Combining Particle Filter and Fiducial Markers in a SLAM-Based Approach to Indoor Localization of Mobile Robots

Alexandre de Oliveira Júnior - 42161

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering. Work developed during the double degree exchange program between the Polytechnic Institute of Bragança (IPB) and the Federal Technological University of Paraná (UTFPR).

Work oriented by:

Prof. PhD. Paulo Jorge Pinto Leitão

Prof. PhD. Eduardo Giometti Bertogna

Bragança

2021

Combining Particle Filter and Fiducial Markers in a SLAM-Based Approach to Indoor Localization of Mobile Robots

Alexandre de Oliveira Júnior - 42161

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering. Work developed during the double degree exchange program between the Polytechnic Institute of Bragança (IPB) and the Federal Technological University of Paraná (UTFPR).

Work oriented by:

Prof. PhD. Paulo Jorge Pinto Leitão

Prof. PhD. Eduardo Giometti Bertogna

Bragança

2021

Remember to look up at the stars and not down at your feet. Try to make sense of what you see and wonder about what makes the universe exist. Be curious. And however difficult life may seem, there is always something you can do and succeed at. It matters that you don't just give up.

Stephen Hawking

Dedication

I dedicate this work to my mother for all the support she has given me throughout my life, being my superhero. Also dedicating to all my friends and teachers who helped me to make this project possible.

Acknowledgement

Initially, I would like to thank my family, my father Alexandre, my stepfather João, my grandmother Maria, and especially my mother Vera. Mom, I would never have reached this goal without all your support and encouragement, thank you for everything!

I would also like to thank my supervisor, Professor PhD. Paulo Leitão for the support and trust given to me for the development of this work. Thank you for everything I could learn from you, and for all the opportunities that helped me to become a better professional and researcher.

I also thank my co-supervisor, Professor PhD. Eduardo Giometti Bertogna, for all his help, advice, and for making possible my first contact with scientific research.

I thank the PhD student Luis Fernando Piardi for all the help during the accomplishment of this project, which was only possible due to his teachings, mentoring, and patience. Thank you very much for guiding me on this journey.

I also would like to thank Professor PhD. André Chaves Mendes, for all the advice, suggestions, and brainstorms that helped me so much in solving problems in the implementation of this project.

I am also very grateful to all my research colleagues and friends, Jonas Queiroz, Heitor Assumpção, João Braun, Lucas Sakurada, Matheus Zorawski, Leonardo Sestrem, Gustavo Funchal, Victória Melo, Thadeu Brito, Arezki Chellal, Rubén Montanõ, Fernando García, José Lima, Pedro Pecora, Raul Kaizer and Tiago Franco, thank you very much for all the support and for welcoming me as part of CeDRI.

I would like to thank all my dear friends at UTFPR, this journey were the best years of my life thanks to each one of you, thank you for all the moments of joy, sorrow, adventure

and learning that we spent together. There are so many of you and I wish I could thank each one individually, but in special I would like to thank Edson Junior, Adrielli Goulart, Fábio Amaral, Roberta Leone, Karina Caetano, Lucas Vaz, Matheus Alves, Rafael Alves, Larissa Malizan, Carlos Boduar, Leonardo Furini and Gabriele Takano for becoming my second family, thank you for all the memories that I will carry with me throughout my life.

I also would like to thank the UTFPR in Campo Mourão, for all the experiences provided in the last 5 years, for the opportunity to be part of the dual degree program, and to give my sincere thanks to all my professors in the Electronic Engineering department who taught me so much.

Finally, I thank the IPB for having received me and for making the development of this work possible, as well as I thank all my professors in the Master in Industrial Engineering.

Abstract

Mobile Robotics is an ever-expanding field in several application areas, but it is still subject to numerous challenges, mainly related to localization in indoor environments. Approaches commonly used to solve localization problems in these environments, such as Simultaneous Localization and Mapping (SLAM), while quite effective in static environments, are still easily subject to inaccuracies and failures in dynamic environments, or environments with a very limited number of distinguishing features. In this work, an approach that aims to integrate a SLAM technique based on particle filtering with a position tracking algorithm for fiducial markers distributed in the indoor environment will be presented. The proposed approach allows to perform corrections for intrinsic errors accumulated in the particle filter, as well as a way to solve global localization problems by obtaining the position of global landmarks based on the detection of the markers. Experiments to validate the proposed localization system were performed in a simulation environment, which allowed to ensure the effectiveness of the method compared to the traditional approach with the particle filter. Based on the results obtained, the system was adapted for testing in a real environment, making use of a mobile robot integrated with the ROS framework.

Keywords: Mobile Robots; Indoor Localization; Robot Operation System, Simultaneous Localization and Mapping, Particle Filtering, Adaptive Monte Carlo Localization, Fiducial Markers.

Resumo

A robótica móvel é um campo em constante expansão em diversas áreas de aplicação, mas que ainda está sujeita a inúmeros desafios, principalmente relacionados a localização em ambientes interiores. Abordagens comumente utilizadas para a solução de problemas de localização nesses ambientes, como por exemplo, a Localização e Mapeamento Simultâneos (SLAM), apesar de bastante efetivas em ambientes estáticos, ainda são facilmente sujeitas a imprecisões e falhas em ambientes dinâmicos, ou com poucos características distintivas em sua extensão. Neste trabalho será apresentado uma abordagem que visa a integração de uma técnica SLAM baseada em filtragem de partículas com um algoritmo de rastreamento de posição de marcadores fiduciais espalhados pelo ambiente interior. A abordagem proposta permite a realização de correções de erros intrínsecos acumulados no filtro de partículas, além de uma maneira de resolver problemas de localização global obtendo-se a posição de pontos de referências globais com base na detecção dos marcadores fiduciais. Experimentos para a validação do sistema de localização proposto foram realizados em ambiente de simulação, que permitiram garantir a eficácia do método em relação a abordagem tradicional com o filtro de partículas. Com base nos resultados obtidos o sistema foi adaptado para a realização de testes em um ambiente real, fazendo uso de um robô móvel com integração ao framework ROS.

Palavras-chave: Robótica Móvel; Localização Interior; Sistema Operacional de Robôs, Localização e Mapeamento Simultâneos, Filtro de Partículas, Adaptive Monte Carlo Localization, Marcadores Fiduciais.

Contents

List of Tables

List of Figures

Acronyms

AGV Automated Guided Vehicle.

AMCL Adaptive Monte Carlo Localization.

AMR Autonomous Mobile Robot.

API Application Programming Interface.

CeDRI Research Center in Digitalization and Intelligent Robotics.

Cobot Collaborative Robot.

EKF Extended Kalman Filter.

GPS Global Positioning System.

HRI Human-Robot Interaction.

ID Identifier.

IoT Internet of Things.

IP Internet Protocol.

IPB Polytechnic Institute of Bragança.

KF Kalman Filter.

LiDAR Light Detection and Ranging.

MCL Monte Carlo Localization.

PF Particle Filter.

RBPF Rao-Blackwellized Particle Filter.

RMFS Robotic Mobile Fulfillment System.

ROS Robot Operating System.

SLAM Simultaneous Localization and Mapping.

UAV Unmanned Aerial Vehicles.

UTFPR Federal Technological University of Paraná.

UV-C Ultraviolet C.

V-REP Virtual Robot Experimentation Platform.

Chapter 1

Introduction

Mobile Robotics is one of the fastest growing fields of scientific research today, as its applications contribute to countless areas such as industrial automation, agriculture, health care, planetary exploration, entertainment, rescue operations, transportation, personal services, product delivery, smart warehouses, construction, sports, autonomous vehicles, drones applications, and many others [1]. Such a diversity of applications ensures a strong economic impact, even with the Covid-19 pandemic directly affecting the global industry, the Mobile Robotics market is expected to continue growing at record pace, moving US$23 billion in 2021 and reaching US$54 billion by 2023 [2].

A significant highlight in Mobile Robotics is found in the Autonomous Mobile Robots (AMRs), which possess the ability to navigate independently, without the need for human intervention, making intelligent decisions for task performance based on the information collected from the environment by sensors (e.g., laser sensors, sonar and cameras) [3]. However, in order to be able to perform its functions correctly, the robot needs to have knowledge of its position and orientation relative to a coordinate system, something that is extremely challenging especially in indoor environments, where it is impossible to take advantage of some localization technologies such as Global Positioning System (GPS), making it necessary to adopt other strategies like for instance the Simultaneous Localization and Mapping (SLAM) [4].

The SLAM techniques allow a mobile robot to acquire a topological representation

of the surrounding environment, generating a map through the information collected by sensors, mainly based on the encoders of the robot's wheels (odometry) and laser-based sensors such as Light Detection and Ranging (LiDAR) [5]. To ensure robustness in laser-based positioning systems, approaches involving the use of Particle Filters (PFs) are commonly used to more accurately estimate the position of a robot, trying to overcome local and global localization problems. However, the estimated pose may diverge from the actual pose due to errors in the sensor readings, besides that, laser-based systems are not as accurate in environments and areas that are featureless such as hallways, or that present multiple dynamic obstacles [6].

In this context, this work aimed to develop an indoor localization system for a wheeled mobile robot that would allows its autonomous navigation through the hallways of the Research Center in Digitalization and Intelligent Robotics (CeDRI), located at the Polytechnic Institute of Bragança (IPB). For this purpose, the scope of this project will be to develop an approach that combines laser-based and particles filters SLAM techniques with an artificial vision position tracking system of fiducial markers distributed in the environment, in order to achieve the localization of the robot with better reliability than using SLAM with LiDAR and odometry data only.

1.1 Objectives

The main purpose of this thesis is to develop a localization system for a mobile robot that can integrate a SLAM approach with particle filter and a system that performs fiducial marker detection which will be responsible for updating the robot's position information in the environment, removing intrinsic errors accumulated in the particle filter. In such a way, that the localization system can be applied to effectively perform the navigation of the robot in a dynamic indoor environment.

The methodology for accomplishing this objective comprises the execution of the following steps:

- Study and analysis of existing systems and algorithms for indoor localization for mobile robots, mainly based on SLAM and fiducial markers, as well as their integration with the Robot Operating System (ROS);

- Implementation of SLAM-based interior localization system integrating a particle filter and fiducial markers;

- Building a representation of the CeDRI hallways environment in 3D simulation for testing the proposed localization system;

- Integration of the developed system with a real mobile robot for validation in the real environment.

1.2 Document Structure

This document is divided into 7 chapters, for the description of the work carried out throughout the dissertation development.

The Introduction is transcribed in Chapter 1, in which some of the main problems involving the localization of mobile robots in indoor environments are briefly exemplified, being proposed a localization system based on SLAM techniques and a particle filter integrating them with a localization approach through fiducial marker detection.

Chapter 2 presents a State of the Art about AMRs, introducing its main applications in different fields, the main difficulties in obtaining a fully autonomous system focusing mainly on problems involving indoor localization. Some of the main techniques and approaches used for mobile robot localization in indoor environments will be presented.

Chapter 3 describes the architecture of the proposed localization system, that combines a SLAM approach over a particle filter, namely Adaptive Monte Carlo Localization (AMCL), and the performance of position correction updates based on fiducial markers tracking. For this purpose, the role of each component, their functionalities and operation process will be described.

Chapter 4 presents the procedures performed for the implementation of the proposed system in a simulation environment, namely the construction of the simulation scenario and the use of the ROS framework.

In Chapter 5, the activities performed to implement the system in a real environment and real mobile robot will be detailed, as well as the necessary changes performed when compared to the simulation experimentation.

Chapter 6 discusses the results obtained through the tests performed in both simulated and real scenarios, comparing the results in both environments, and discussing their implications.

Finally, Chapter 7 presents the conclusions and potential works and new approaches to be taken in the future.

Chapter 2

State of the Art

This chapter will make a brief contextualization about AMRs, presenting the main challenges found in this field, especially regarding the indoor localization, and introducing some of the main approaches used to solve this problem.

2.1 Autonomous Mobile Robots

Robotics had its origin in science fiction literature, and was popularized between 1917 and 1921, when the brothers Joseph and Karel Capek wrote stories that described concepts of intelligent automats and first introduced the term robot, derived from the Czech word, *robota*, meaning servant or worker. Over time, robots start to become an integrated part of reality, with the introduction of the first Automated Guided Vehicle (AGV) by Barret Electronics in 1953, and with the development of the Unimate, the first manipulator robot used on a production line created by General Motors Company in 1958 [7], [8].

Since then, industrial robots have been developed and deployed to replace or assist humans in the execution of repetitive, tedious and unsafe manufacturing processes. Furthermore, the robotics field has evolved to allow humans and robots to work collaboratively on tasks that are too complex to be fully performed only by robots or too expensive to be fully automated. Thus the Collaborative Robots (Cobots) have made Human-Robot Interactions (HRIs) possible by allowing the secure sharing of workspace with workers,

in contrast to the traditional industrial robotic systems that required the robot to be isolated from human collaborators in order to avoid accidents [9], [10].

Cobots and other industrial manipulators can often be attached to mobile platforms, such as in AGVs or in AMRs, making it possible to perform intralogistics and material handling tasks across the manufacturing floor. AGVs have some disadvantages compared to AMRs since they are entirely dependent on supporting infrastructure for their navigation, such as wires or magnets, which often keeps them stuck on fixed paths [8]. In contrast, AMRs can move to any location in a given space that is accessible and unobstructed, adapting and making their own decisions according to the dynamism of the environment.

The concept of self decision-making allows AMRs to have decentralized control systems, offering the possibility that different AMRs can independently communicate and negotiates about the division of resources and tasks among them or with others devices, dynamically adapt and react to demands or changes. Unlike the AGVs system, which is centralized and requires an external control unit for decision-making and task-sharing. Above all, the technological advances and the high interest in scientific research with AMRs have allowed them to be not only restricted to industrial applications but present in countless fields [11].

In agriculture, autonomous robots are being used for activities such as planting and harvesting, irrigating, providing nutrients for the soil, detecting pests and diseases, and more [12]. On planetary exploration, space rovers such as Curiosity and Perseverance, presented in Figure 2.1, explore the planet Mars studying its geology, composition, atmosphere, and searching for potential bio-signatures [13]. Several e-commerce companies, such as Amazon and Alibaba, use Robotic Mobile Fulfillment Systems (RMFSs) in their smart warehouses, where multiple mobile robots work in a coordinated way to transport products to pick stations, in this way saving time, optimizing pick order processes and prevent human-based errors [14].

The applications of AMRs in the e-commerce sector are not only restricted to warehouses, as pioneering companies such as DHL, FedEx, SF Express, Google, UPS and

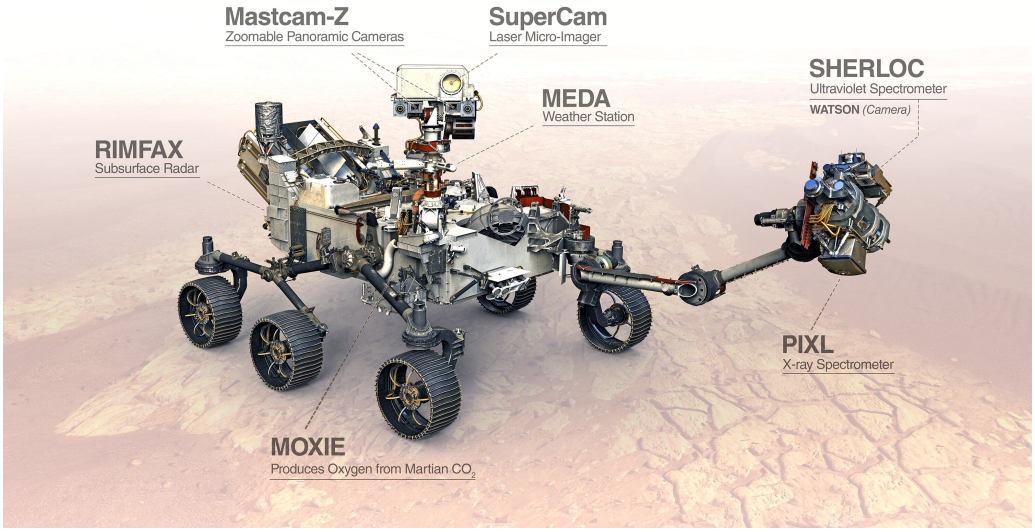

Figure 2.1: Overview of the Perseverance space rover [15].

Amazon are applying self-driving delivery robots to deliver products to customers in some urban areas, and not only that but are also developing and testing Unmanned Aerial Vehicless (UAVs) to carry out the product delivery services [16].

When it comes to healthcare, Mobile Robotics has become an important ally especially with the Covid-19 pandemic scenario, in which robots are being applied in the delivery and distribution of medical supplies, performing attendance though telepresence, carrying out tests for pre-diagnosis and disinfecting hospital environments through methods of emitting Ultraviolet C (UV-C) lights or spraying disinfectant liquids [17]. Social robots also known as service robots are widespread in real-world applications, mainly focused on HRIs they usually have interfaces, appearance and other aspects that make them more friendly and tangible for contact with people, being applied for example, as autonomous robotic pets, guides in hotels [18] and museums [19].

2.2 Challenges in Autonomous Mobile Robotics

The potential of mobile robotics proves to be vast and of significant impact on the contemporary society; however, in order to develop an autonomous mobile robotic system, it

is necessary to transcend numerous challenges such as locomotion, cognition, perception and navigation. These basic elements, that are integrated in the operation of an AMR, are exemplified in Figure 2.2.

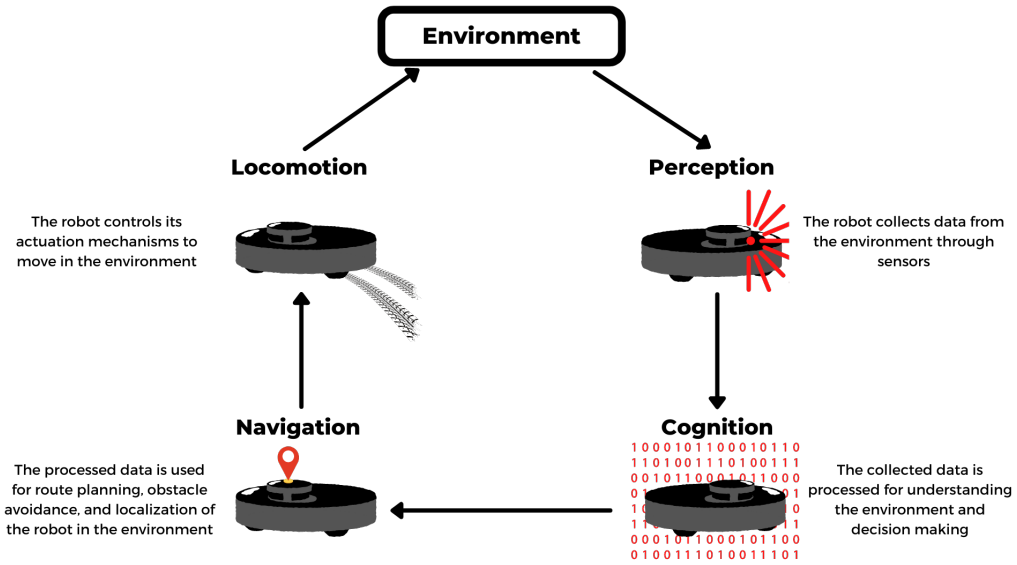

Figure 2.2: Basic processes required for the operation of an AMR.

Locomotion mechanisms are essential for a robot to move and perform its tasks, and their implementation demands an understanding of control concepts, dynamics, and kinematics. It is also necessary to know the environment in which the robot must move to ensure that the locomotion system is appropriate to the local conditions, allowing the control to be done with efficiency, stability, and maneuverability. For indoor environments for example, mobile robots are designed to move, run, or jump, and are usually equipped with legs or wheels, with the latter being considered one of the most efficient methods [4].

