "Internal correction of dead-reckoning errors with the smart encoder trailer". In *In Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 127–134, IEEE, Munich, Germany (1994).

[Borenstein et al., 1996] J. Borenstein, L. Feng and C. J. Borenstein, "Measurement and correction of systematic odometry errors in mobile robots". *IEEE Transactions on Robotics and Automation*, volume 12, pages 869–880 (1996).

[Borenstein and Koren, 1987] J. Borenstein and Y. Koren, "Motion control analysis of a mobile robot". In *Transactions of ASME, Journal of Dynamics, Measurement and Control, Vol.109, No. 2*, pages 73–19 (1987).

[Borges and Aldon, 2004] G. Borges and M. Aldon, "Line extraction in 2d range images for mobile robotics". *J. Intell. Robotics Syst.*, volume 40(3), pages 267–297 (2004).

[Bosch, 2000] Bosch (Editor), *Automotive Handbook – 5th Edition*. Robert Bosch GmbH., Stuttgart (2000).

[Bouguet, 2000] J. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker". *Intel Corporation, Microprocessor Research Labs* (2000).

[Bradski and Kaehler, 2013] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., 2nd edition (2013).

[Braun et al., 2012a] D. Braun, M. Howard and S. Vijayakumar, "Optimal variable stiffness control: formulation and application to explosive movement tasks". *Autonomous Robots*, volume 33, pages 237–253 (2012a), 10.1007/s10514-012-9302-3. http://dx.doi.org/10.1007/s10514-012-9302-3

[Braun et al., 2012b] D. J. Braun, M. Howard and S. Vijayakumar, "Optimal variable stiffness control: Formulation and application to explosive movement tasks". *Autonomous Robots (in press)* (2012b).

[Brechtel et al., 2010] S. Brechtel, T. Gindele and R. Dillmann, "Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments". In *2010 IEEE International Conference on Robotics and Automation*, pages 3932–3938 (2010).

[Censi et al., 2005] A. Censi, L. Iocchi and G. Grisetti, "Scan matching in the hough domain". In *In Proc. of the IEEE Intern. Conference on Robotics and Automation (ICRA)* (2005).

[Chang et al., 2008] Y. Chang, H. Kuwabara and Y. Yamamoto, "Novel application of a laser range finder with vision system for wheeled mobile robot". In *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pages 280–285, IEEE (2008).

[Chavez-Garcia and Aycard, 2015] R. O. Chavez-Garcia and O. Aycard, "Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking". *IEEE Transactions on Intelligent Transportation Systems*, volume 17(2), pages 252–534 (2015).

[Chen et al., 2006] C. Chen, C. Tay, C. Laugier and K. Mekhnacha, "Dynamic environment modeling with gridmap: A multiple-object tracking application". In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6 (2006).

[Choi et al., 2007] Y. Choi, J. Hong and K. Park, "Obstacle avoidance using active window and flexible vector field with a laser range finder". *International Conference on Control, Automation and Systems (ICCAS)*, pages 2123 – 2128 (2007).

[Coué et al., 2006] C. Coué, C. Pradalier, C. Laugier, T. Fraichard and P. Bessiere, "Bayesian Occupancy Filtering for Multitarget Tracking: an Automotive Application". *Int. Journal of Robotics Research*, volume 25(1), pages 19–30 (2006), voir basilic : http://emotion.inrialpes.fr/bibemotion/2006/CPLFB06/. http://hal.inria.fr/inria-00182004/en/

[Cox, 1993] I. J. Cox, "A review of statistical data association techniques for motion correspondence". *International Journal of Computer Vision*, volume 10(1), pages 53–66 (1993). http://dx.doi.org/10.1007/BF01440847

[Danescu et al., 2011] R. Danescu, F. Oniga and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid". *IEEE Transactions on Intelligent Transportation Systems*, volume 12(4), pages 1331–1342 (2011).