The perception deals with the robot's need to collect data about its surroundings and itself. For this purpose, different sensors can be used for data collection, from which relevant information is extracted for tasks such as localization, positioning, creation of maps or other representations of the environment, identification of obstacles, and in some cases even object recognition. The main sensors used in robotics applications are laser range finders, encoders, vision-based sensors, accelerometers, gyroscopes, depth sensors, sonars, infrared sensors, ultrasonic sensors, and others. It is the role of cognition to analyze

and interpret the input data collected by the sensors and to control the actuators, in order to allow the robot to operate correctly and perform its tasks [1].

The navigation can be considered as the main challenge for AMRs since for a robot to move autonomously between different points in an environment, be it known or unknown, it is necessary to integrate efficient solutions to the previously presented problems of locomotion, perception and cognition. Furthermore, the navigation involves other fundamental problems related to path planning (in most cases, a robot cannot take a direct route between the start and end points, so it is necessary to determine routes that guarantee arriving at the destination in an optimal way, minimizing time consumption and unnecessary detours), obstacle avoidance (it is important that the robot can safely reach its destination without being obstructed by an obstacle thus avoiding collisions), and most importantly the localization [3].

The localization of mobile robots is a huge challenge especially when it comes to indoor environments. GPS systems that are widely used for outdoor localization are still subject to errors that comprehend ranges about 10 meters and the high-frequency waves coming from satellites are easily obstructed by metal structures and walls, making GPS impractical for indoor localization [20]. Therefore, indoor localization needs to resort to other strategies that it can accurately determine the absolute position and orientation of the robot relative to a coordinate system [1].

In order to make the localization process possible, it is first necessary for the robot to understand the existing features in its surroundings by having some kind of spatial representation of the environment. The most widely used approaches is based on maps, which can be created from the data collected by sensors during the robot's navigation through the environment. It is important to ensure that the features represented by the map are equivalent to the features collected by the sensors, and that the accuracy of the map is equal to the accuracy required for robot navigation [4].

There are different typologies of maps such as topological, spatial, density, and metric, in which we can highlight the grid occupancy representation. In this typology the map is divided into two distinct types of cells arranged in grids, which are filled based on an

occupation threshold given the interpretation of the readings made by a sensor. This threshold corresponds to a probability rate of the cell being occupied or not, i.e., cells that represent regions where navigation is not possible have a high probability value and are considered occupied, while cells with low probability values of occupancy will be represented as a free space indicating zones where non-collision is guaranteed [21].

Once the map is generated, it can be used to assist in path planning for the robot and for the limiting errors in position estimation since distinguishable landmarks represented on the map enable error resets operations when comparing current sensor readings [22]. However, even with a representation of the environment, it is necessary to be concerned with some more generalized problems of localization in mobile robotics which can be classified into three different categories namely position tracking, global localization, and the kidnapped robot problem, which will be discussed below:

- **Position tracking**: in this case the robot's initial position is known, so the localization can be done by tracking the robot at each time instant while navigating the environment, which is accomplished through odometry and sensor data. While navigation occurs, the localization system uses the previously stored position to determine the new position of the robot. However, the localization by position tracking is subject to errors, due to the lack of precision of the odometry and of the sensors, besides it is necessary to be extremely secure about the initial position of the robot, otherwise errors will start to accumulate right from the start [23], [24];

- **Global localization**: is a more complex problem than the position tracking, since the robot does not know its initial position, and there is no previously stored information to start searching for its position in the environment map. This requires the application of methods known as global techniques to determine or estimate the robot's initial position [25];

- **Kidnapped robot problem**: this can be considered a branch of the global localization problems, but in this case the robot has an excellent localization and

without its knowledge is abducted and taken to a different location in the environment, continuing to believe it was in its previous position. Being able to identify that it was kidnapped, and having a way to estimate its new position, is essential for the operation of an AMR [23].

One factor that further exacerbates these problems is the existing dynamism in the indoor environments. In real scenarios an environment will hardly remain static, with objects and even obstacles moving, which confuses and even corrupts the data collected by the robot, making it even more difficult to obtain an accurate localization [25].

2.3 Indoor Localization Strategies

This section will address some of the main strategies used in the localization of mobile robots in indoor environments.

2.3.1 Localization with Fiducial Markers

Fiducial markers are elements widely used in mobile robotics localization applications. They can be characterized as artificial marks of well-defined sizes and shapes and have intrinsic Identifiers (IDs) for their identification, are used as reference points or measurements. Can be placed in environments as reference points, what makes possible to accurately determine their relative position and orientation between them and a camera positioned on the mobile robot. This information can be used to significantly increase the accuracy of an localization system [26].

There are several methods for fiducial marker generation and detection, but what makes them excellent for applications involving robotics, is the large number of open-source packages of fiducial markers that have integration with ROS, e.g., ARTag, ARToolkit, STag, AprilTag, ArUco, and others [27]. In Figure 2.3 as an example, some fiducial markers present in the ARTag package, also know as *ar-track-alvar*, are shown. However, since is necessary to apply artificial vision strategies for tracking the markers,

the use of the cited packages normally requires a high computational power, which can interfere with the efficiency of the localization system in the navigation of a robot that doesn't have much processing power, once that the data must be processed in real time.

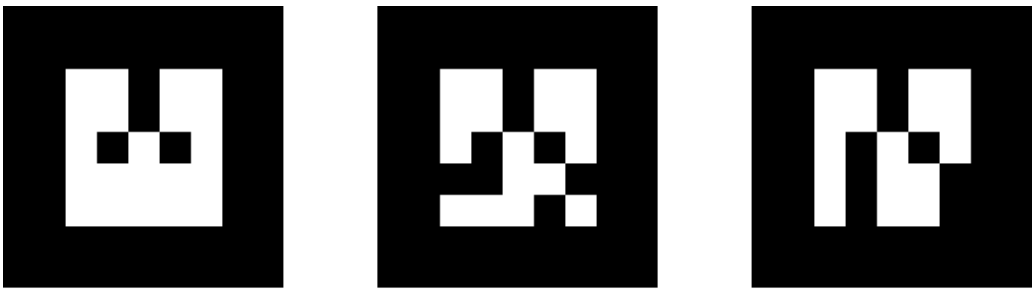

Figure 2.3: Ar-track-alvar markers, also know as AR tags, with the respective ids 0, 1, and 2.

Factors such as low cost, easy production and implementation, and small size, make fiducial markers a very versatile strategy to be applied in various scenarios of mobile robotics, mainly due to the great accuracy that can be obtained while tracking their position. For example, Zhang [28] has made use of QR codes as fiducial markers to perform localization and navigation of a mobile robot in an indoor environment, having positioned the QR codes on the ceiling of the environment and having an industrial camera positioned on the robot facing the markers. In this approach, the robot localization is entirely up to the markers, and a SLAM technique based on laser range is applied only to assist in route planning and obstacle avoidance.

In [29] fiducial markers are used as a complementary localization system for a UAV in power distribution line inspections, in simulations for target tracking for UAVs [30], and for the localization of mobile robots in a smart warehouse system, in with fiducial markers were positioned on top of the robots and a camera positioned on the warehouse ceiling is used to estimate the position of robots in the environment [31].

In a similar way, in [32] visual markers are captured by a ceiling-mounted camera to position virtual LiDAR sensors on small robots. Fiducial markers are also applied as a method to improve the localization system of a turtlebot robot in an indoor environment, combining the tracking position of the tags in walls with SLAM techniques and the

Kalman filter algorithm, achieving an autonomous navigation [33].

2.3.2 Simultaneous Localization and Mapping

SLAM can be defined as a method in which a mobile robot can build a map of an environment while simultaneously using it to locate itself. The mapping process is done online without requiring any prior knowledge of the robot's location, to estimate the position of the landmarks that make up the environment, as well as the trajectory the robot is taking. In the last decades the robotics community has been studying and developing new methods of performing SLAM, in order to make it more and more robust, for applications that comprise the domains of outdoor localization, underwater, for UAVs systems, and especially indoor localization [34].

Most of the SLAM techniques developed are based on probabilistic concepts as a way to reduce the effects of noise and uncertainty generated by the sensors. In mathematical terms, probabilistic SLAM approaches are based on estimating the position of the mobile robot in an unknown environment through a probabilistic distribution of positions over time $x_0 : _{\mathrm{T}} = x_0, x_1, x_2, ..., x_{\mathrm{T}}$ and the map m, taking into account the robot's odometry information $u_1 : _{\mathrm{T}} = u_1, u_2, u_3, ..., u_{\mathrm{T}}$ and the data collected from the environment through sensor observations $z_1 : _{\mathrm{T}} = z_1, z_2, z_3, ..., z_{\mathrm{T}}$. The Equation 2.1 describes the probabilistic SLAM method for position estimation [35].

$$p(x_{0:T}, m | z_{1:T}, u_{1:T}) \tag{2.1}$$

The performance of a SLAM approach is directly related to the desired application and the sensor to be used, which will determine the quality and quantity of information collected from the environment to create the map. Sonars are more applicable to underwater location, while ultrasonic sensors are used for dark environments, but have poor spatial resolution. Monocular, Stereo, and Omni-directional cameras are widely used in Visual-SLAM strategies both indoors and outdoors environments, but they are easily affected by factors such as changing lighting and there is a lack of depth information.

RGB-D sensors such as Kinects make it possible to obtain depth and image information by combining the strengths of vision-based and range-based sensors, normally are used indoors since they are very sensitive to light, and have very limited ranges [36], [37].

Range-based sensors are the most widely used for indoor localization, especially with 2D and 3D LiDARs. These sensors have the ability to measure distances to objects and structures according to the time required for the reflection of the light beams emitted by the sensor, thus they can have large reading ranges, are less affected by the environment [4], and although they are particularly more expensive their technology continues to advance given their applicability in the field of autonomous vehicles [38]. Several well-known SLAM algorithms use this type of sensing, such as the Gmapping, HectorSlam, LagoSLAM approaches for 2D SLAM systems and Loam, Google Cartographer, and Lego-Loam for 3D SLAM [39].

Figure 2.4 shows, as an example, the Slamtec's RPLiDAR sensor model A1M8, a relatively low cost LiDAR, widely used in SLAM applications for indoor localization and obstacle detection [40].

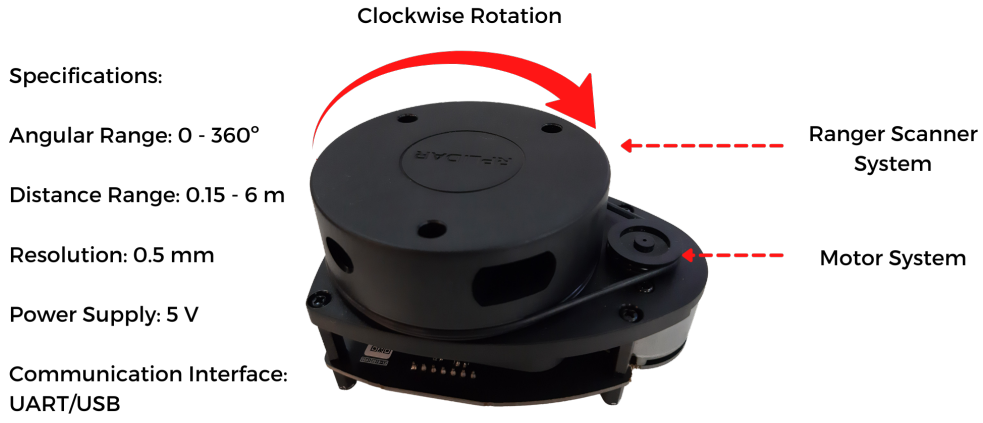

Figure 2.4: RPLiDAR model A1M8 (adapted from [41]).

Son [42] applied it to build a small mobile robot for mapping tasks by applying the Hector SLAM approach for map construction embedded in the ROS framework, while

Madhavan [43] used this sensor for the designer of an efficient system for obstacle detection, identification and avoidance.

As an example of applying a range-based SLAM approach, Figure 2.5 represents an indoor environment in simulation, by making use of a SLAM approach and a LiDAR sensor with similar range and resolution as the RPLiDAR on board of a mobile robot, an grid occupancy map represented in Figure 2.6 was generated, on it the white cells correspond to free spaces and the black cells represent occupied spaces, such as existing walls in the environment and obstacle contours.

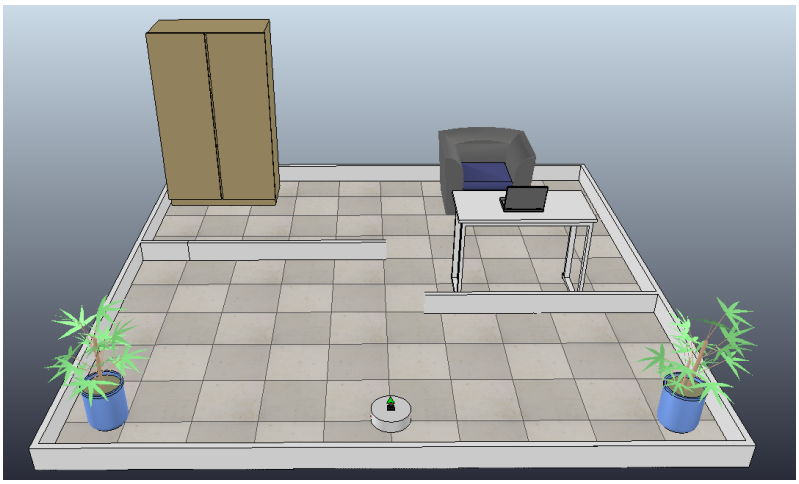

Figure 2.5: Simulated indoor environment.

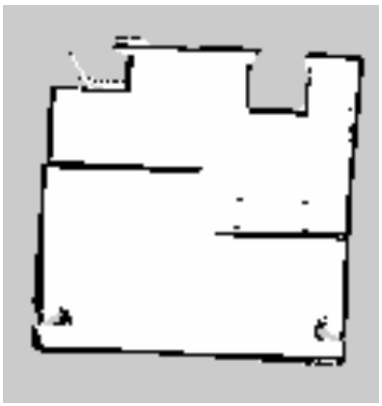

Figure 2.6: Grid occupancy map generated using a SLAM algorithm.

Although visually the generated map looks like a good representation of the environment, it is undeniable that there are discrepancies. SLAM-based approaches are subject to all kinds of discrepancies not only during mapping, but also during localization in a properly mapped environment. Many factors contribute to the lack of accuracy in addition to errors in the maps, such as inaccuracy of sensor readings and robot odometry, uncertainties in localization based on match scanning, conditions in the environment such as the presence of moving obstacles and geometric constraints, position uncertainties, and inaccuracy or inability to determine the initial position of the robot [44].

Although there are no absolute solutions for solving these problems, the following section will discuss some of the main methods and algorithms that are used to reduce the

errors derived from the sensor readings in order to provide a more accurate localization solution for the robot.

2.3.3 Kalman Filter

The Kalman Filter (KF) is an well known algorithm for estimating states or others unknown variables in linear dynamic systems, based on series of data collected over time, reducing noise and inaccuracies from the samples being collected. Characterized by real-time processing and robustness against interference, the KF is an efficient approach in trajectory tracking, robot control, orbit calculation, and in localization and navigation applications, having an important role in sensor data fusion [45].

The operation of the KF algorithm consists of two different stages, the first being responsible for the prediction of the state of the variables, while the second stage performs an update or correction of the state. This process occurs recursively, so the current state is estimated according to the last state [46]. Figure 2.7, shows as an example, the basic operation of KF in a localization system for a mobile robot.

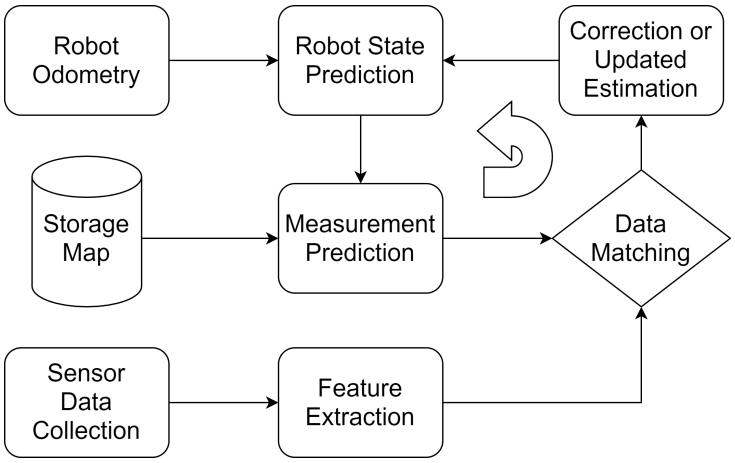

Figure 2.7: Kalman Filter Localization Flowchart (adapted from [4]).

During the first stage, the robot's state is estimated according to the system's dynamic variables, in this case the data from the robot's odometry. Through the information derived from the observable variables obtained by the readings of the sensors on board

the robot, which are compared with the stored map, updates are performed to correct the estimated robot state in the second stage. Since the process is recursive, when a new state is predicted at instant t, and a correction is applied to it in the next step, the corrected state is used as the initial condition for the estimation at instant $t + 1$.

The state of the robot x_k at time k defined by the KF in a linear system can be described by Equation 2.2, where A represents the matrix of state transitions, x_{k-1} the previous state defined by the filter, B the control signal input matrix applied to the control vector u_{k-1}, and w_{k-1} the process noise vector. Equation 2.3, represents the observations obtained from the environment through the robot's sensors, where the observation vector is represented by z_k, the observation matrix by H, and the noise vector of the sensor observations by v_k [45], [46].

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{2.2}$$

$$z_k = Hx_k + v_k \tag{2.3}$$

During the prediction stage the robot state is given by the mean, $\hat{x}_k^-$, being represented by the Equation 2.4. In addition, the estimated covariance deviation P of the state is determined, represented by Equation 2.5, where Q represents the covariance matrix of the process noise [47].

$$\hat{x}_k^- = A_k\hat{x}_{k-1} + Bu_k \tag{2.4}$$

$$\hat{P}_k^- = A_kP_{k-1}A^H + Q_k \tag{2.5}$$

In the correction stage, the KF algorithm calculates through Equation 2.6, the Kalman gain K, which indicates the confidence to be given to the information collected by the sensors. In the equation, R represents the covariance matrix of the noise in the collected data. Based on the Kalman gain, the robot state and the estimated covariance deviation are corrected through Equations 2.7 and 2.8 respectively, where in the last equation I represents an identity matrix [47].

$$K_k = P_k{}^{-} H_k{}^{H} (H_k P_k H_k{}^{H} + R_k)^{-1} \tag{2.6}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \tag{2.7}$$

$$\hat{P}_k = (I - K_k H_k)\hat{P}_k^- \tag{2.8}$$

The KF algorithm presents a high performance for the localization of a mobile robot, offer a good solution for the position tracking problem. However, it does not provide solutions for global localization or to the kidnapped robot problem, being its performance limited to linear systems [4].

Since a majority of practical applications, including in robotics, are related to non-linear systems, an improved form of KF was created, being known as Extended Kalman Filter (EKF), which is based on centering on the value of the first-order nonlinear Taylor expansion around the estimated system state, in order to transform the nonlinear system into a linear equation. The Equations 2.9, 2.10 and 2.11 represent how the EKF works in its prediction state, where f represents the state transition matrix [45].

$$A = \frac{df}{dx} | x = \hat{x}_{k-1} \tag{2.9}$$

$$\hat{x}_k^- = f(\hat{x}_{k-1}) \tag{2.10}$$

$$\hat{P}_k^- = A_k P_k A^T + Q \tag{2.11}$$

Equations 2.12, 2.13, 2.14 and 2.15 describe the EKF update step, where h represents the sensor observation matrix.

$$A = \frac{df}{dx} | x = \hat{x}_{k-1}^- \tag{2.12}$$

$$K_k = P_k{}^{-} H_k{}^{T} (H P_k{}^{-} H^T + R)^{-1} \tag{2.13}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-)) \tag{2.14}$$

$$\hat{P}_k = (I - K_k H)\hat{P}_k^- \tag{2.15}$$

2.3.4 Adaptive Monte Carlo Localization

The Monte Carlo Localization (MCL) approach makes use of a particle filter for position estimation of a robot. The particles act as a representation of a probability density function, that is, each particle represents a value that describes the possibility of the robot being in a given location. The higher the value assigned to the particle, the greater the chance that it is correct about the robot's location [48].

The value of the weight of each of the particles is defined by the ratio between the target and the proposed function, where each function is an estimate of the robot. The Equation 2.16, demonstrates the calculation of the noise and motion model that is applied to the particles according to the density function [4].

$$x_t{}^j \sim p(z_t|x_t, m) \tag{2.16}$$

The measurements collected by the sensor used for robot localization can be denoted by z_t and are used for comparison between the proposed position of the particle and the target, so that the weight of each particle can be defined as shown by the Equation 2.17, where $w_t{}^j$ represents the computed weight of each particle.

$$w_t{}^j = \frac{target}{proposal} \sim p(z_t|x_t, m) \tag{2.17}$$

A group of particles is created where the particles with the highest weight are kept. The MCL algorithm works more accurately if a large number of samples are used, however this brings higher computational consumption. MCL is also widely used for global and abducted robot localization problems, however it works better in static environments [4].

The approach called Adaptive Monte Carlo Localization (AMCL) is an improved version of MCL. It uses a KLD-Sampling strategy, which adapts the number of particles over time. So in situations where the position of the robot is uncertain, a larger number of particles are used to try to determine the position of the robot. Once there is more certainty about the robot's position, the algorithm reduces the number of particles to

reduce computational consumption [49].

In order to illustrate the operation of the AMCL particle filter, Figure 2.8 shows the beginning of the localization process of the algorithm applied to the indoor simulation environment introduced earlier in Figure 2.5. At this initial moment, the particles represented by the set of arrows are spread out on the environment map, where each one represents a possible position that the robot is occupying in the environment.

As the robot moves and collects new information with its sensors, the algorithm starts to converge its particles to locations that can best match the actual position occupied by the robot, as this process is performed the particles that represent positions with lower possibility are eliminated, decreasing the processing cost. Figure 2.9 represents the convergence process of the particles after the robot moves through the environment.

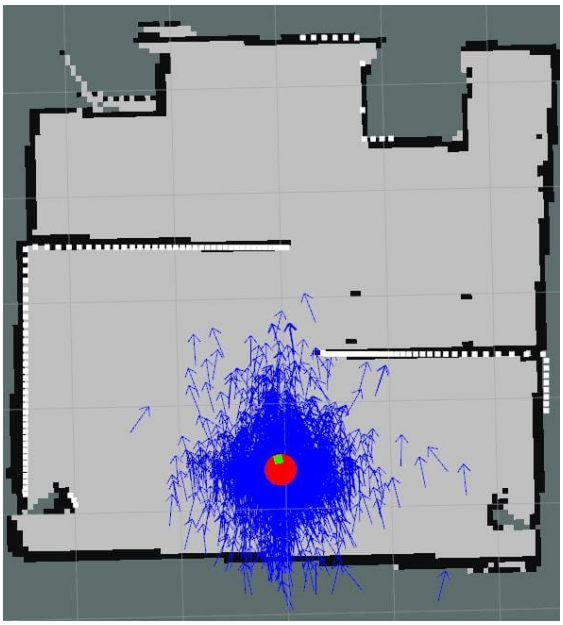

Figure 2.8: Initialization of the particle filter.

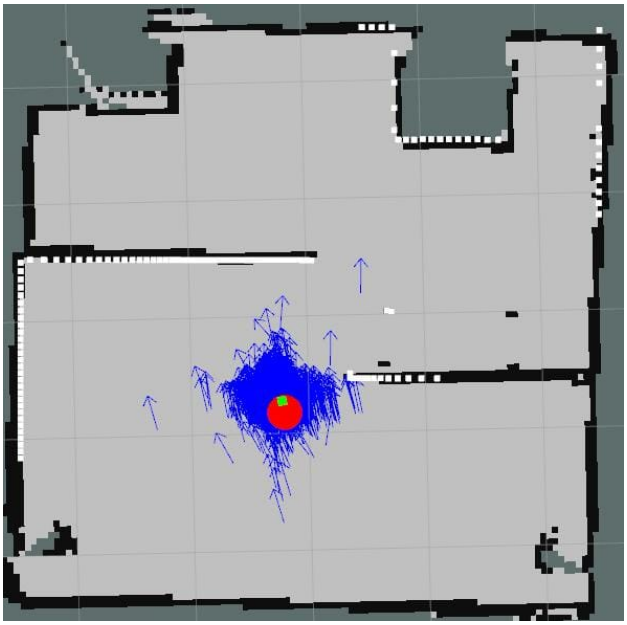

Figure 2.9: Convergence of the particle filter after the movement of the robot.

Chapter 3

Architecture of the System

This chapter will describe the proposed approach for an indoor localization system for mobile robots, which aims to take advantage of a SLAM technique combined with the particle filtering algorithm namely AMCL with the visual tracking of fiducial markers positioned on the ceiling of the environment. The fiducial markers are intended to provide global reference points, which will be integrated into the position estimation of the AMCL algorithm, to ensure greater robustness in the localization process of the mobile robot.

The position data obtained by detecting the markers will assist the robot to obtain a more accurate estimation for its initial position, helping to solve global localization problems. Additionally, it enables further updates throughout the robot's navigation through the environment, which eliminated errors accumulated by the AMCL algorithm's position estimation or due to errors or lack of accuracy of the robot's sensors and odometry, thus ensuring a solution to the problems of position tracking and the abducted robot, making it possible to obtain an indoor localization system that enables autonomous navigation of a mobile robot. A detailed description of the main concepts of operation and integration of the proposed approach will be presented below.

3.1 Mapping and Markers Database

The first step in making the system work is mapping the environment. As mentioned in Chapter 2, the robot needs a representation of the environment so that it can locate itself and perform the path planning and obstacle avoidance tasks. But something that causes a lot of confusion and doubt among beginners in Mobile Robotics is: "If a SLAM approach allows to localize the robot simultaneously to the mapping process, why is necessary to use a previously built map?"

In fact, SLAM allows the robot to locate itself in an unknown environment while building a map, however, imagine this location as being directly related to the robot's position in relation to the map, as described in Equation 2.1. When it is necessary to relate real points in an environment it is essential to have a correlation between the map and those points. With this correlation, it is easier to determine between which points the robot needs to move, and to understand which position on the map corresponds to the global position in the real environment. In addition, the robot will continue to apply the SLAM algorithm but now using the generated map to help it to determine its position through scan matching. The generated map also helps the robot to create local maps, in which information about changes in the environment, such as new obstacles, can be updated.

In the flow diagram shown in Figure 3.1, the basic operation of the mapping process in a LiDAR-based SLAM approach is presented. During map creation, the data collected by the LiDAR sensor are interpreted to extract features and landmarks present in the environment, which can be for example walls, furniture, and other obstacles. This process makes easier the localization of the robot by combining the odometry information with the correspondence of the already mapped landmarks with the data collected by the sensor. When new landmarks or unexplored areas are identified, they and their respective locations in the environment are updated to the map.

Having a properly generated map, another important task to be performed is the creation of a database which contains the corresponding coordinates of the fiducial markers

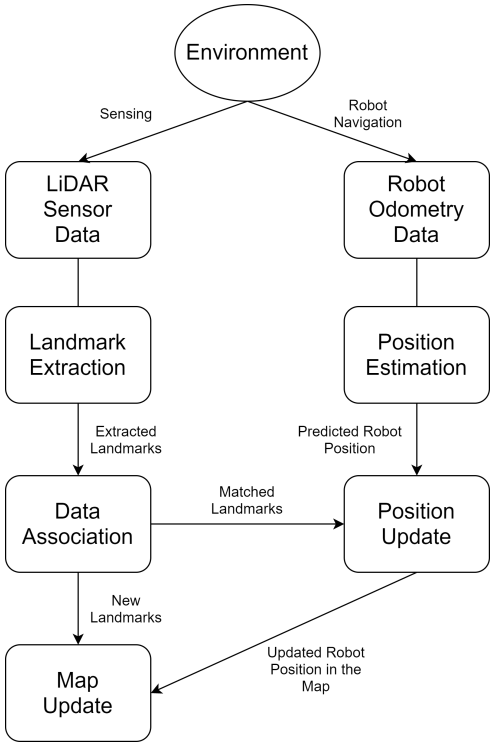

Figure 3.1: Mapping process flow diagram.

distributed throughout the environment. In the proposed approach it was decided to place the markers on the ceiling of the environment, to prevent obstacles from getting between the field of view of the camera positioned on the robot and the markers, if they were fixed on the walls of the environment.

It is crucial that the coordinates stored in the database correspond precisely to the position and orientation occupied by each one of the markers in relation to the coordinate system used to describe the environment, and that this system is equivalent to the coordinate system that describes the environment map. This means, if a point is considered as the origin of the environment's coordinate system, this point must also be the relative origin of the markers and of the map.

The database will be used by the localization system to assist in the global position and orientation estimation of the robot, this is why it is so important to avoid any errors in the measurements of the marker positions or even when fixing them in the environment, any inconsistency will inevitably introduce inaccuracies to the system.

3.2 Localization with SLAM and Fiducial Markers

As established earlier, the proposed indoor localization system is based on the integration of a SLAM approach with the tracking of fiducial markers distributed around the environment. The flow diagram given in Figure 3.2 presents an overview of the architecture of the system.

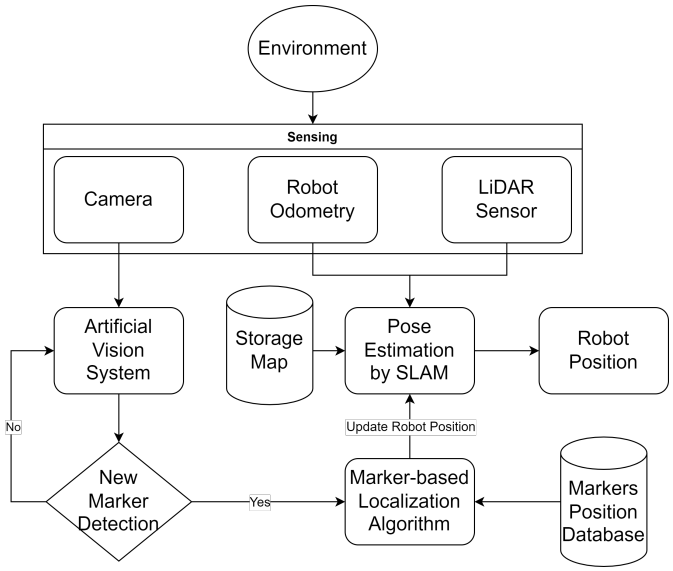

Figure 3.2: Architecture of the indoor localization system.

During the navigation of the robot through the environment, its sensors keep collecting data, and the SLAM method interprets the odometry information from the robot and the LiDAR sensor to estimate the robot's position using the previously created map as a reference. In order to get a more accurate estimation, it is possible to use some strategy that helps reduce the errors generated due to discrepancies in the sensor readings, such as the AMCL particle filter described in Chapter 2.

Despite of the improvement in position estimations with the application of the particle filter, discrepancies easily remain, especially in cases related to global localization or kidnapped robot problems in dynamic or low-features environments and due to the lack of precision in the robot's odometry. The robot's absence of knowledge of its initial position accompanied by changes in the sensor readings due to the existence of moving obstacles

makes it difficult for the AMCL to converge on a correct position, or even can make a correct position diverge to an incorrect position. The same can occur in environments without many landmarks, since the readings obtained by the sensors can be similar at various points in the environment.

In order to prevent these problems and also to correct minor localization errors accumulated during navigation, updates for correcting the position estimated by AMCL are performed based on the tracking of fiducial markers. A camera positioned on the robot and facing the ceiling collects images that are processed by machine vision techniques that allow the detection of the markers, as well as determining the distance between the marker and the camera. This information is used to determine the robot's global position and consequently update it in the AMCL, whenever this update is required.

There can be numerous strategies that can be used to define the timing of the position update that can impact the accuracy of the system as well as its processing cost. In the strategy chosen for this system, the update request is made whenever the robot reaches a predefined waypoint that is part of the calculated path to be taken by the robot to reach its destination, and there is a marker within the camera's field of view. This periodic process is used so that a constant process of position updates does not overload the system. A request is also always sent at the beginning of the robot's navigation process so that an initial position can be obtained for the system, if there is no marker in the field of view, the robot navigates using only the position estimate from the SLAM algorithm and the particle filter until a marker becomes available. The operation of the localization correction system for the mobile robot based on the fiducial markers is described by the Algorithm 1.

With the correction update request made, the artificial vision system identifies the ID i of the fiducial marker. Then, the next procedure gets the position and orientation $(x_m, y_m, z_m, \theta_m)$ of the marker i in relation to the camera (P_m). It is worth noting that if there is more than one marker within the camera's field of view, only the most closest marker to the robot will be considered. Subsequently, the algorithm identifies the P_g, i.e., the global position and orientation $(x_g, y_g, z_g, \theta_g)$ of the marker checking in the database.

Algorithm 1 Marker-based Localization Algorithm

```
1: while Robot_navigation do
2:     if Marker_detected then
3:         i = Get_marker_ID(Marker);
4:         P_m = Get_marker_pose_related_to_camera();
5:         P_g = Get_global_marker_pose_database(i);
6:         P_r = Calculate_robot_position(P_g, P_m);
7:         Correct_error_from_pose_estimator(P_r);
8:     else
9:         Estimate_robot_position();
10:    end if
11: end while
```

With the procedure that correlates the global position of the marker P_g, and the position in relation to the camera obtained by the artificial vision system P_m, it is possible to determine the global position and orientation of the camera in the environment. Once the camera is fixed, through the existing offset between the camera and the robot, the robot's position (P_r) is determined.

The algorithm then updates the robot's position P_r in the AMCL algorithm, thus removing any intrinsic error accumulated by the estimator, since it will use this position for the convergence of its particles and to estimate the next positions according to the robot's navigation.

Chapter 4

Simulation of the Localization System

This chapter presents the process of implementation of the proposed indoor localization system in a simulation environment integrated with the ROS framework. An overview of the created simulation environment, the used ROS packages, and the developed communication structure between the different packages and algorithms for the operation of the system will be presented. The configuration of a ROS network for splitting processing between different devices will also be covered.

4.1 Simulation Environment Overview

In this section the simulation environment developed for testing the localization system will be presented, reviewing the simulation software used, the main characteristics of the simulation scenario as well as the distribution of the fiducial markers.

4.1.1 V-REP Simulator

The Virtual Robot Experimentation Platform (V-REP), which has become CoppeliaSim Robot Simulator in versions later than 2019, is a scalable and versatile framework for

simulating multi-robot applications available for free in its student version. V-REP has its own integrated development system for distributed control architecture, which allows control of its models and objects through embedded algorithms in systems such as remote Application Programming Interface (API) clients, the BlueZero node or which the ROS framework [50].

V-REP offers many different calculation modules that can be used to operate the behavior of your objects such as kinematics, dynamics, collision detection, path and motion planning. In addition, the platform offers numerous objects to be used in scenarios such as visual sensors, force sensors, proximity sensors, lighting elements, and several robotic models [51]. The Figure 4.1 shows the V-REP in its version 3.3.2 PRO EDU, the same version used during the development of this work. In the scene, it is also presented some of the robotic models available in the simulator.

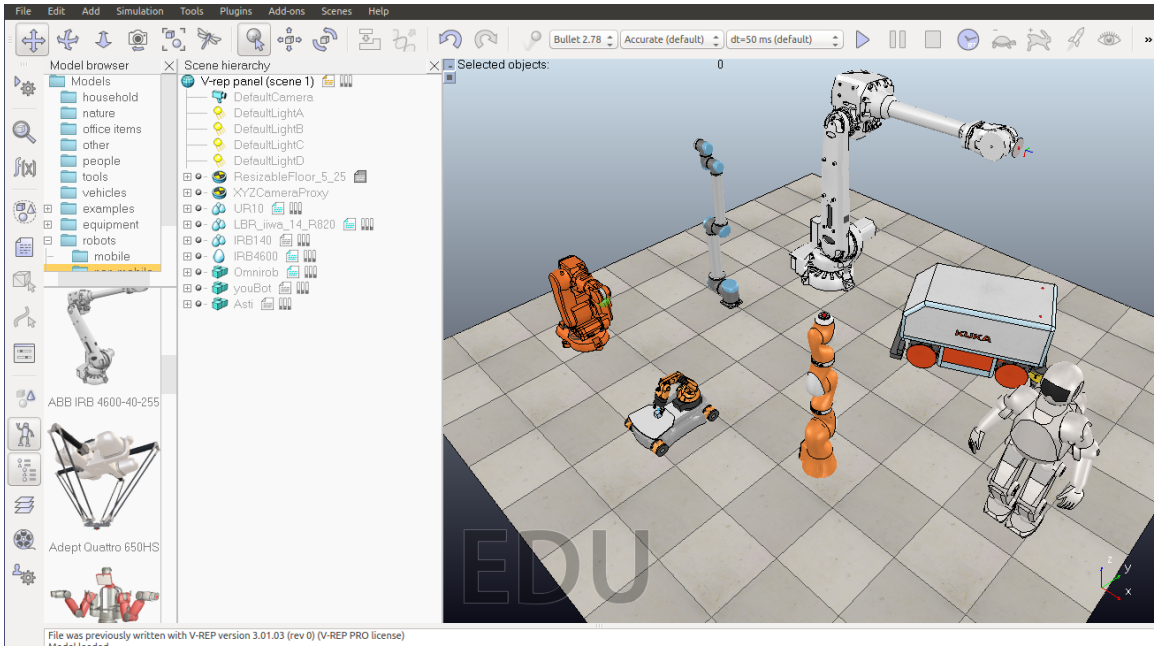

Figure 4.1: Example of a V-REP simulation scene showing the diversity of robotic models available on the platform.

The control performed through scripts of the models present in the simulator can be done using a wide range of programming languages such as Python, Java, C/C++, Matlab or Octave and Lua making it very versatile for the development of applications

by the user. Communication between the V-REP and ROS is made possible through an implemented ROS node, which makes it possible to call commands present in the V-REP through ROS services, and also to perform data transmission through publishers and subscribers [51].

V-REP is available for Windows, Linux, and Mac OS and can be downloaded from the simulator's official website [52], where is also possible to found tutorials and documentation of the framework.

4.1.2 The Real Indoor Environment

The simulation scenario built in the V-REP simulator for the tests with the localization system was based on the hallways of CeDRI located at the IPB, which are shown in Figures 4.2, 4.3, 4.4 and 4.5.

Figure 4.2: CeDRI entrance.

Figure 4.3: CeDRI main hallway.

The environment of the hallways, have dimensions of approximately 26x20 meters, consisting of a main hallway with access to the entrance and exit doors and the robotics laboratories, and a side hallway that gives access to the computer laboratories area. It is a complex environment for the navigation and localization of a mobile robot, because it has a large flux of people that end up interfering in the sensor readings, and there are regions with very large extensions of hallways where there are few features that make it difficult to distinguish them by the SLAM algorithm. However, there are also some distributed

Figure 4.4: Computer laboratories hallway.

Figure 4.5: Robotics laboratories and exit.

elements, which act as landmarks, such as the IPB Eco Buggy [53] a light electric vehicle, which is shown in the Figure 4.5, and also cabinets used by the IPB students.

4.1.3 Simulated Environment

In the V-REP simulator, a scaled representation of the CeDRI hallways was built, which is shown in Figure 4.6. In the scenario, the gray structures represent the walls of the environment, the red blocks the cabinets of the students, the green block the electric vehicle, and the blue object a mobile robot.

The wheeled mobile robot used for the localization tests in V-REP was equipped with a LiDAR sensor and an RGB camera. The LiDAR sensor is responsible for taking the necessary readings for the SLAM algorithm, along with the robot's odometry information, which can be obtained by a ROS topic of the simulator. The RGB camera was positioned facing the ceiling of the environment in order to be possible to detect the positioned fiducial markers. The Figure 4.7 shows the mobile robot as well as the attached sensors.