[Dempster, 1967] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping". *Ann. Math. Statist.*, volume 38(2), pages 325–339 (1967). http://dx.doi.org/10.1214/aoms/1177698950

[Elfes, 1989] A. Elfes, "Using occupancy grids for mobile robot perception and navigation". *Computer*, volume 22(6), pages 46–57 (1989).
http://dx.doi.org/10.1109/2.30720

[Enge and Misra, 1999] P. Enge and P. Misra, "Special issue on global positioning system". *Proceedings of the IEEE*, volume 87(1), pages 3 –15 (1999).

[Fernández et al., 2004] J. L. Fernández, R. Sanz, J. A. Benayas and A. R. Diéguez, "Improving collision avoidance for mobile robots in partially known environments: the beam curvature method". *Robotics and Autonomous Systems*, volume 46(4), pages 205–219 (2004).

[Foley et al., 1990] J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes, *Computer Graphics: Principles and Practice (2Nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1990).

[Fortmann et al., 1983] T. Fortmann, Y. Bar-Shalom and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association". *IEEE Journal of Oceanic Engineering*, volume 8(3), pages 173–184 (1983).

[Fox et al., 1997] D. Fox, W. Burgard and S. Thrun, "The dynamic window approach to collision avoidance". *Robotics Automation Magazine, IEEE*, volume 4(1), pages 23 –33 (1997).

[Fox et al., 2000] D. Fox, S. Thrun, F. Dellaert and W. Burgard, "Particle filters for mobile robot localization". In A. Doucet, N. de Freitas and N. Gordon (Editors), *Sequential Monte Carlo Methods in Practice*, Springer Verlag, New York (2000), to appear.

[Friedman et al., 2000] J. Friedman, T. Hastie and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)". *Ann. Statist.*, volume 28(2), pages 337–407 (2000).
http://dx.doi.org/10.1214/aos/1016218223

[Fulgenzi, 2009] C. Fulgenzi, *Autonomous navigation in dynamic uncertain environment using probabilistic models of perception and collision risk prediction.* Theses, Institut National Polytechnique de Grenoble - INPG (2009).
https://tel.archives-ouvertes.fr/tel-00398055

[Fulgenzi et al., 2007] C. Fulgenzi, A. Spalanzani and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid". In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Rome (2007).
http://emotion.inrialpes.fr/bibemotion/2007/FSL07

[Gillespie, 1992] T. Gillespie, *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers (1992).

[Gindele et al., 2009] T. Gindele, S. Brechtel, J. Schroder and R. Dillmann, "Bayesian occupancy grid filter for dynamic environments using prior map knowledge". In *2009 IEEE Intelligent Vehicles Symposium*, pages 669–676 (2009).

[Gourley and Trivedi, 1994] Gourley and M. Trivedi, "Sensor based obstacle avoidance and mapping for fast mobile robots". In *IEEE International Conference on Robotics and Automation* (1994).

[Granström, 2012] K. Granström, *Extended target tracking using PHD filters*. Ph.D. thesis (2012).

[Hähnel et al., 2003a] D. Hähnel, D. Schulz and W. Burgard, "Mobile robot mapping in populated environments". *Advanced Robotics*, volume 17(7), pages 579–597 (2003a).
http://dx.doi.org/10.1163/156855303769156965

[Hähnel et al., 2003b] D. Hähnel, R. Triebel, W. Burgard and S. Thrun, "Map building with mobile robots in dynamic environments". In *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE Computer Society Press (2003b).

[Hentschel et al., 2008] M. Hentschel, O. Wulf and B. Wagner, "A gps and laser-based localization for urban and non-urban outdoor

environments". In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 149 –154 (2008).

[Husqvarna, 2008] Husqvarna, "Lawn-mowing robot" (2008). http://www.husqvarna.com

[Iocchi and Pellegrini, 2007] L. Iocchi and S. Pellegrini, "Building 3d maps with semantic elements integrating 2d laser, stereo vision and imu on a mobile robot". In *In Proc. of the 2nd ISPRS Int. Workshop 3D-ARCH*, Zurich, Switzerland (2007).