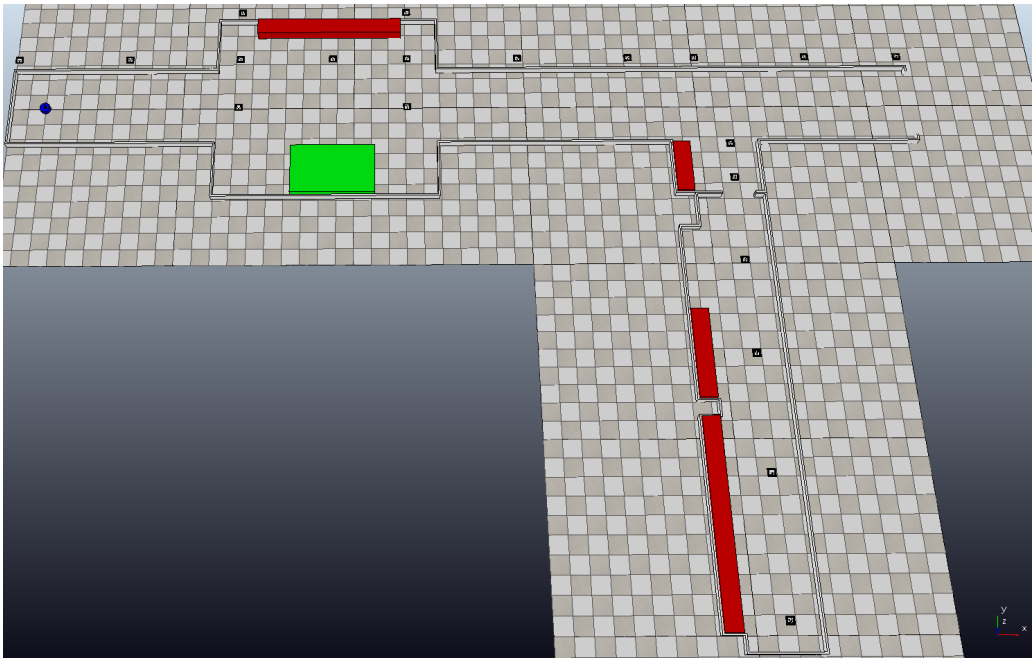

Figure 4.6: CeDRI hallways simulation scenario.

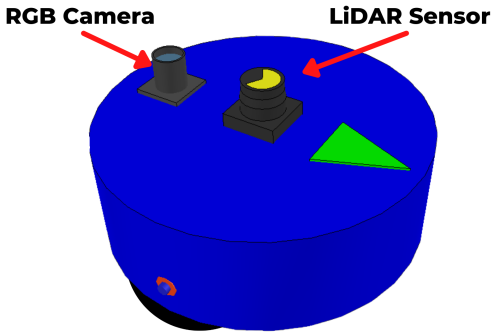

Figure 4.7: Wheeled mobile robot and its on-board sensors.

The fiducial markers chosen to be used in the localization system were the AR tags belonging to the ROS package *ar_track_alvar*, some examples of these markers were shown in Figure 2.3. A total of 20 AR tags of dimensions 20x20 cm were distributed across the ceiling of the simulation environment (which is 2.5 m in height). The distribution of the markers with their respective IDs is portrayed in the Figure 4.8 which represents a floor plan of the environment, was carried out in a way to allow the observation of the markers in different routes between different points of the environment. Aspects of the

real environment were considered when choosing the position of the markers, avoiding areas close to pillars and doors that could interfere in the field of view of the camera, or even near to lights, since the illumination can affect the detection of the markers.

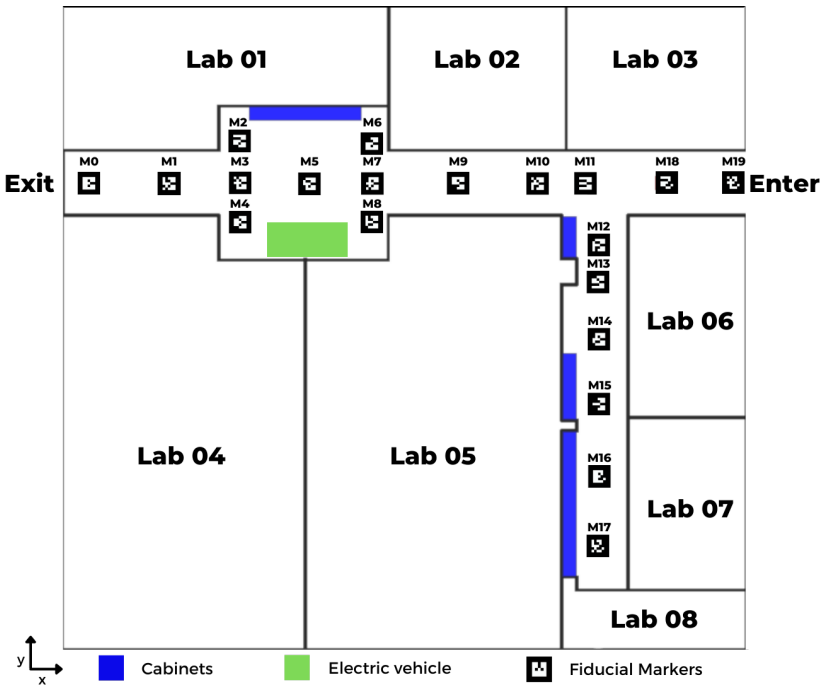

Figure 4.8: Flor plan of the indoor environment.

As the robot is intended to navigate between the labs, markers were placed at the entrances of the labs as a way to ensure that in most scenarios the robot will be able to obtain its initial position based on the markers, and they were also placed at points that act as waypoints on the routes. Another important aspect is that since the patch updates are not performed constantly, there will not always be a marker within the camera's field of view, so the markers have been distributed so that they act as the waypoints on the different routes that can be taken by the robot, and are considered in the system's path planning. The image processing to determine the distance of the marker is a procedure that consumes a lot of computational resources, so it is preferable to use as few markers as possible, and positioning the markers in a real environment is something that can be laborious, and even introduce errors to the system if they are not correctly positioned.

The position of the ID 0 marker was assumed to be the origin point of the environment

coordinate system, i.e. the point (0, 0) for the x and y axes, and the same point for mapping the environment as well as for defining waypoints. The Table 4.1 presents the coordinates of each of the markers distributed in the environment.

Marker ID	X (m)	Y (m)	Marker ID	X (m)	Y (m)
0	0	0	10	16.5	0
1	3	0	11	18.3	0
2	6	1.5	12	19.5	-1.5
3	6	0	13	19.5	-2.5
4	6	-1.5	14	19.5	-6
5	8.5	0	15	19.5	-8.5
6	10.5	1.5	16	19.5	-11.5
7	10.5	0	17	19.5	-14.9
8	10.5	-1.5	18	21.3	0
9	13.5	0	19	23.8	0

Table 4.1: Coordinates of the fiducial markers in the environment

4.2 Making Use of the ROS Framework

The ROS framework allows the integration of numerous packages and libraries for a quick prototyping of software in robotics applications, without being necessary to start from scratch and recreate algorithms that have already been well developed by the community. In addition, the standardization of communication between different robotic systems allows the developed algorithms to be tested in the simulation environment and on a real robot without substantial changes, which is also made possible by the excellent integration between V-REP and ROS. In this section, the ROS packages used in the development of the proposed indoor localization system will be briefly introduced.

4.2.1 SLAM Gmapping

SLAM gmapping is a widely used ROS package for the mapping process, being a laser scan-based SLAM approach that makes use of Rao-Blackwellized Particle Filter (RBPF) for the estimation of the robot pose [54]. The package has a node called *slam_gmapping*

which compiles the odometry and laser sensor information to create a grid occupancy map, which can be saved through the *map_saver* node of the *map_server* topic which generates a *.yaml* and a *.pgm* file describing the generated map [55].

The gmapping algorithm works in such a way that a message filter evaluates whether the system has collected a position reading from the robot's odometry along with a reading from the laser sensor, discarding any odometry data that does not come with a scan from the laser sensor. Gmapping also uses a motion filter to determine whether or not information from the LiDAR sensor will be processed; if the robot has not performed a sufficient amount of linear or angular motion or has not reached a timestamp since the last update, the data is discarded in order to save processing. The thresholds for amount of motion and time can be changed by the user, according to his needs as well as the computational power of the system. After the message and motion filtering process, RBPF is applied to determine the new position of the robot based on its particle convergence [56].

4.2.2 Navigation Stack

The ROS navigation stack consists of a cluster of numerous navigation-related algorithms that can be used to develop applications with AMRs, with the main goal of enabling a robot to travel along routes in an environment without colliding with any obstacles [57]. Among the available algorithms, the two most pertinent to the idealization of the proposed localization system will be discussed: AMCL and *move_base*.

The ROS implemented version of the Adaptive Monte Carlo Localization algorithm, the AMCL package is set up as a probabilistic localization system for a robot moving in 2D, localizing it relative to a known map. The algorithm works mainly using grid occupancy maps and laser sensors, but also allows applications using sonar or vision sensors. The package allows the customization of different parameters by the user for the adaptation of the algorithm to the type of odometry of the robot and the laser sensor, besides allowing the change of the number of particles used for the position estimation,

the higher the number of particles the higher the accuracy as well as the computational consumption [58].

The process of robot location estimation by AMCL initially consists of a set of possible poses that can represent the robot's position and orientation relative to the map. Each of these poses has a probability associated with it, the higher the probability the greater the chance that the robot is occupying that location. As the robot moves through the environment, the readings taken by the LiDAR sensor are compared to the expected readings in each of the poses that make up the set. If the readings are consistent the probability assigned to that position increases, and if it is not consistent the probability decreases. Over time poses with very low probabilities are eliminated, eventually causing the pose to converge to the robot's actual position. While the estimation is being done the candidate poses to be proven follow the robot's movement, based on the robot's odometry data [59].

The initial position of the robot can be published in an AMCL topic called *initial_position*, using it causes particle convergence to occur relative to the given position. If the robot's initial position is not known it is possible to use the *global_localization* service that will distribute the particles throughout the environment in search of the robot's global position [59]. Although the algorithm works well for global localization of the robot in a static environment, the same cannot be said for dynamic environments, since the divergence in the readings taken by the laser sensor can cause convergence to take a long time to occur, or even not to occur correctly.

The *move_base* package enables a robot to navigate through the environment by implementing an action system, which actuates the robot to reach a desired destination given its current position. The *move_base* node provides an interface for configuration, interaction and execution with the other packages in the navigation stack. In the Figure 4.9 a high-level view of the node's interaction with the other components of the navigation stack is shown [60].

The goal poses can be published to the *move_base* node via the *move_base_simple/goal*

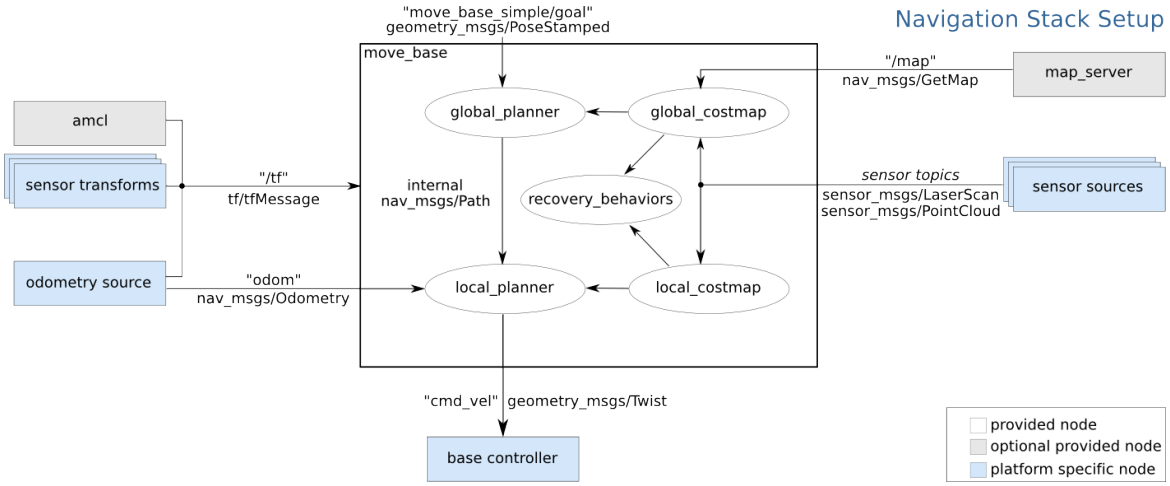

Figure 4.9: Integration of the *move_base* node with the navigation stack modules [60].

topic that receives position and orientation information about the goal. The grid occupancy map previously generated through gmapping or another algorithm can be uploaded to *move_base* via the *map_server* node, which will be used by the system along with the data coming from LiDAR to create global and local cost maps [60].

The localization of the robot is done through the AMCL algorithm. The *global_planner* node is responsible for planning the optimal route between the robot's current position and the goal based on the global cost map, integrating algorithms such as A* and Dijkstra to find the shortest route between the points. The *local_planner*, is responsible for local route planning, i.e., small segments of the global route, based on the local cost map. Using the information provided by the robot odometry and LiDAR, the *local_planner* sends speed commands to the robot controller to reach the goal. The speed commands needed for obstacle avoidance are sent to the robot controller through *recovery_behaviours* [57].

4.2.3 Creating the Fiducial Markers

The *ar_track_alvar* package allows the creation of fiducial markers of different sizes, resolutions, and encoding IDs. It is possible to identify and track the position of individual markers by integrating an RGB camera, or by using kinetic depth information for better estimations, being also possible to group markers for even more accurate position and

orientation estimates [61].

The markers of the *ar_track_alvar* package are designed to be identified in different lighting conditions and viewing angles, and are characterized by being formed by a 5x5 square of black and white pixels surrounded by a black border, which allows their detection through image segmentation and edge recognition [33]. For the markers to be identified correctly by the artificial vision strategy intrinsic in the package it is necessary to perform the camera calibration, and indicate the size of the marker to be identified. The package makes use of the detected corner points to extract a patch of cloud points that represent the marker, so it is possible to compute the relative real-world distance between the camera and the markers, as well as the orientation by aligning the centroid of the image with the center of the marker [30], [62].

The calculated position and orientation information is published to a ROS topic called *ar_pose_marker*, messages sent by this topic have a header that specifies the ID of each marker identified, the time corresponding to the information, the reference frame, and a message of type PoseStamped containing the position and orientation information relative to the markers and the camera [29], [33].

4.3 Setting up a ROS Network

Something that has been talked about a lot throughout this chapter, even if indirectly, is that most of the approaches, packages, and strategies used demand a lot of computational resources. From the particle filtering process of AMCL, or the marker processing done by *ar_track_alvar*, and the execution of the V-REP simulator, this can demand too much of the CPU, something that affects the real-time data processing.

But since ROS is a framework designed to enable distributed computing, it is possible to divide up the nodes between different devices that can communicate over a network. Detailed information on setting up a ROS network can be found in the documentation on this topic on the ROS Wiki page [63]. For communication between different devices to be possible, it is necessary that one of them acts as a ROS Master, providing logging services

for all existing nodes, and that all other devices are connected to the same Master, and the devices must know the hostnames and the Internet Protocol (IP) addresses of the other devices.

For the location system tests, two computers connected to a ROS Network established by WiFi connection were used. The distribution of processing between the two computers is illustrated in the Figure 4.10. One of the computers acts as the ROS Master, and it has been assigned to run the V-REP simulator. In the simulator, the environment information is collected by the sensors on board the mobile robot, the LiDAR and the RGB camera, in addition to the robot's odometry information. This information is published in topics and is available to be received by the second device. The first computer still receives the speed messages from *move_base* to act as the robot's controller, thus performing the navigation.

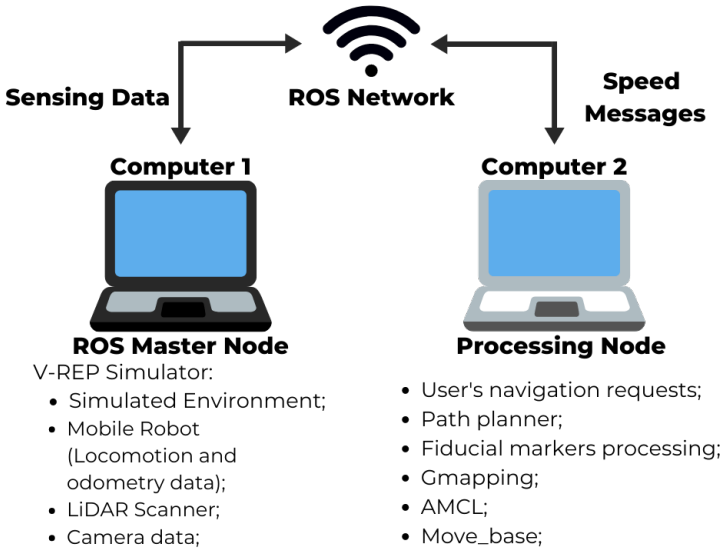

Figure 4.10: Distributed computation of the localization system processes between two computers in a ROS network.

The second computer was destined for data processing tasks, thus being responsible for the execution of gmapping during mapping processes, and in tests involving the localization system, being in charge of processing the AMCL, *move_base* and *ar_track_alvar* algorithms, being the last one responsible for sending its information to the algorithm

implemented for the localization correction based on fiducial markers, which is also executed in the second computer. This device also receives requests from the user with the robot's destination, and runs a path planning algorithm to define some waypoints to be sent to *move_base*.

4.4 Integrating the Localization System with ROS

Having already introduced the ROS network concepts, the simulation environment and the ROS packages used in the implementation of the proposed system, it is necessary to understand how all these aspects are connected to the system architecture, and how the system actually works.

For the mapping process the gmapping package is used to create the grid occupancy map. The */vrep* node provides to the system via the */vrep/info* topic the simulation time, and also the LiDAR sensor information which is published through the */scan* topic, which send the information to the */slam_gmapping* node responsible for interpreting it and creating the map, for that, the node also receive a transform from the */odom_to_map* node, which transforms the odometry data to assist in the creation of the map. The */robot_state_publisher* node makes available the state information of the robot, representing a kinematic model. The process described is shown in the Figure 4.11.

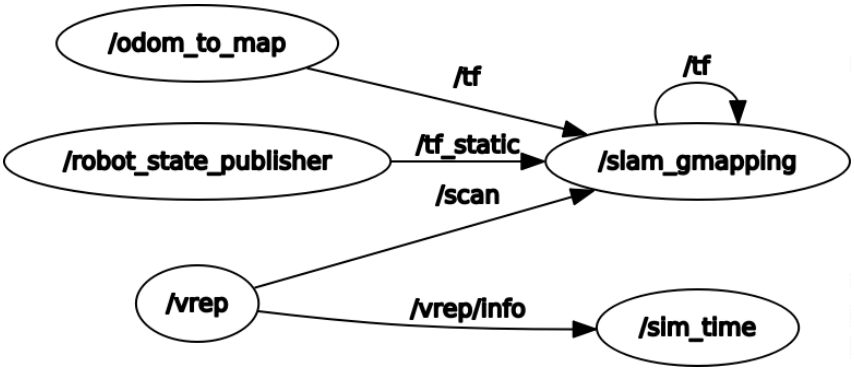

Figure 4.11: ROS topics and nodes during environment mapping using the gmapping package.

Having aimed to perform comparison tests between the proposed localization system

combining the AMCL particle filter and marker detection and the more traditional approach using only AMCL, the two versions were implemented and configured for testing in the simulation environment. The graph representing the topics and nodes ROS of the traditional approach is presented in Figure 4.12.

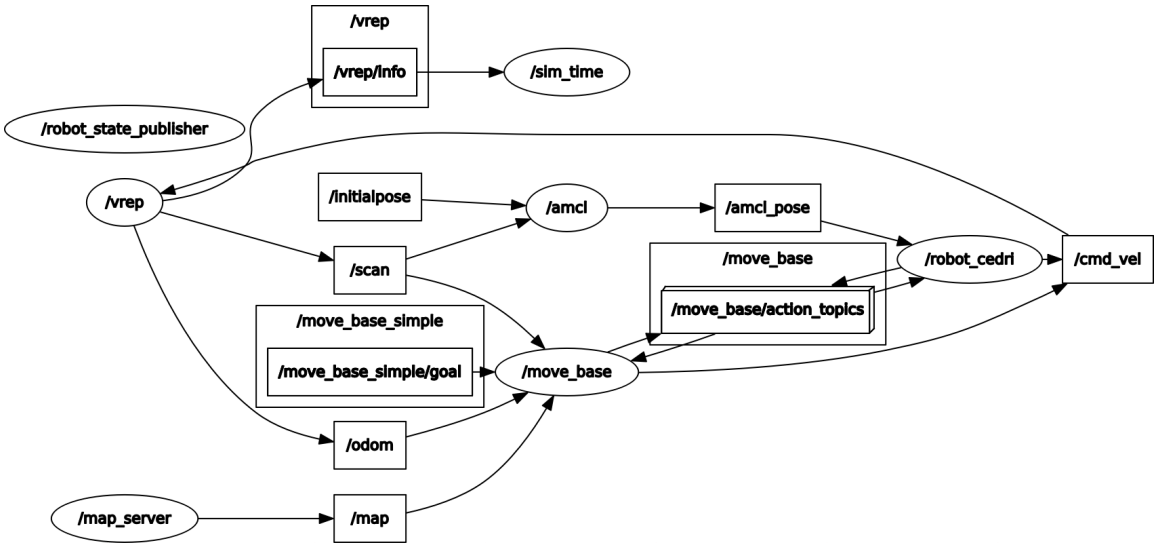

Figure 4.12: Graph representing the ROS topics and nodes of the mobile robot's navigation, with only AMCL as the localization algorithm.

The */robot_cedri* node is created by a python algorithm developed for collecting the goals for the robot entered by the user. In this algorithm when a destination for the robot is sent, it performs a path planning for the robot through a Dijkstra algorithm, so that the robot passes through some waypoints that were defined in a database in order to optimize the route possibilities for the robot, and that it can detect the fiducial markers in the proposed approach that will be discussed soon. The route planning performed by Dijkstra is based on the robot's current position, which is obtained from the AMCL position estimate received by the */amcl_pose* topic.

The */vrep* node provides the */vrep/info* topic which is used to report the simulation time to the */sim_time* node helping to keep the tfs needed for the system happening in real time. In addition */vrep* provides the */odom* and */scan* topics which are integrated with the */move_base* topic for route planning and */amcl* for position estimation.

The */map_server* node loads the grid occupancy map to the /map topic for use by

/move_base in route planning. The */initialpose* topic allows you to send an initial global position to */amcl* for the convergence of particles to this point. The */robot_cedri* node also acts in the goal sending topics to */move_base* receiving the control messages needed to control the robot, which are sent as velocity messages by the */cmd_vel* topic to the */vrep* node that will move the mobile robot in the simulation.

For the localization approach involving AMCL and the error corrections based on the fiducial markers, there are some more topics and nodes added for the operation of the system, with the complete system being presented in Figure 4.13. Now the */vrep* node also publishes to the */camera* topic the data collected by the camera positioned on the robot. The images are sent to the */ar_track_alvar* node which uses the package's software infrastructure to determine the relative distance between the markers and the camera.

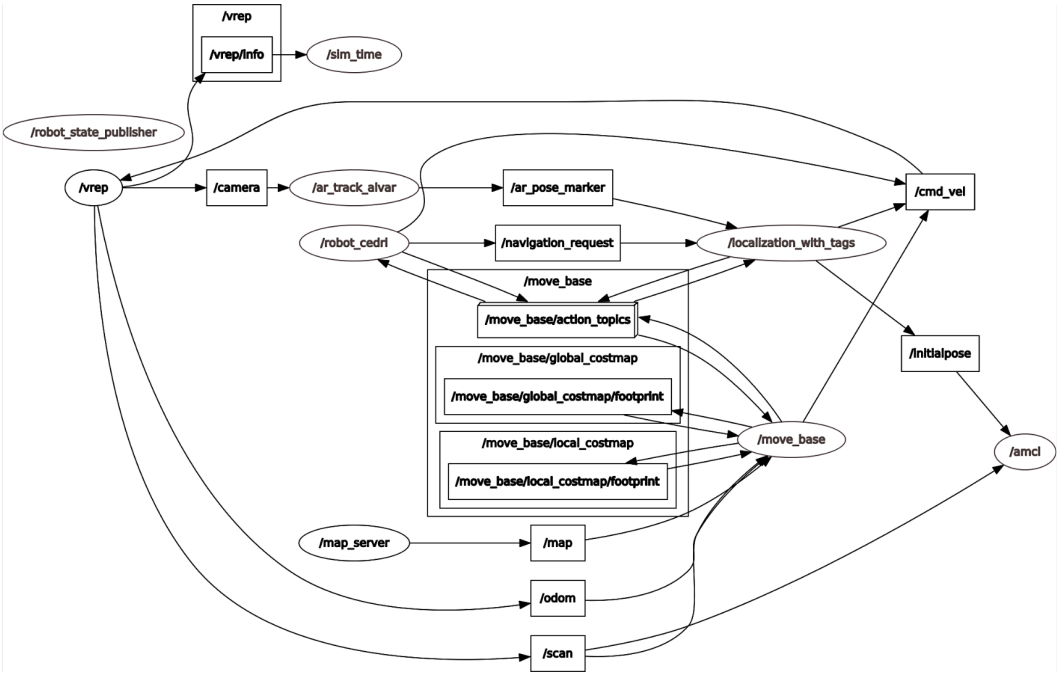

Figure 4.13: Graph representing the ROS topics and nodes of the mobile robot's navigation, with localization being performed by AMCL and fiducial markers.

The data processed by */ar_track_alvar* is published to the */ar_pose_marker* topic

and is received by the */localization_with_tags* node, this node is created by an implemented algorithm corresponding to the Algorithm 1 that was presented in the Chapter 3, being responsible for the calculations that determine the robot's global position based on the fiducial markers, performing the correction updates in the AMCL through publications in the */initialpose* topic. The */robot_cedri* node sends position update requests to the Algorithm 1 node through the */navigation_request* topic.

Chapter 5

Experiments with the Real Robot

This chapter will present the implementation of the proposed localization system for the tests in the real environment, introducing the mobile robot used and its main characteristics, as well as the modules that were added to the mobile platform and their purposes. The adaptations made to the hallways environment for the tests will be detailed, as well as the software changes made to the localization system algorithms and their integration with the framework, in order to adapt the system used in the simulated robot to the real robot.

5.1 The Mobile Robot Platform

The mobile robot used for the implementation of the system in the real environment was the Magni [64] mobile platform from Ubiquity Robots in its Silver model which is shown in Figure 5.1. The Magni platform was developed as a way to facilitate the development of robotic applications, having its control system based on ROS, the robot comes equipped with a Raspberry Pi 4 with an image based on the Ubuntu 16.04 Operating System with integration with ROS in its Kinetic distribution. Currently a new image based on the Ubuntu 20.04 and Noetic distribution of ROS is under development by Ubiquity Robotics, with no definite release date.

The robot is designed to move in various environments having a differential traction

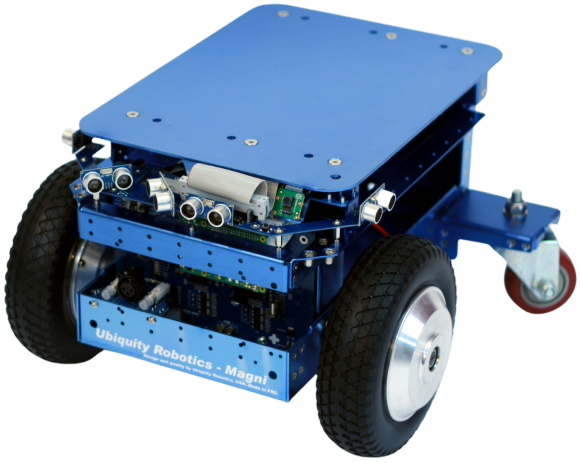

Figure 5.1: Magni robot silver model from Ubiquity Robotics.

system, and supporting a payload of up to 100 kg. The robot can reach a maximum speed of 1 m/s limited by software, and its odometry is based on a Hall sensor with an accuracy of 2 mm. It can be powered by 24 V batteries, providing 12 V and 5 V power supplies for powering external hardware. The robot also comes equipped with a 5 point Sonar array and a camera module for Raspberry, which can be used to aid in the navigation of the robot [64].

As mentioned the Magni platform has integration with ROS, this allows that the software structure developed for the simulation tests to be applied to the robot as well, without any major changes being necessary, since the communication between the modules remains the same. Also in the official GitHub repository of Ubiquity Robotics [65], it is possible to find several packages that can be applied not only for the communication system with the motor node, but also algorithms that allow to control the robot by teleometry or even its navigation through a system based on the tracking of fiducial markers.

5.2 External Hardware

To incorporate the proposed indoor localization system into the Magni robot, it was necessary to add some extra external hardware components to its structure. Each of

these components, their specifications and functions will be discussed below. The Figures 5.3 and 5.2 shows the mobile robot with the added hardware modules.

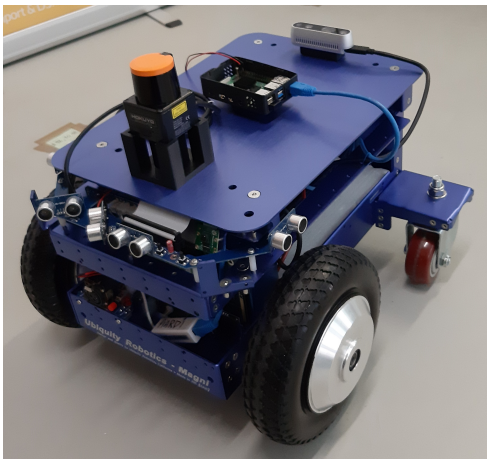

Figure 5.2: View of the Magni robot and on-board modules.

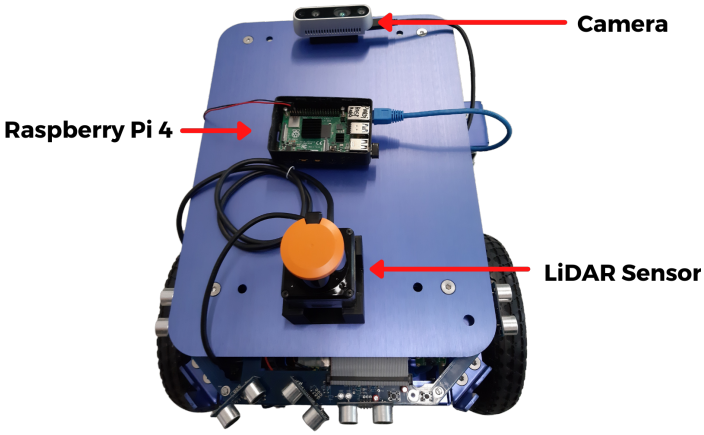

Figure 5.3: Hardware modules added to the mobile robot.

5.2.1 Raspberry Pi 4 Model B

To perform the robot control and operation of the ROS framework, it is necessary to have some computational component acting as a processor. For this, two Raspberry Pi 4 model B were used, characterized as low-cost single-board computers but with excellent processing power and widely used in automation, Internet of Things (IoT), and robotics applications. The Figure 5.4 shows an image of the microcomputer.

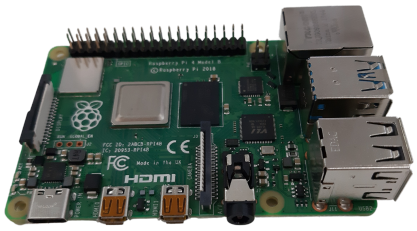

Figure 5.4: Raspberry Pi 4 Model B.

The microcomputer is available in 1, 2 or 4 GB RAM versions, having a Quad core 64-bit ARM-Cortex A72 processor with 1.5GHz clock. It has two micro-HDMI ports

supporting displays up to 4K60p resolution, two USB2 ports, two USB3 ports, a microSD slot, an Ethernet connection port, and integrated Bluetooth 5.0 and Wireless connection technology [66].

Both Raspberry used have an Ubuntu 16.04 image with the Kinetic version of ROS, to split the processing of the localization system over a ROS network. One of them is connected to the main control board of the robotic platform as shown in the Figure 5.5, while the other was positioned on top of the robot, as shown in the Figure 5.3, is powered by one of the 5 V power outputs of the robot control board, being connected to pins P04 (VCC) and P06 (GND).

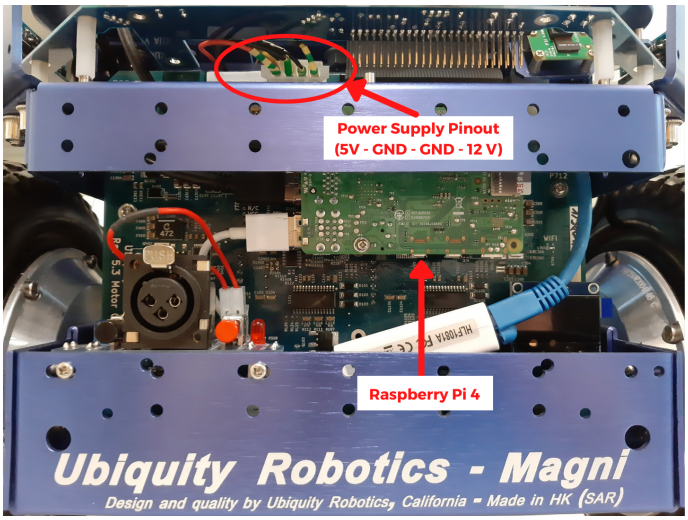

Figure 5.5: Raspberry connected to the magni robot control board and power supply pinouts from the control board.

5.2.2 LiDAR UST-10LX

The range-based sensor chosen to be used in the robotic system, for the data collection needed to perform the mapping process by gmapping and for the localization of the robot by the SLAM approach based on the AMCL particle filter, was the LiDAR Smart-URG mini model UST-10LX from Hokuyo Automatic, which is shown in Fig 5.6.

This model with ethernet-based communication interface has a higher cost 12 times higher than the RPLiDAR presented in Chapter 2, but it is a more robust LiDAR with

Figure 5.6: LiDAR UST-10LX from Hokuyo.

a higher range and resolution, which guarantees more accurate data reading and reduces the errors generated in SLAM. Some of the main technical information is presented in Table 5.1.

Specification	UST-10LX	Unit
Accuracy	±40	mm
Scan Angle	270º	deg
Repeated accuracy	$\sigma < 30$	mm
Detection range	0.06 to 10	m
Angular Resolution	0.25º	deg
Supply voltage	12/24	VDC
Supply current	150 (450 during start up)	mA

Table 5.1: Sensor LiDAR UST-10LX specification [67].

The LiDAR is connected to the Raspberry integrated magni robot control board via the ethernet port, and is powered by one of the robot's 12 V power supplies. Internal communication between the LiDAR and the ROS framework is possible via the urg_node package, whose documentation can be accessed on the ROS Wiki page [68].

5.2.3 Intel RealSense D435i

To perform the detection of the fiducial markers positioned on the ceiling of the environment, an Intel RealSense D435i depth camera was attached to the Magni platform, the

Figure 5.7 shows a view of the camera. This camera is widely used in augmented robotics and augmented and virtual reality applications, this is a depth camera with two depth sensors, an infrared projector, an RGB sensor, and an USB port. Some other technical specifications are shown in Table 5.2.

Figure 5.7: Intel RealSense Depth Camera D435i.

Specification	RealSense D435i	Specification	RealSense D435i
RGB frame resolution	1920 x 1080	Depth resolution	1280 x 720
RGB frame rate	30 fps	Depth frame rate	90 fps
RGB sensor resolution	2 MP	Depth Accuracy	<2% at 2 m
RGB sensor technology	Rolling Shutter	Depth range	0.3 to 3 m (ideal)

Table 5.2: Intel RealSense D435i specification [69].

Although the *ar_track_alvar* package allows the use of depth to better estimate the position of the markers, given its higher computational power, it was chosen to use only the RGB sensor of the camera. The communication and power supply of the camera is done by USB and it is connected to the raspberry on the magni robot control board. The integration of Ralsense D435i with the ROS framework is facilitated by the realsense-ros package [70].

5.3 Adapting the Environment

Before the tests of the proposed localization system with the robot could be carried out, it was necessary to fix the fiducial markers on the ceiling of the hallways environment. The distribution of the markers followed the coordinates used in the simulation environment and listed in the Table 4.1, and in Figures 5.8 and 5.9 some of the markers placed in the environment are shown.

Figure 5.8: Fiducial markers fixed on the celing of the hallways in the robotics laboratories area.

Figure 5.9: Fiducial markers fixed to the celing of the main hallway near the entrance.

Fixing the markers proved to be an arduous and complex task, given the difficulty in taking measurements from the ceiling, to make sure that each marker was as close as possible to the desired position. But the biggest challenge is to ensure that the orientation of the markers is correct, and it will be necessary to develop a method in the future to certify with precision that the orientation of the marker is correct.

Besides fixing the markers to the ceiling, it was necessary to make some small adaptations in the environment. The doors in the hallways have parts made of glass, which are penetrated by the light beams emitted by the LiDAR sensor. To prevent this from occurring and adding too much noise to the SLAM system, paper strips were positioned on the doors at the height of the laser sensor, as illustrated in the Figures 5.10 and 5.11.

5.4 Adapting the ROS Network

The ROS Network had to be subjected to some changes during the implementation of the localization system in the real robot when compared to what was implemented in

Figure 5.10: Paper band positioned on the access door to the computer hallway.

Figure 5.11: Paper band positioned on the access door of one of the laboratories.

simulation. Here the processing is no longer split between two computers, but between two Raspberry Pi's, the one connected directly to the robot control board, which will be referred to as Raspberry 1, and the second one positioned on top of the robot as presented previously in Figure 5.2, which will be referred to as Raspberry 2.

In simulation testing, there was concern about processing mainly due to the V-REP simulator, which is not needed in the real tests. However, the Raspberry has a lower processing capacity than the computers used previously, furthermore, the *ar_track_alvar* and the navigation stack still demand a lot of computational power. Based on tests and monitoring the CPU usage of the Raspberry, the distribution of processes over the ROS network is as shown in the Figure 5.12.

The Raspberry 1 is responsible for all the sensing, collecting information from the LiDAR sensor, the camera, and the odometry, communicating with the robot's control system, and processing the images through *ar_track_alvar*, to determine the robot's position based on the fiducial markers, and updating the AMCL. While the Raspberry 2 is responsible for receiving the navigation requests by the user and performing the path

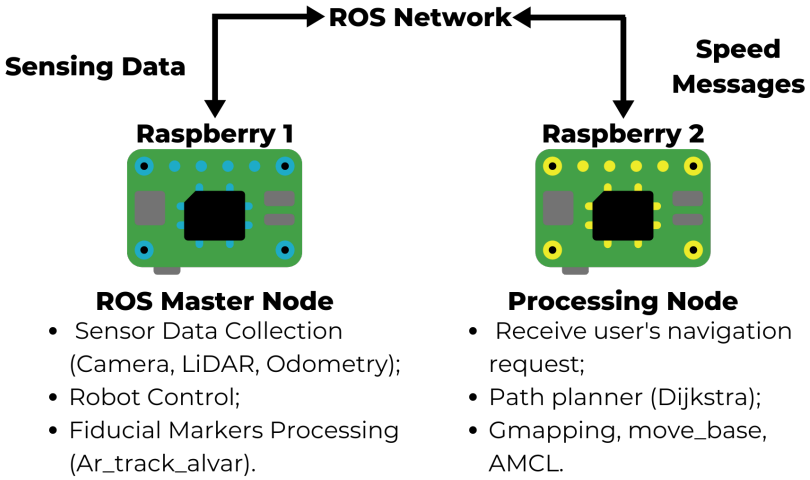

Figure 5.12: Distributed computation of the localization system processes in the real robot.

planning with the Dijkstra algorithm based on the desired destination and the current position of the robot. Besides performing all the necessary computing for the corresponding nodes of the *move_base*, AMCL and gmapping packages.