[iRobot, 2002] iRobot, "Roomba" (2002). http://www.irobot.com

[iRobot, 2007] iRobot, "Scooba" (2007). http://www.irobot.com

[Isard and Blake, 1998] M. Isard and A. Blake, *Icondensation: Unifying low-level and high-level tracking in a stochastic framework*, pages 893–908. Springer Berlin Heidelberg, Berlin, Heidelberg (1998). http://dx.doi.org/10.1007/BFb0055711

[Isard, 1998] M. A. Isard, *Visual Motion Analysis by Probabilistic Propagation of Conditional Density*. Ph.D. thesis, Department of Engineering Science, University of Oxford (1998).

[Jiménez-Palacios, 1999] J. L. Jiménez-Palacios, *Understanding and quantifying motor vehicle emissions with vehicle specific power and TILDAS remote sensing*. Ph.D. thesis, Massachusetts Institute of Technology (1999).

[Kaasalainen et al., 2011] S. Kaasalainen, A. Jaakkola, M. Kaasalainen, A. Krooks and A. Kukko, "Analysis of incidence angle and distance effects on terrestrial laser scanner intensity: Search for correction methods". *Remote Sensing*, volume 3(10), pages 2207–2221 (2011). http://www.mdpi.com/2072-4292/3/10/2207

[Kawata et al., 2008] H. Kawata, K. Miyachi, Y. Hara, A. Ohya and S. Yuta, "A method for estimation of lightness of objects with intensity data from SOKUIKI sensor". In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI 2008, Seoul, South Korea, August 20-22, 2008*, pages 661–664 (2008). http://dx.doi.org/10.1109/MFI.2008.4648020

[Kneip et al., 2009] L. Kneip, F. Tâche, G. Caprari and R. Siegwart, "Characterization of the compact hokuyo urg-04lx 2d laser range scanner". In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ICRA'09, pages 2522–2529, IEEE Press, Piscataway, NJ, USA (2009). http://dl.acm.org/citation.cfm?id=1703775.1703858

[Ko and Simmons, 1998] N. Y. Ko and R. Simmons, "The lane-curvature method for local obstacle avoidance". In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 3, pages 1615 –1621 vol.3 (1998).

[Kogut et al., 2007] G. Kogut, G. Ahuja, B. Sights, E. B. Pacis and H. R. Everett, "Sensor fusion for intelligent behavior on small unmanned ground vehicles". In *In Proc. SPIE*, page 6561, Orlando, FL (2007).

[Krantz, 1996] D. Krantz, "Non-uniform dead-reckoning position estimate updates". In *Proc. IEEE Int. Conf. Robotics and Automation* (1996).

[Kurdej et al., 2012] M. Kurdej, J. Moras, V. Cherfaoui and P. Bonnifait, *Map-Aided Fusion Using Evidential Grids for Mobile Perception in Urban Environment*, pages 343–350. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). http://dx.doi.org/10.1007/978-3-642-29461-7_40

[Latombe, 1991] J.-C. Latombe, *Approximate Cell Decomposition*, pages 248–294. Springer US, Boston, MA (1991). http://dx.doi.org/10.1007/978-1-4615-4022-9_6

[Lego, 1998] Lego, "Lego mindstorms" (1998).
http://www.mindstorms.lego.com

[Leonard et al., 2008] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy and J. Williams, "A perception-driven autonomous urban vehicle". *Journal of Field Robotics*, volume 25(10), pages 727–774 (2008).
http://dx.doi.org/10.1002/rob.20262

[Lingemann et al., 2005] K. Lingemann, A. Nüchter, J. Hertzberg and H. Surmann, "High-speed laser localization for mobile robots". *Robotics and Autonomous Systems*, volume 51(4), pages 275 – 296 (2005).

[Llamazares et al., 2013] A. Llamazares, V. Ivan, E. Molinos, M. Ocaña and S. Vijayakumar, "Dynamic obstacle avoidance using bayesian occupancy filter and approximate inference". *Sensors*, volume 13(3), pages 2929–2944 (2013).
http://www.mdpi.com/1424-8220/13/3/2929

[Llamazares et al., 2012] Á. Llamazares, E. J. Molinos, M. Ocaña, L. M. Bergasa, N. Hernández and F. Herranz, *3D Map Building Using a 2D Laser Scanner*, pages 412–419. Springer Berlin Heidelberg, Berlin, Heidelberg (2012).
http://dx.doi.org/10.1007/978-3-642-27579-1_53

[López, 2004] E. López, *Sistema de navegación global basado en procesos de decisión de Markov parcialmente observables. Aplicación a un robot de asistencia personal*. Ph.D. thesis, University of Alcalá (2004).
www.educacion.gob.es/teseo/mostrarRef.do?ref=317928

[Madgwick et al., 2010] S. Madgwick, R. Vaidyanathan and A. Harrison, "An efficient orientation filter for inertial measurement units (imus) and magnetic angular rate and gravity (marg)

sensor arrays". Technical report, Department of Mechanical Engineering (2010).
http://www.scribd.com/doc/29754518/
A-Efficient-Orientation-Filter-for-IMUs-and-MARG-Sensor-Arrays

[Mahony et al., 2005] R. Mahony, T. Hamel and J. M. Pflimlin, "Complementary filter design on the special orthogonal group so(3)". In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1477–1484 (2005).

[Mahony et al., 2008] R. Mahony, T. Hamel and J. M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group". *IEEE Transactions on Automatic Control*, volume 53(5), pages 1203–1218 (2008).

[Mekhnacha et al., 2008] K. Mekhnacha, Y. Mao, D. Raulo and C. Laugier, "Bayesian occupancy filter based "Fast Clustering-Tracking" algorithm". In *IROS 2008*, Nice, France (2008).
https://hal.inria.fr/inria-00336356

[Mertz et al., 2013] C. Mertz, L. E. Navarro-Serment, D. Duggins, J. Gowdy , R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppe, C. Urmson, N. Vandapel, M. Hebert and C. Thorpe, "Moving object detection with laser scanners". *Journal of Field Robotics*, volume 30(1), pages 17 – 43 (2013).

[Minguez and Montano, 2004] J. Minguez and L. Montano, "Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios". *Robotics and Automation, IEEE Transactions on*, volume 20(1), pages 45 – 59 (2004).

[Miyasaka et al., 2009] T. Miyasaka, Y. Ohama and Y. Ninomiya, "Ego-motion estimation and moving object tracking using multi-layer lidar". In *Proceedings of the IEEE Intelligent Vehicles Symposium*, page 151–156 (2009).

[Molinos, 2017] E. Molinos, *Dynamic Obstacles Avoidance Algorithms for Unmanned Ground Vehicles*. Ph.D. thesis, Universidad de Alcalá (2017).

[Montemerlo et al., 2008] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt and S. Thrun, "Junior: The stanford entry in the urban challenge". *Journal of Field Robotics* (2008).

[Montemerlo et al., 2009] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt and S. Thrun, *Junior: The Stanford Entry in the Urban Challenge*, pages 91–123. Springer Berlin Heidelberg, Berlin, Heidelberg (2009).
http://dx.doi.org/10.1007/978-3-642-03991-1_3

[Montesano et al., 2005] L. Montesano, J. Minguez and L. Montano, "Modeling the static and the dynamic parts of the environment to improve sensor-based navigation". In *IEEE International Conference on Robotics and Automation (ICRA)* (2005).

[Moore and Stouch, 2016] T. Moore and D. Stouch, *A Generalized Extended Kalman Filter Implementation for the Robot Operating System*, pages 335–348. Springer International Publishing, Cham (2016).
http://dx.doi.org/10.1007/978-3-319-08338-4_25

[Moras et al., 2014] J. Moras, V. Cherfaoui and P. Bonnifait, "Evidential grids information management in dynamic environments". In *17th International Conference on Information Fusion (FUSION)*, pages 1–7 (2014).

[Nakanishi et al., 2011] J. Nakanishi, K. Rawlik and S. Vijayakumar, "Stiffness and temporal optimization in periodic movements: An optimal control approach". In *Proc. of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 718–724, IEEE, San Francisco, CA, USA (2011).

[Nashashibi and Bargeton, 2008] F. Nashashibi and A. Bargeton, "Laser-based vehicles tracking and classification using occlusion reasoning and confidence estimation". In *2008 IEEE Intelligent Vehicles Symposium*, pages 847–852 (2008).