During the simulation tests the two computers were connected to the same Wi-Fi network which was the channel for the ROS network. However, in the real environment, the robot is navigating through the hallways and the connection to the Wi-Fi network is not always stable, being interrupted frequently. As it is important to keep the ROS network working so that the message exchange between the two Raspberry is not interrupted at any time, harming the localization system, the ROS network was configured through a direct Ethernet connection between the two Raspberry ensuring the permanence of the message exchange and even decreasing delays that were happening in the approach involving Wi-Fi.

For the initialization of the ROS and the tracking system, as well as for sending the destinations for the robot's navigation, an SSH connection between a computer and the Raspberry is established over Wi-Fi. Since these are one-time execution tasks, instability of the Wi-Fi network does not impact the operation of the system.

Chapter 6

Results and Discussion

This chapter will present the results obtained from the tests performed in the simulation environment and with the real robot. The main points observed regarding the performance of the different localization systems, both the traditional approach using only the AMCL particle filter and the proposed approach that integrates the fiducial markers with the particle filter, will be discussed, in order to analyze their effectiveness when applied as part of an autonomous navigation system.

6.1 Simulation Results

This section will discuss the results obtained for the tests performed in the simulation environment created in V-REP. The two localization systems were tested in different scenarios related to traditional mobile robot localization problems and in different trajectories, in order to compare their performance in each of these scenarios.

6.1.1 Generated Map of the Simulated Environment

First the mapping process of the simulated environment of the CeDRI hallways was performed. With the help of an implemented python algorithm, speed messages were sent to the */cmd_vel* topic of the mobile robot, which is responsible for controlling its

locomotion system. Thus, the robot was teleoperated through the entire extension of the environment in order to collect the necessary information for the creation of the map through the readings of the LiDAR sensor.

The LiDAR data, as well as the odometry information, were interpreted by the gmapping package node, which performed the SLAM process, simultaneously localizing the robot while generating the environment map, according to the robot's movement. Figure 6.1 shows the obtained grid occupancy map.

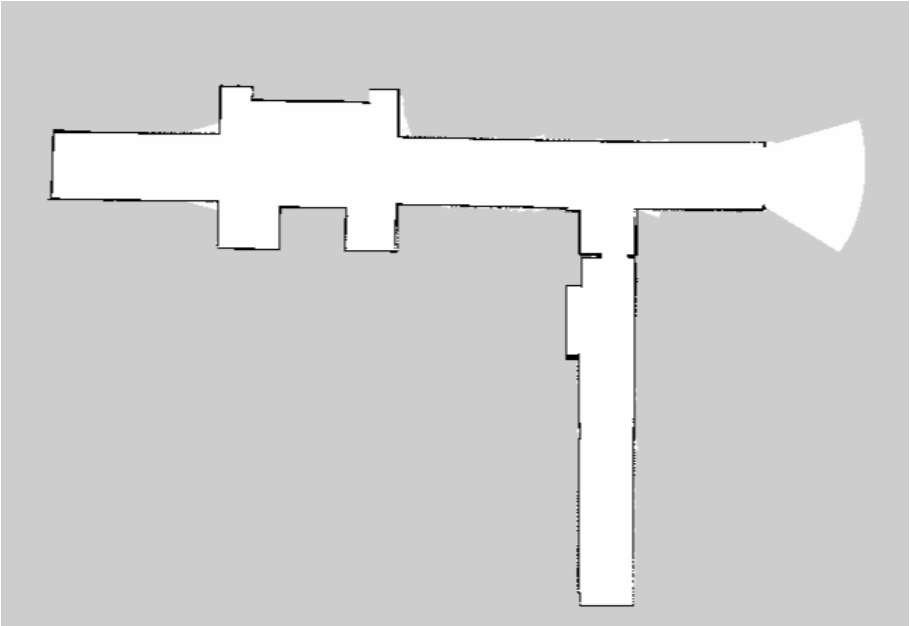

Figure 6.1: Generated map of the hallways in the simulation environment.

The map generated by gmapping proved to be quite accurate in relation to the simulation environment, presenting only a few small discrepancies in the portrayal of the walls of the environment, and a rotation error in the representation of the map in relation to the environment. This accuracy is an important factor so that the map could be adopted as a reference to perform the localization of the robot through the AMCL algorithm.

6.1.2 Proposed Testing Scenarios

To conduct the tests, four different scenarios were defined to be covered, so that the two different localization systems could be tested on different localization problems. Which

one of the scenarios are detailed below:

- **Position tracking using only the AMCL algorithm (Scenario 1)**: in this first scenario the problem involving position tracking, which was introduced in Chapter 2, is handled. In it the robot's initial position is known to the robot, in this way the AMCL algorithm will start its particle filtering process trying to converge its particles to the position initially occupied by the robot, and will localize the robot based on the odometry information received during the robot's navigation, and the readings taken by the LiDAR sensor;

- **Position tracking using the AMCL algorithm and the fiducial markers (Scenario 2)**: as in the first scenario, the robot knows its initial position and performs its localization through the SLAM approach involving the AMCL algorithm, but in this case the proposed localization system is applied, so updates based on the detection of the fiducial markers are performed, aiming to remove the intrinsic errors accumulated by the AMCL position estimation;

- **Global localization problem, performing localization only with the AMCL algorithm (Scenario 3)**: in this case the global localization problem, which was also discussed in Chapter 2, is tested. Since the robot doesn't know its initial position, the AMCL algorithm will start its global localization service, spreading its particles all over the environment in an attempt to determine the robot's true position, based on the matching between the LiDAR sensor readings and the existing landmarks on the map;

- **Global localization problem, using the AMCL algorithm and the fiducial markers for localization (Scenario 4)**: in the last scenario, the robot will also not know its initial position, however, the fiducial markers are expected to act as global reference points, aiding in the convergence of AMCL particle filter to the robot's current position, as well as eliminating the position estimation errors performed by AMCL once the position is known.

Each of the four scenarios were tested on three different routes defined to be navigated by the robot in the environment, which are shown in Figure 6.2.

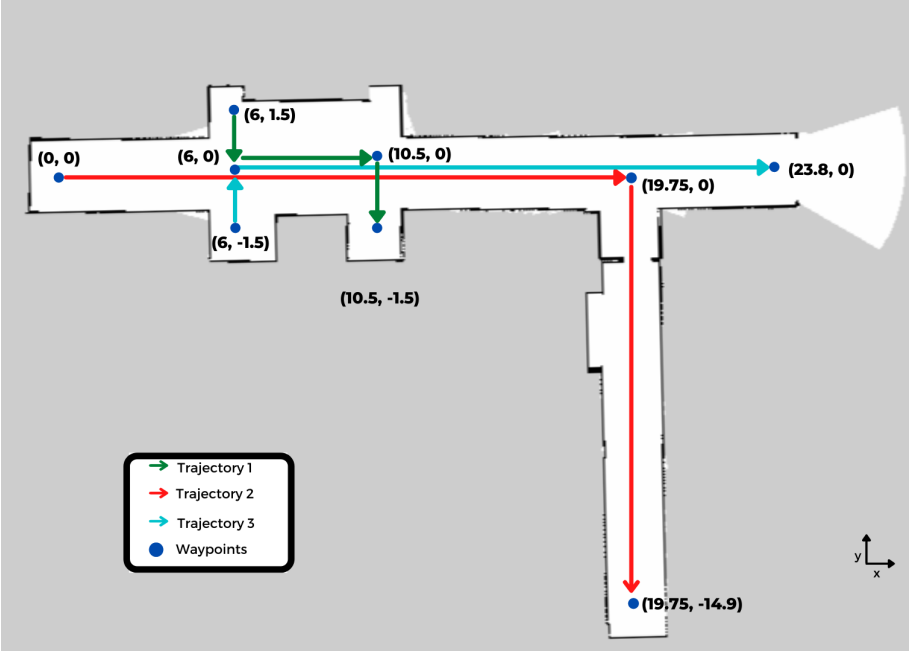

Figure 6.2: Routes used for the robot navigation during the localization tests.

6.1.3 Testing the Position Tracking Problem Scenarios

As mentioned, the first two scenarios are related to the position tracking problem, since the robot has knowledge of its initial position. Using the V-REP simulator the two scenarios were tested for each of the desired trajectories.

Figures 6.3 and 6.4 show the navigation performed by the robot for the first trajectory in scenarios 1 and 2 respectively. In the figures, the green arrows represent the position and orientation of the robot at different time instants, obtained through the simulator's ground truth, corresponding to the actual position of the robot in the environment. While the red arrows correspond to the robot's position and orientation estimation performed by the AMCL algorithm. The black dots in Figure 6.4 correspond to the position updates performed by detecting the fiducial markers. The graphs represent the trajectories and

estimates starting at the first estimation update performed by the AMCL algorithm, disregarding the initial instant when the error is equivalent to zero, since the initial position is informed to the system.

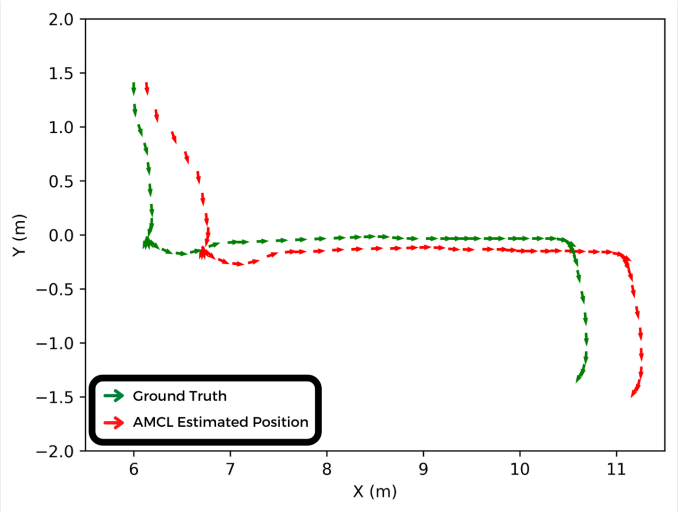

Figure 6.3: Navigation performed by the robot for trajectory 1 of scenario 1.

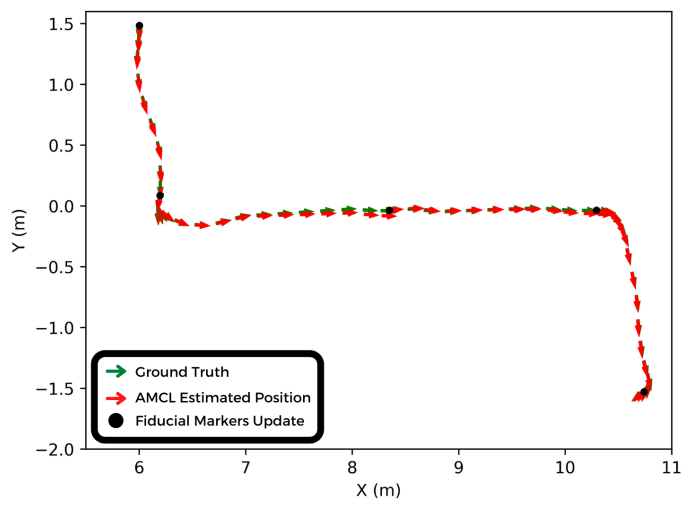

Figure 6.4: Navigation performed by the robot for trajectory 1 of scenario 2.

It can be seen in Figure 6.3 that although the AMCL algorithm maintains its estimates relatively close to the real position occupied by the robot, there is an existing divergence, even in the initial convergence of the algorithm after the beginning of the robot's movement, which increases at times when the robot motion control exerted by

move_base performs angular movements, thus accumulating errors along the path. With the proposed localization system, seen in Figure 6.4, which integrates periodic position updates, a discrepancy still occurs; however, the updates allow the errors to be kept at lower values. The difference between the estimated position and the real position is reduced as soon as a new update is performed.

The tests for both scenarios were also performed for the second defined route, and the results are shown in Figures 6.5 and 6.6. Again AMCL was able to perform a good localization of the robot, however it is noted that the results obtained by the system with the fiducial markers were more accurate. There is only a higher discrepancy in the curve performed by the robot near the waypoint coordinates (19.75, 0), which may be related to the existing angle between the robot and the marker at the moment of the update, since different orientations between the camera and the marker can reduce the accuracy of the distance estimation between them [27].

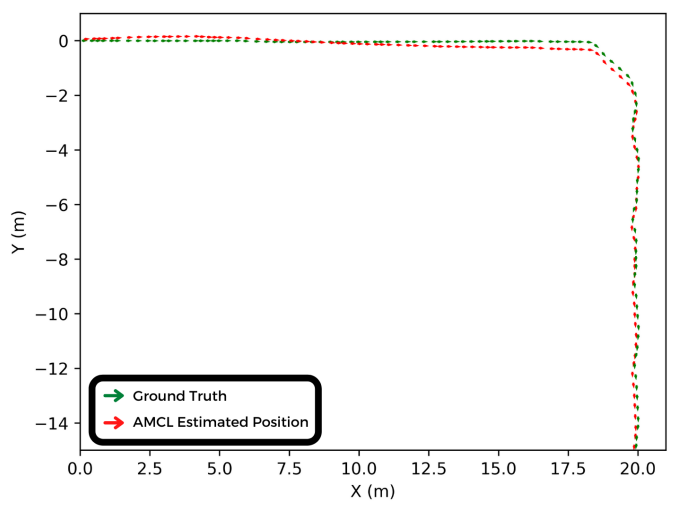

Figure 6.5: Navigation performed by the robot for trajectory 2 of scenario 1.

Figures 6.7 and 6.8 show the navigation of the mobile robot, performed for the third trajectory. In this path it is possible to observe that there was a higher discrepancy in the localization for the first localization system, and remaining a higher accuracy for the system with the markers.

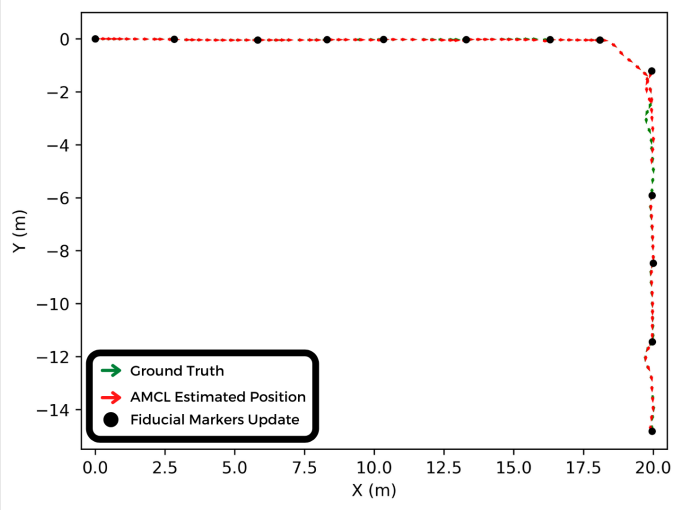

Figure 6.6: Navigation performed by the robot for trajectory 2 of scenario 2.

Table 6.1 shows the errors obtained by each of the algorithms on the different trajectories, Where the Gain is represented by the achieved percentage improvement over the average error between the position estimates obtained by the proposed localization system compared to the traditional approach, as represented by Equation 6.1.

$$Gain = \frac{Ea_{AMCL} - Ea_{Markers}}{Ea_{AMCL}} * 100 \tag{6.1}$$

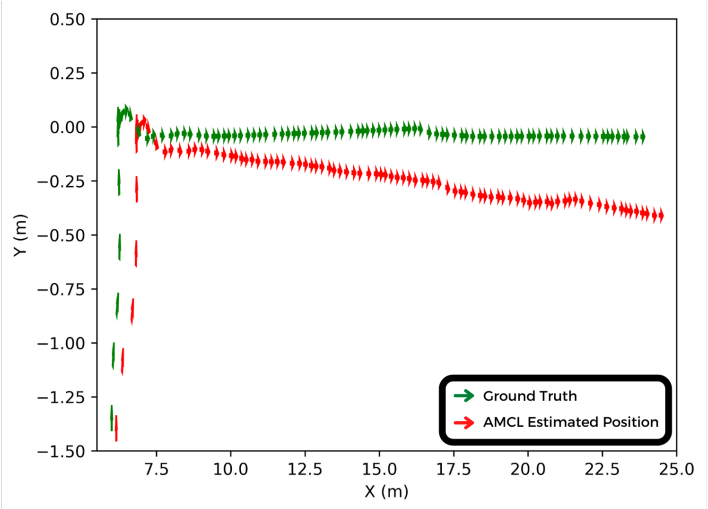

Figure 6.7: Navigation performed by the robot for trajectory 3 of scenario 1.

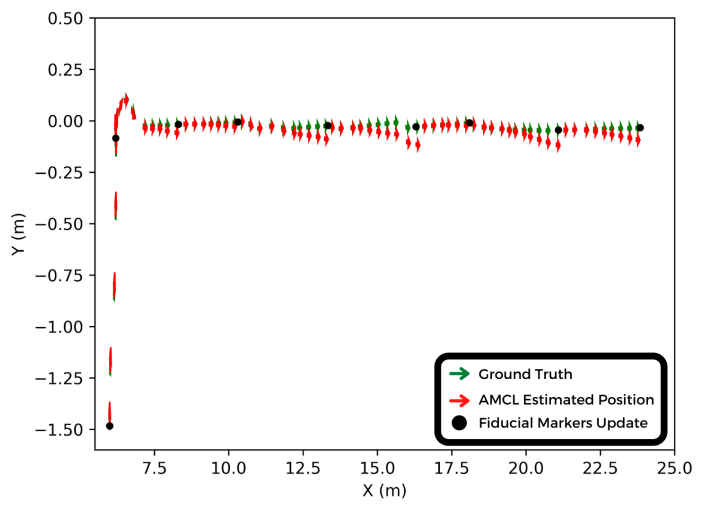

Figure 6.8: Navigation performed by the robot for trajectory 3 of scenario 2.

	Minimun Error		Maximun Error		Average Error (Ea)		Gain
	AMCL	AMCL Markers	AMCL	AMCL Markers	AMCL	AMCL Markers	
Trajectory 1	0.13m	0.00m	0.64m	0.05m	0.56m	0.01m	98.21%
Trajectory 2	0.09m	0.00m	0.34m	1.15m	0.21m	0.11m	47.62%
Trajectory 3	0.16m	0.01m	0.75m	0.09m	0.63m	0.02m	96.82%

Table 6.1: Errors of the localization systems for the tests of the position tracking problem.

It can be seen that in all the tests performed, the proposed localization system performed better when compared to the same test with the system using only the AMCL algorithm. The worst performance of the marker system was on the second trajectory, where there was a discrepancy in one of the updates, as previously discussed.

There are a few factors that may have contributed to the localization performed by the SLAM approach based on AMCL, obtaining such excellent results. The first factor is that the simulation used considered only a static environment, with no moving objects or obstacles moving through the environment, something that could affect the matching between the LiDAR sensor readings and the characteristics represented in the map, making it more difficult for AMCL to converge its particles and consequently to correctly estimate the robot's position throughout the trajectory.

Another factor concerns the information collected from odometry and even from the

LiDAR sensor. In the simulation environment none of the sensors present errors in their readings, since all information is mathematically generated by the simulator. With real sensors the same does not happen, since the sensors present a rate of uncertainty and inaccuracy, and external external factors of the environment itself can impact the reading of these sensors.

6.1.4 Testing the Global Localization Problem Scenarios

For the tests involving the global localization problem, the initial position of the robot was not informed to the AMCL algorithm, which should estimate the position occupied by the robot based only on its particle filtering system for the third scenario. For the fourth scenario, besides particle filtering, the robot could obtain its position by detecting one of the fiducial markers scattered around the environment. Figures 6.9 and 6.10 shows the localization obtained by the two systems during the robot navigation on the first defined trajectory.