[Nègre et al., 2014] A. Nègre, L. Rummelhard and C. Laugier, "Hybrid sampling bayesian occupancy filter". In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1307–1312 (2014).

[Newman et al., 2006] P. Newman, D. Cole and K. Ho, "Outdoor slam using visual appearance and laser ranging". In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1180 –1187 (2006).

[Nguyen et al., 2005] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis and R. Siegwart, "A comparison of line extraction algorithms using 2d laser range finder for indoor mobile robotics". In *In Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1929– 1934, Edmonton, Canada (2005).

[Nuchter et al., 2003] A. Nuchter, H. Surmann and J. Hertzberg, "Automatic model refinement for 3d reconstruction with mobile robots". In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pages 394–401 (2003).

[Nuss et al., 2013] D. Nuss, B. Wilking, J. Wiest, H. Deusch, S. Reuter and K. Dietmayer, "Decision-free true positive estimation with grid maps for multi-object tracking". In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 28–34 (2013).

[Ocaña, 2005] M. Ocaña, *Sistema de localización global WiFi aplicado a la navegación de un robot semiautónomo*. Ph.D. thesis, Universidad de Alcalá (2005).

[Ocaña et al., 2004] M. Ocaña, L. Bergasa and M. Sotelo, "Robust navigation indoor using wifi localization". In *Proc. of the 10th*

IEEE Internacional Conference on Methods and Models in Automation and Robotics (MMAR2004), pages 851–856 (2004).

[Ocaña et al., 2005] M. Ocaña, L. Bergasa, M. Sotelo, J. Nuevo and R. Flores, "Indoor robot localization system using wifi signal measure and minimizing calibration effort". In *Proc. of the IEEE Internacional Symposium on Industrial Electronics (ISIE2005)*, pages 1545–1550 (2005).

[Pintor et al., 2016] R. Pintor, A. Llamazares, E. Molinos and M. Ocaña, "Mejora de la odometría de un robot móvil aplicando medidas inerciales". In *Proceedings of the 2016 Seminario Anual de Automática, Electrónica Industrial e Instrumentación SAEI 2016, Jul 6-8, 2016, Elche Spain*, page 57 (2016).

[Premebida and Nunes, 2005] C. Premebida and U. Nunes, "Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications". *Robotica, Actas do Encontrol Cientifico* (2005).

[Quigley et al., 2009] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler and A. Y. Ng, "ROS: an open-source robot operating system". In *ICRA Workshop on Open Source Software* (2009).

[Rawlik et al., 2010] K. Rawlik, M. Toussaint and S. Vijayakumar, "An approximate inference approach to temporal optimization in optimal control". In *Proc. of Neural Information Processing Systems (NIPS)*, Vancouver, B.C., Canada (2010).

[Rawlik et al., 2012] K. Rawlik, M. Toussaint and S. Vijayakumar, "On stochastic optimal control and reinforcement learning by approximate inference". In *Proceedings of Robotics: Science and Systems*, Sydney, Australia (2012).

[Reid, 1979] D. Reid, "An algorithm for tracking multiple targets". *IEEE Transactions on Automatic Control*, volume 24(6), pages 843–854 (1979).

[Ripley, 1987] B. D. Ripley, *Stochastic Simulation*. John Wiley & Sons, Inc., New York, NY, USA (1987).

[Saez and Escolano, 2004] J. M. Saez and F. Escolano, "A global 3d map-building approach using stereo vision". In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1197–1202 Vol.2 (2004).

[Samet, 1988] H. Samet, "An overview of quadtrees, octrees, and related hierarchical data structures". In *Theoretical Foundations of Computer Graphics and CAD*, pages 51–68, Springer (1988).

[Saval-Calvo et al., 2017] M. Saval-Calvo, L. Medina-Valdés, J. M. Castillo-Secilla, S. Cuenca-Asensi, A. Martínez-Álvarez and J. Villagrá, "A review of the bayesian occupancy filter". *Sensors*, volume 17(2) (2017).
http://www.mdpi.com/1424-8220/17/2/344

[Schleicher et al., 2010] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea and E. López, "Low-cost GPS sensor improvement using stereovision fusion". *Signal Processing*, volume 90(12), pages 3294 – 3300 (2010).

[Schulz et al., 2001] D. Schulz, W. Burgard, D. Fox and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association". In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation, ICRA 2001, May 21-26, 2001, Seoul, Korea*, pages 1665–1670 (2001).
http://dx.doi.org/10.1109/ROBOT.2001.932850

[Singh et al., 1991] S. Singh, D. Feng, P. Keller, G. Shaffer, W. Shi, D. H. Shin, J. West and B. X. Wu, "A system for fast navigation of autonomous vehicles". Technical Report CMU-RI-TR-91-20, Robotics Institute, Pittsburgh, PA (1991).

[Sohn and Kim, 2005] H. Sohn and B. Kim, "A robust localization algorithm for mobile robots with laser range finders". In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3545–3550 (2005).

[Sony, 1999] Sony, "Aibo" (1999).
http://www.sony.com

[Sovran and Bohn, 1981] G. Sovran and M. Bohn, "Formulae for the tractive-energy requirements of vehicles driving the epa schedules". *SAE Technical Paper*, (810184) (1981).

[Suzuki and Abe, 1985] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following". *Computer Vision, Graphics, and Image Processing*, volume 30(1), pages 32–46 (1985).
http://dx.doi.org/10.1016/0734-189X(85)90016-7

[Tartan Racing, 2007] Tartan Racing, "Boss" (2007).
http://www.tartanracing.org/

[Tay et al., 2008a] M. Tay, K. Mekhnacha, C. Chen, M. Yguel and C. Laugier, "An efficient formulation of the bayesian occupation filter for target tracking in dynamic environments". *International Journal of Vehicle Autonomous Systems*, volume 6(1-2), pages 155–171 (2008a).
https://hal.inria.fr/inria-00182089

[Tay et al., 2008b] M. K. Tay, K. Mekhnacha, M. Yguel, C. Coué, C. Pradalier, C. Laugier, T. Fraichard and P. Bessière, *The Bayesian Occupation Filter*, pages 77–98. Springer Berlin Heidelberg, Berlin, Heidelberg (2008b).
http://dx.doi.org/10.1007/978-3-540-79007-5_4

[Thrun, 1998] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation". *Artificial Intelligence*, volume 99(1), pages 21–71 (1998).

[Thrun, 2002] S. Thrun, "Robotic mapping: A survey". In *Exploring Artificial Intelligence in the New Millenium*, Morgan Kaufmann (2002).

[Thrun et al., 2000] S. Thrun, W. Burgard and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping". In *Proc. of the IEEE International*

Conf. on Robotics and Automation (ICRA), IEEE, San Francisco (2000).

[Thrun et al., 1998] S. Thrun, D. Fox and W. Burgard, "A probabilistic approach to concurrent mapping and localization for mobile robots". In *Machine Learning*, pages 253–271 (1998).

[Thrun et al., 2006] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian and P. Mahoney, "Winning the darpa grand challenge". *Journal of Field Robotics*, volume 23(9), pages 661–692 (2006).

[Tipaldi and Ramos, 2009] G. D. Tipaldi and F. Ramos, "Motion clustering and estimation with conditional random fields". In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 872–877 (2009).

[Topp and Christensen, 2005] E. A. Topp and H. I. Christensen, "Tracking for following and passing persons". In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, pages 70–76 (2005).

[Toussaint, 2009] M. Toussaint, "Robot trajectory optimization using approximate inference". In *Proc. of ICML* (2009).

[Ueda et al., 2006] T. Ueda, H. Kawata, T. Tomizawa, A. Ohya and S. Yuta, "Mobile sokuiki sensor system-accurate range data mapping system with sensor motion". In *International Conference on Autonomous Robots and Agents* (2006).

[Ulrich and Borenstein, 1998] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots". In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1572 –1577 vol.2 (1998).