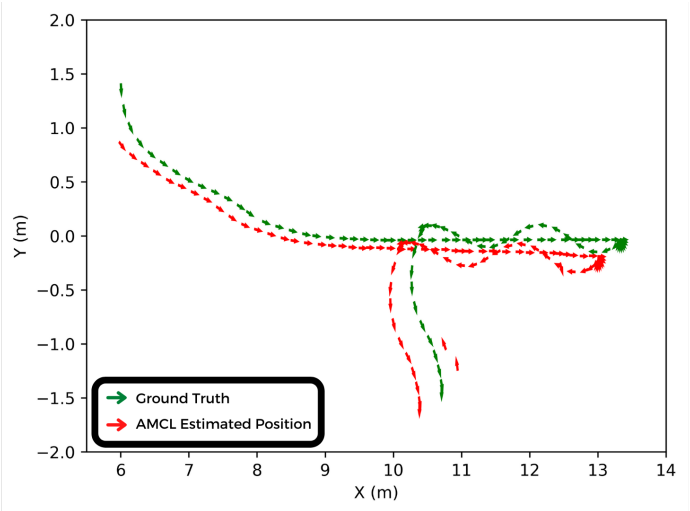

Figure 6.9: Navigation performed by the robot for trajectory 1 of scenario 3.

For the test with the localization system based on the AMCL algorithm, shown in Figure 6.9, it is possible to observe that initially the particles did not converge correctly to the robot's position, which caused the route planning to be impaired, sending the robot

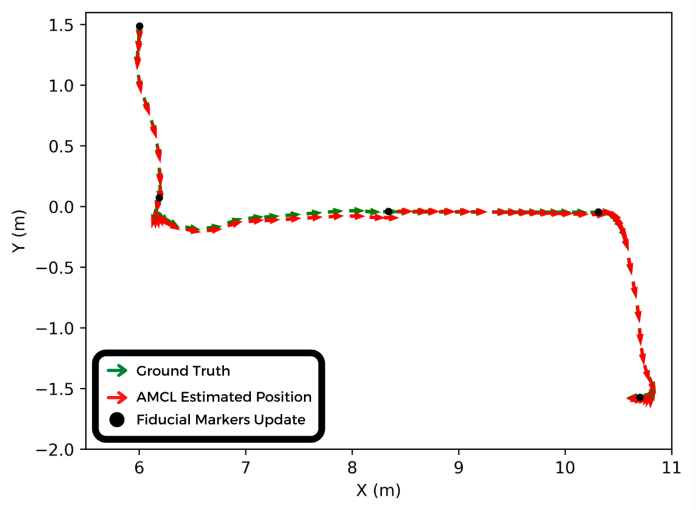

Figure 6.10: Navigation performed by the robot for trajectory 1 of scenario 4.

to another waypoint existing in the environment. Despite the route change, for the error calculations that will be presented in this section, only the existing discrepancies between the real and the estimated position were considered. For the test shown in Figure 6.10, since the robot obtained its initial position based on the fiducial markers, the robot's location remained accurate throughout the robot's navigation.

Figures 6.11 and 6.12 show the navigation performed by the robot for the second proposed trajectory.

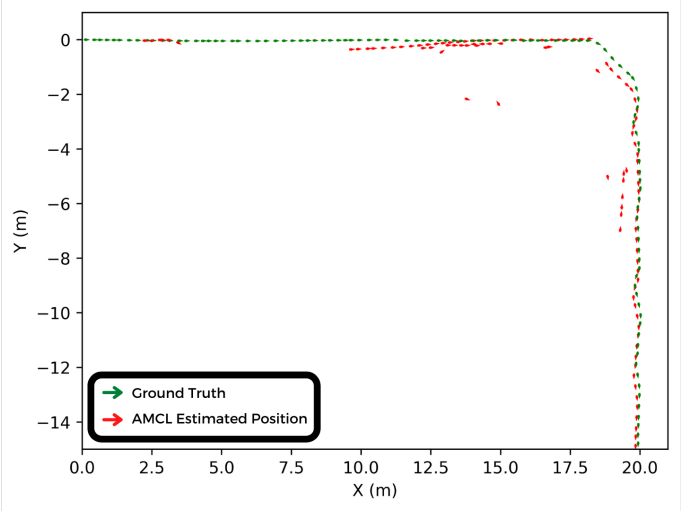

Figure 6.11: Navigation performed by the robot for trajectory 2 of scenario 3.

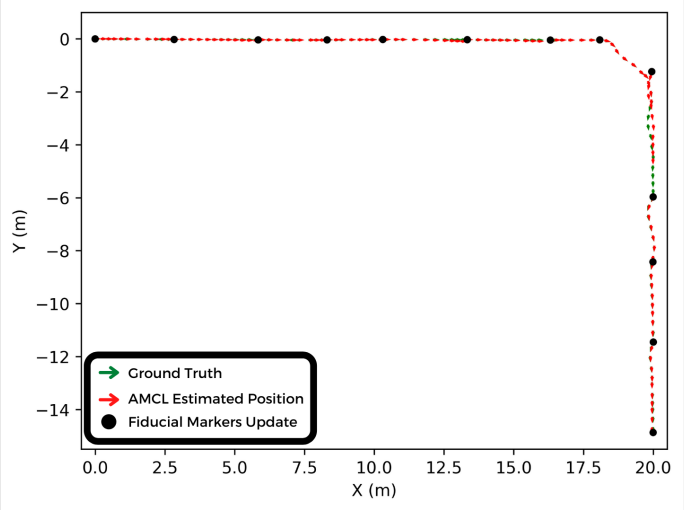

Figure 6.12: Navigation performed by the robot for trajectory 2 of scenario 4.

Similar to the previous test, the AMCL algorithm was slow to converge its particles to a location close to the robot's actual position. For the system with the fiducial markers, once again, because they offer a reference point for obtaining the initial position, the localization remained consistent, but there was an update error similar to the one that occurred in the test with the position tracking problem.

The results obtained during navigation on the third trajectory set for both systems are shown in Figures 6.13 and 6.14.

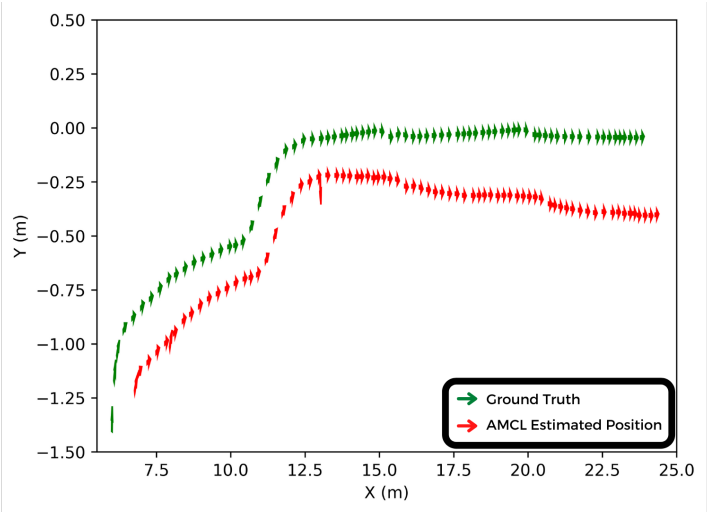

Figure 6.13: Navigation performed by the robot for trajectory 3 of scenario 3.

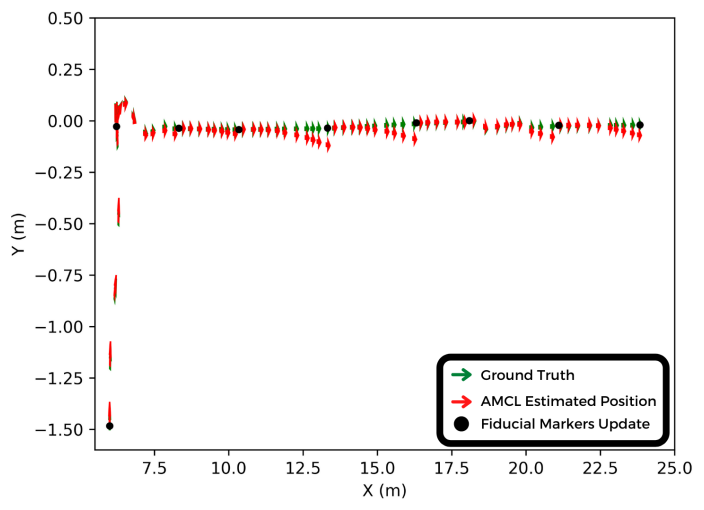

Figure 6.14: Navigation performed by the robot for trajectory 3 of scenario 4.

As expected, the results obtained by the proposed approach outperform the localization obtained by AMCL, which despite achieving an approximate position estimation for the robot, the discrepancies and accumulated errors are still significant.

The Table 6.2 shows the errors obtained by the systems in the tests performed in addition to the accuracy gain obtained by the system through the proposed approach. Given the distribution of the markers performed by the environment, it is possible to guarantee a more accurate and efficient position estimation for the robot compared to the particle filter global localization service.

	Minimun Error		Maximun Error		Average Error (Ea)		Gain
	AMCL	AMCL Markers	AMCL	AMCL Markers	AMCL	AMCL Markers	
Trajectory 1	0.24m	0.01m	6.65m	0.06m	0.51m	0.01m	98.04%
Trajectory 2	0.12m	0.00m	21.04m	1.17m	5.15m	0.11m	97.86%
Trajectory 3	0.53m	0.00m	13.05m	0.08m	0.67m	0.02m	97.01%

Table 6.2: Errors of the localization systems for the tests of the global localization problem.

6.1.5 Considerations About the Localization System in Simulation

Based on the results obtained from the tests performed in the simulation environment, the proposed localization system integrating the AMCL particle filter and the error correction updates based on the detection of the fiducial markers proved to be more effective than the method using only the particle filter, for the position tracking and global localization problems.

Using the fiducial markers as landmarks allows the robot localization system to achieve more accurate position estimation, and effectively helps eliminate intrinsic errors that can be accumulated by the SLAM approach with particle filtering. The processing of the fiducial markers by the *ar_track_alvar* package proved to be efficient for determining the relative distance between the markers and the camera, taking into account that the tests were performed disregarding lighting issues, which can end up impairing the effectiveness of the algorithm in the real environment.

Because the environment used in simulation is not subject to changes with dynamic obstacles, the AMCL algorithm achieved a good performance in estimating the position of the mobile robot, something that might not have occurred in a non-static environment, such as the real environment.

The V-REP simulator proved to be efficient for performing the tests and communicating with the ROS framework, however, it requires a large amount of processing power in order to use the necessary modules. Even through the processing division with the ROS network, where one of the computers (Intel Core i7-6500U CPU 2.6 GHz, 8 GB RAM) was exclusively dedicated to the software execution, there were numerous tests and processes that were interrupted due to the CPU overload.

6.2 Results with the Real Robot

This section will discuss the results obtained through the tests with the navigation of the Magni mobile platform through the hallways of CeDRI, as well as describe the main difficulties for the operation of the system in the real environment.

6.2.1 Generated Map of the Real Environment

Similar to the procedure performed in the simulation environment, for the creation of the grid occupancy map, the mobile robot was teleoperated through the corridors so that the */slam_gmapping* node could perform the mapping procedure based on the information collected by the robot odometry and the LiDAR sensor. The Figure 6.15 shows the map generated by the SLAM process.

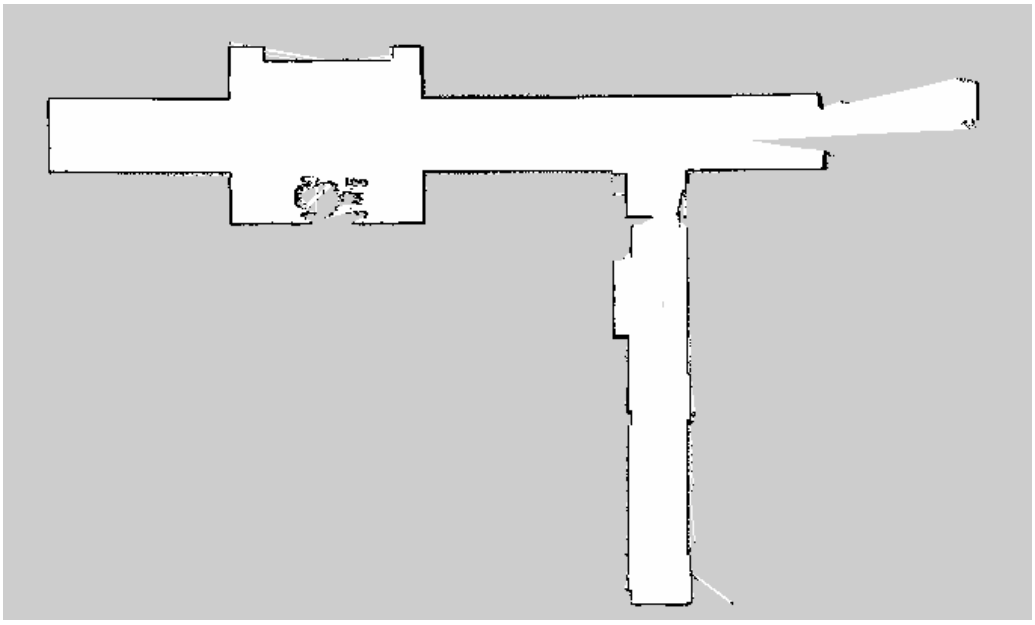

Figure 6.15: Generated map of the hallways in the real environment

Given the high accuracy of the Magni robot's odometry sensor, as well as the LiDAR UST-10LX the map created presents an excellent fidelity of the real environment, and closely resembles the map created in the simulation environment, shown in Figure 6.1.

6.2.2 Tracking Position Tests

Just as performed in simulation, to prove the effectiveness of the localization system integrating the fiducial markers and the particle filter, the four different scenarios and their three distinct trajectories were tested with the real robot.

Unlike in the simulation environment that it is possible to obtain a ground truth that accurately indicates the actual position occupied by the robot, the same cannot be achieved in the real environment. Therefore, since the markers have proven to be a robust way to determine the position of the robot, we will compare the errors obtained between the position estimated by the AMCL algorithm and the positions estimated based on the markers at the same time instants. In order to analyze the effect of marker-based updates in reducing the potential intrinsic cumulative position estimation errors.

Figures 6.16 and 6.17 show the results obtained for scenarios 1 and 2 in their first desired trajectory. The arrows represent the position and orientation of the robot estimated by the AMCL algorithm along the trajectory for the tests, the black dots represent the estimated position of the robot based on the markers and the purple "X" the position estimated by AMCL at this same instant of time. For the approach with the updates with the markers the accumulated error at the end of the path is smaller, than with only the particle filter being responsible for the localization process.

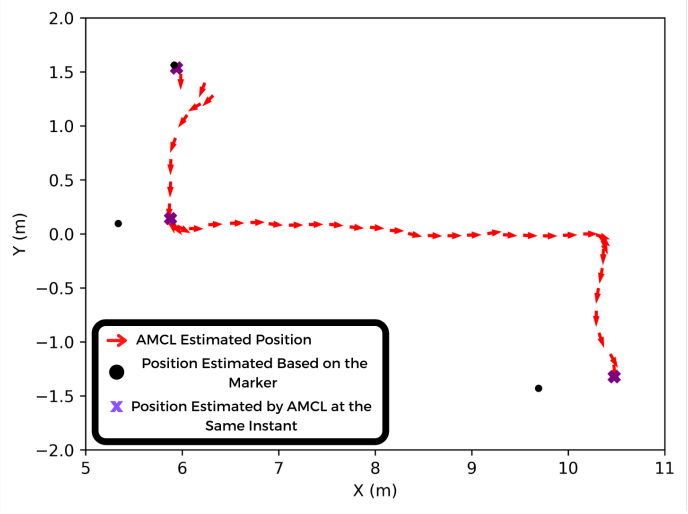

Figure 6.16: Position estimates for scenario 1 in its trajectory 1.

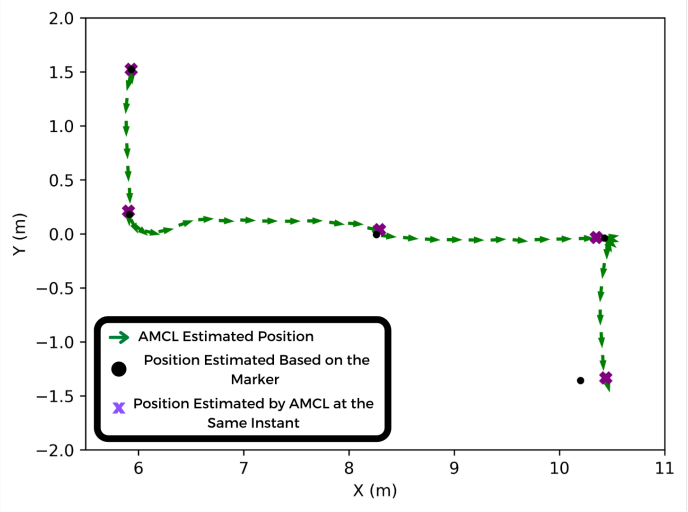

Figure 6.17: Position estimates for scenario 2 in its trajectory 1.

The position estimates for the second trajectory in both position tracking scenarios are shown in Figures 6.18 and 6.19.

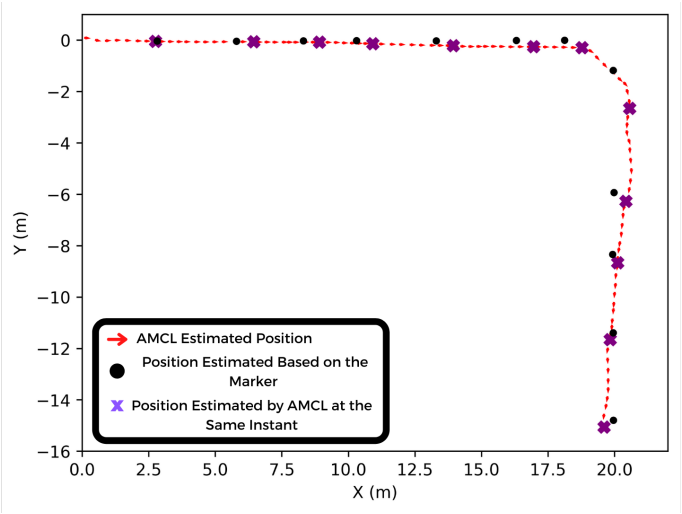

Figure 6.18: Position estimates for scenario 1 in its trajectory 2.

It is possible to notice that for the update algorithm based on the markers, there was also a discrepancy regarding the estimation based on the markers ID 12 and 13, as occurred in simulation. These markers, which were distributed in their respective positions to help the robot maintain its position estimate due to the existence of a narrow door that hinders the robot's passage, will have to be repositioned in the future, since the

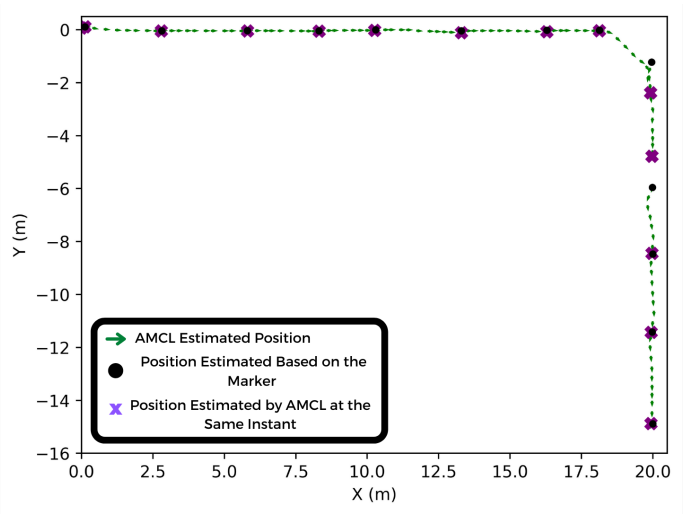

Figure 6.19: Position estimates for scenario 2 in its trajectory 2.

robot's entry angle defined by the *move_base* navigation system for this trajectory does not allow an accurate update of its position.