[Urmson et al., 2008] C. Urmson, J. Anhalt, H. Bae, J. A. D. Bagnell, C. R. Baker , R. E. Bittner, T. Brown, M. N. Clark, M. Darms, D. Demitrish, J. M. Dolan, D. Duggins, D. Ferguson, T. Galatali, C. M. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, M. Likhachev, B. Litkouhi, A. Kelly, M. McNaughton, N. Miller, J. Nickolaou, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, V. Sadekar, B. Salesky, Y.-W. Seo, S. Singh, J. M. Snider, J. C. Struble, A. T. Stentz, M. Taylor , W. R. L. Whittaker, Z. Wolkowicki, W. Zhang and J. Ziglar, "Autonomous driving in urban environments: Boss and the urban challenge". *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, volume 25(8), pages 425–466 (2008).

[van de Ven et al., 2010] J. van de Ven, F. Ramos and G. D. Tipaldi, "An integrated probabilistic model for scan-matching, moving object detection and motion estimation." In *ICRA*, pages 887–894, IEEE (2010).

[Vu and Aycard, 2009] T.-D. Vu and O. Aycard, "Laser-based detection and tracking moving objects using data-driven markov chain monte carlo." In *ICRA*, pages 3800–3806, IEEE (2009).

[Vu et al., 2011] T.-D. Vu, J. Burlet and O. Aycard, "Grid-based localization and local mapping with moving object detection and tracking". *Inf. Fusion*, volume 12(1), pages 58–69 (2011). http://dx.doi.org/10.1016/j.inffus.2010.01.004

[Wang, 2004] C.-C. Wang, *Simultaneous localization, mapping and moving object tracking*. Ph.D. thesis, Pittsburgh, PA, USA (2004), chair-Thorpe,, Charles. http://portal.acm.org/citation.cfm?id=1023485#

[Wang et al., 2004] C.-C. Wang, D. Duggins, J. Gowdy, J. Kozar, R. MacLachlan, C. Mertz, A. Suppe and C. Thorpe, "Navlab slammot datasets" (2004), carnegie Mellon University. www.cs.cmu.edu/~{}bobwang/datasets.html

[Wang et al., 2003] C.-C. Wang, C. Thorpe and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas". In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan (2003).

[Wang et al., 2015] D. Z. Wang, I. Posner and P. Newman, "Model-free detection and tracking of dynamic objects with 2d lidar". *Int. J. Rob. Res.*, volume 34(7), pages 1039–1063 (2015). http://dx.doi.org/10.1177/0278364914562237

[Woodman, 2007] O. J. Woodman, "An introduction to inertial navigation". Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory (2007). http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf

[Yang and Wang, 2011] S.-W. Yang and C.-C. Wang, "Simultaneous egomotion estimation, segmentation, and moving object detection." *J. Field Robotics*, volume 28(4), pages 565–588 (2011).

[Ye and Borenstein, 2002] C. Ye and J. Borenstein, "Characterization of a 2-d laser scanner for mobile robot obstacle negotiation". In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11-15, 2002, Washington, DC, USA*, pages 2512–2518 (2002). http://dx.doi.org/10.1109/ROBOT.2002.1013609

[Yguel et al., 2006] M. Yguel, C. Tay, K. Mekhnacha and C. Laugier, "Velocity Estimation on the Bayesian Occupancy Filter for Multi-Target Tracking". Research Report RR-5836, INRIA (2006). https://hal.inria.fr/inria-00070190

[Yuan et al., 2015] T. Yuan, D. S. Nuss, G. Krehl, M. Maile and A. Gern, "Fundamental properties of dynamic occupancy grid systems for vehicle environment perception". In *2015 IEEE 6th*

International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pages 153–156 (2015).

[Zeng and Weng, 2007] S. Zeng and J. Weng, "Online-learning and attention-based approach to obstacle avoidance using a range finder". *Journal of Intelligent and Robotic Systems*, volume 50(3), pages 219–239 (2007).

[Zhao and Thorpe, 1998] L. Zhao and C. Thorpe, "Qualitative and quantitative car tracking from a range image sequence". In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 496–501, IEEE (1998).

[Zhou and Chirikjian, 2003] Y. Zhou and G. S. Chirikjian, "Probabilistic models of dead-reckoning error in nonholonomic mobile robots". In *Proc. of the IEEE International Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1594–1599, Taipei, Taiwan (2003).