Figures 6.20 and 6.21, show the position estimation relationship of the two systems for the third trajectory, and it is noticeable the emergence of intrinsic errors accumulated along the path, which were smoothed out for scenario 2 due to the updates with the markers.

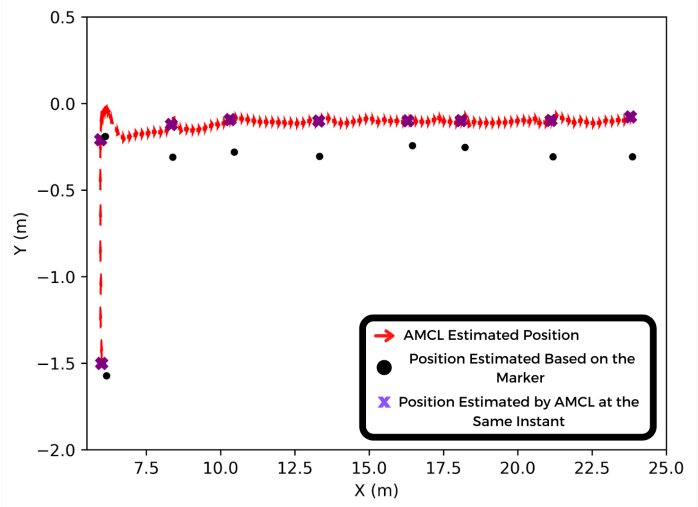

Figure 6.20: Position estimates for scenario 1 in its trajectory 3.

Table 6.3 disposes the errors present in the position estimation performed by AMCL

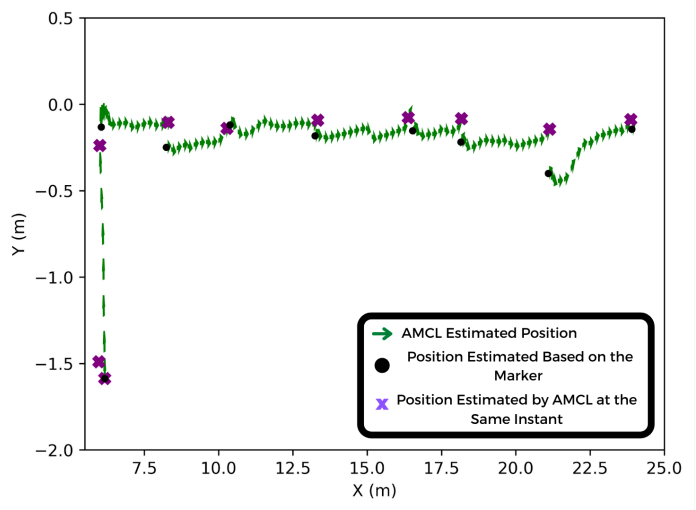

Figure 6.21: Position estimates for scenario 2 in its trajectory 3.

at the times when the marker-based position updates are performed in the case of the proposed system, and upon arrival at the waypoints in the case of the particle filter-only based system. There is improved accuracy for the proposed system, since the updates assist in eliminating intrinsic errors accumulated along the trajectory.

	Minimun Error		Maximun Error		Average Error (Ea)		Gain
	AMCL	AMCL Markers	AMCL	AMCL Markers	AMCL	AMCL Markers	
Trajectory 1	0.03m	0.00m	0.79m	0.24m	0.35m	0.08m	77.14%
Trajectory 2	0.05m	0.01m	1.60m	1.18m	0.61m	0.20m	67.21%
Trajectory 3	0.16m	0.01m	0.24m	0.26m	0.20m	0.13m	35%

Table 6.3: Position estimation errors relative to the markers in the position tracking tests.

6.2.3 Global Localization Tests

Finally, the tests involving the global localization problems were performed. Figures 6.22 and 6.23 show the position estimation carried out by AMCL for scenarios 3 and 4 in its first trajectory. For this trajectory, the system with the particle filter had difficulties to estimate the position occupied by the robot in the environment, without the initial position information to assist in the convergence of its particles, which ended up harming

the navigation of the robot that did not reach the desired final position. In the system with the markers, the robot was able to perform the navigation with an excellent accuracy ensuring the robot's arrival near the final destination.

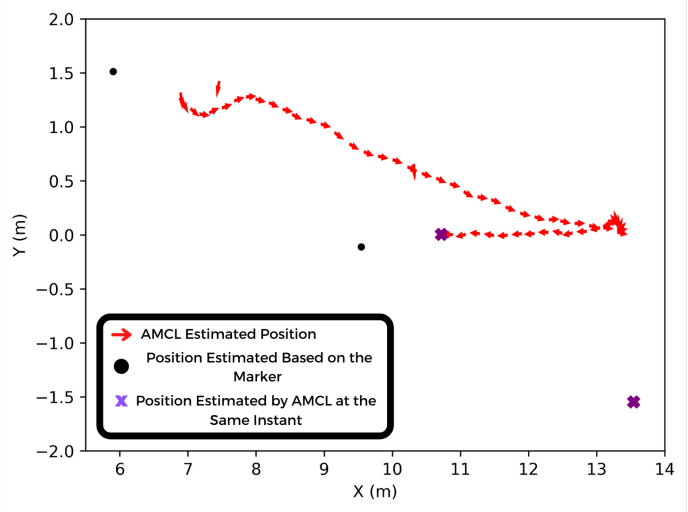

Figure 6.22: Position estimates for scenario 3 in its trajectory 1.

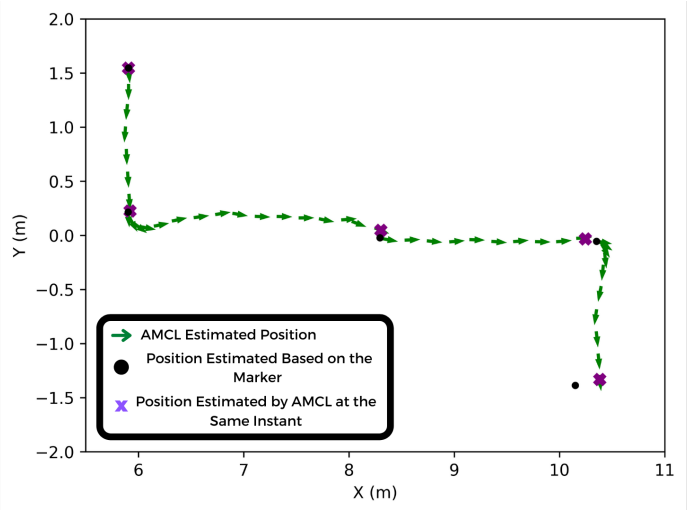

Figure 6.23: Position estimates for scenario 4 in its trajectory 1.

The position estimates for the second trajectory for the particle filter-based localization system are shown in Figure 6.24.

In it the robot was able to reach the end point, and the AMCL algorithm was able to estimate the robot's position accurately, however, the convergence of the particles took

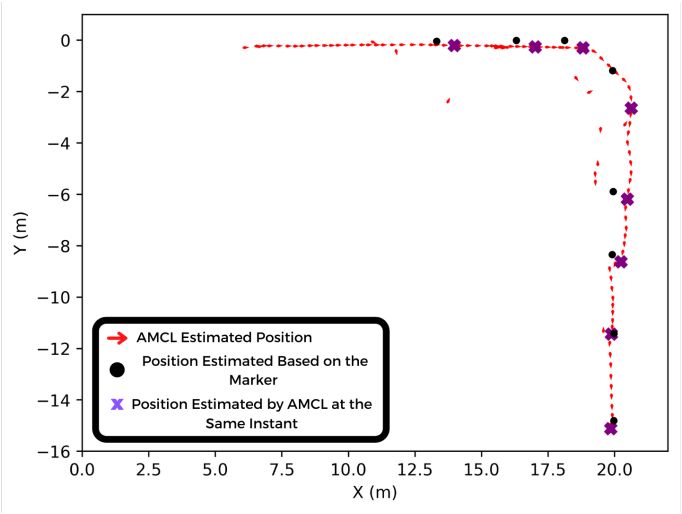

Figure 6.24: Position estimates for scenario 3 in its trajectory 2.

a while to occur, indicating the robot's real position after it had traveled more than 5 meters of its trajectory, which corresponded to a stretch of the corridor without many features, which made the convergence of particles difficult. The error that will be presented later ends up disregarding this stretch in which the algorithm had not yet performed its convergence, since the system also failed to perform the position collection based on one of the markers so that a comparison could be made. Thus the true value of the error should be much higher than what can be calculated. For the test with the fiducial marker updates, which is shown in Figure 6.25, the system was able to determine the location of the robot during the entire trajectory.

Figure 6.26 and 6.27 show the results obtained for the third trajectory defined. The system based on the particle filter had a discrepancy in its particle convergence, causing the robot to finish its navigation before the final destination. The system based on the markers in turn allowed the robot to reach its final destination, besides reducing the intrinsic error accumulation throughout the navigation, thus ensuring better and closer to the desired results.

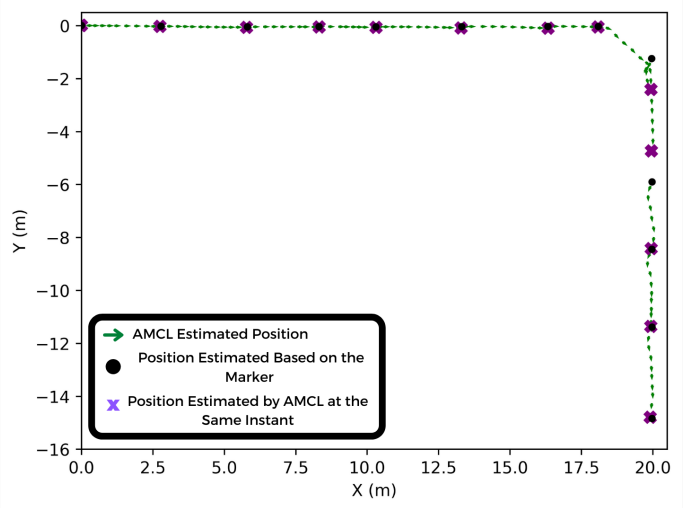

Figure 6.25: Position estimates for scenario 4 in its trajectory 2.

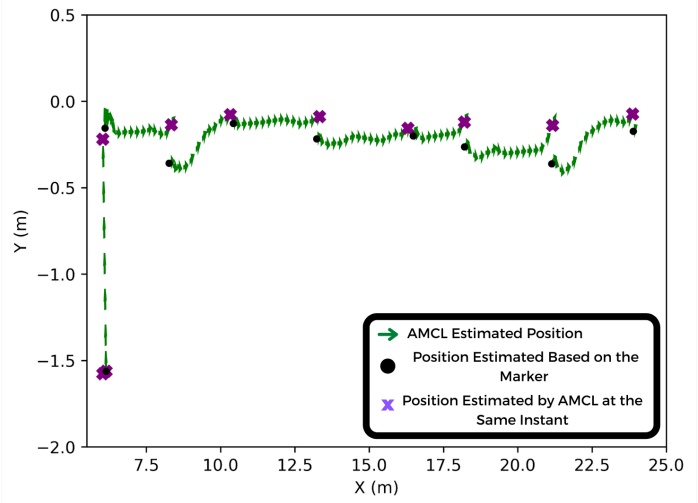

Figure 6.27: Position estimates for scenario 4 in its trajectory 3.

Table 6.4 shows the errors obtained from the estimation comparisons against the markers. Due to the way of implementing the system to store the positions based on the markers, and the discrepancies in the robot navigation only with the particle filter, once the convergence of the particles demands time and some movement of the robot to occur.

The error presented by the traditional approach is smaller than that observed, since it was not possible to obtain references for the calculations at times when the divergence was much larger. However, despite these factors, it is still evident the effectiveness of the

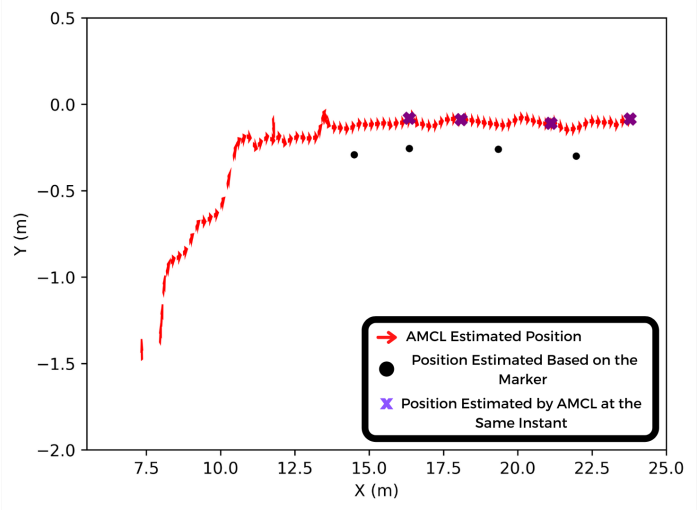

Figure 6.26: Position estimates for scenario 3 in its trajectory 3.

	Minimun Error		Maximun Error		Average Error (Ea)		Gain
	AMCL	AMCL Markers	AMCL	AMCL Markers	AMCL	AMCL Markers	
Trajectory 1	1.18m	0.01m	8.23m	0.24m	4.71m	0.08m	98.30%
Trajectory 2	0.09m	0.01m	1.62m	1.17m	0.66m	0.20m	69.70%
Trajectory 3	1.75m	0.01m	1.87m	0.23m	1.81m	0.14m	92.26%

Table 6.4: Position estimation errors relative to the markers in the global localization tests.

proposed system with the integration of fiducial markers, not only in comparison to the errors displayed, but also in observation of the navigation performed by the robot during the tests.

6.2.4 Consideration About the Localization System with the Real Robot

Based on the results obtained and the observations made during the tests with the real robot, it was possible to validate that the localization system based on the integration of the fiducial markers with the AMCL particle filter is effective to perform the localization of the mobile in the real environment, allowing to guarantee the autonomous navigation of the mobile robot, without the need to indicate or store the initial position of the robot,

since it can be obtained by the detection of the markers.

However, some improvements are still needed to the system. Some markers need to be repositioned in the environment, so that they do not introduce errors into the algorithm, and even create functions that prevent updates from being performed if the angle between the marker and the camera is in a range that does not allow an accurate calculation of distance. Also during the tests, there were difficulties in detecting some of the markers due to lighting variations in the environment, which requires that new calibrations with the camera need to be performed to prevent errors from occurring.

There were also difficulties in sending commands to the robot through the SSH access strategy due to the loss of connection with the Wi-Fi network in some points of the environment, which sometimes made it difficult to perform some tests, and required the search for alternatives to perform some procedures, such as mapping, where the commands for the robot teleoperation needed to be sent through a bluetooth keyboard connected to one of the raspberry.

The magni mobile platform proved to be very versatile during testing, and its integration with the ROS framework facilitated the adaptation of the system implemented in simulation to be used on it. In addition, the odometry data collection proved to be accurate, avoiding the introduction of significant errors in the system.

The localization system based solely on the SLAM approach with the AMCL algorithm, proved effective for localizing the robot for the position tracking problems, however, for the global localization problems the convergence of the algorithm particles for the tests performed were very time consuming, and did not obtain positions as accurate as the proposed approach. It is also necessary to consider that the tests were performed in moments when the corridors were empty, without the presence of people, which could interfere even more in the efficiency of the system, since they would introduce more noise to the readings performed by the LiDAR.

Chapter 7

Conclusions and Future Works

7.1 Developed Works

Having addressed problems related to the localization of mobile robots in indoor environments, during the course of this work, a localization system was developed through the integration of a SLAM approach based on the AMCL particle filter, and the position tracking of fiducial markers distributed throughout the indoor environment. A representation of the CeDRI hallways was built in a simulation environment in order to perform the preliminary tests of the proposed system's operation, as well as to evaluate the traditional system that uses only the particle filter.

The localization system consider the use of the ROS framework for establishing communication with the mobile robot and the sensors used for estimating the robot's position in the environment, namely a LiDAR sensor, odemetric sensor and an RGB camera. The system was also integrated through the developed codes, with other ROS algorithms and integration packages used for localization and navigation of the mobile robot, namely, gmapping, *ar_track_alvar* and navigation stack, which included the implemented ROS version of the AMCL algorithm.

The integration of the AMCL particle filter with position tracking based on fiducial markers showed great results in the tests performed in simulation, comprising a solution

to global localization problems and further improving the position estimation efficiency of the AMCL algorithm.

Making use of a mobile robotic platform with ROS integration, the developed localization system as well as the traditional approach with particle filtering was subjected to testing in the real environment. The proposed localization system proved to be more efficient than the other approach, especially in tests related to the global localization problem. The system with the AMCL algorithm also proved to be a viable solution if only the position tracking problem in a static environment is considered.

7.2 Future Works

Based on the results obtained in the development of the work, on the possibility of improvements and on the different tests that may still be specific, the following topics describe the work to be carried out in the future:

- Make improvements to the position update system based on fiducial markers integration of the EKF into the localization system;
- Optimization of the gmapping and navigation stack packages paramaters to improve the localization and navigation system and reduce computacional comsuption;
- Performing tests in the simulation and in the real environment with dynamic obstacles.

Bibliography

[1] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1 729 881 419 839 596, 2019. DOI: 10.1177/1729881419839596.

[2] M. Cardona, A. Palma, and J. Manzanares, "Covid-19 pandemic impact on mobile robotics market," in *2020 IEEE ANDESCON*, 2020, pp. 1–4. DOI: 10.1109/ANDESCON50619.2020.9272052.

[3] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39 830–39 846, 2020. DOI: 10.1109/ACCESS.2020.2975643.

[4] M. A. K. Niloy, A. Shama, R. K. Chakrabortty, M. J. Ryan, F. R. Badal, Z. Tasneem, M. H. Ahamed, S. I. Moyeen, S. K. Das, M. F. Ali, M. R. Islam, and D. K. Saha, "Critical design and control issues of indoor autonomous mobile robots: A review," *IEEE Access*, vol. 9, pp. 35 338–35 370, 2021. DOI: 10.1109/ACCESS.2021.3062557.

[5] A. R. Khairuddin, M. S. Talib, and H. Haron, "Review on simultaneous localization and mapping (slam)," in *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2015, pp. 85–90. DOI: 10.1109/ICCSCE.2015.7482163.

[6] L. Garrote, M. Torres, T. Barros, J. Perdiz, C. Premebida, and U. J. Nunes, "Mobile robot localization with reinforcement learning map update decision aided by an

absolute indoor positioning system," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1620–1626. DOI: 10.1109/IROS40897.2019.8967957.

[7] N. G. Hockstein, C. Gourin, R. Faust, and D. J. Terris, "A history of robots: From science fiction to surgical robotics," *Journal of robotic surgery*, vol. 1, no. 2, pp. 113–118, 2007. DOI: 10.1007/s11701-007-0021-2.

[8] E. A. Oyekanlu, A. C. Smith, W. P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, D. J. Kuhn, J. D. Mcghinnis, S. C. Buonavita, N. A. Looper, M. Ng, A. Ng'oma, W. Liu, P. G. Mcbride, M. G. Shultz, C. Cerasi, and D. Sun, "A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5g-based smart manufacturing applications," *IEEE Access*, vol. 8, pp. 202 312–202 353, 2020. DOI: 10.1109/ACCESS.2020.3035729.

[9] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, "Human–robot interaction in industrial collaborative robotics: A literature review of the decade 2008–2017," *Advanced Robotics*, vol. 33, no. 15-16, pp. 764–799, 2019. DOI: 10.1080/01691864.2019.1636714.

[10] E. Matheson, R. Minto, E. G. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, 2019, ISSN: 2218-6581. DOI: 10.3390/robotics8040100. [Online]. Available: https://www.mdpi.com/2218-6581/8/4/100.

[11] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021, ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2021.01.019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721000217.

[12] A. S. Aguiar, F. N. dos Santos, J. B. Cunha, H. Sobreira, and A. J. Sousa, "Localization and mapping for robots in agriculture and forestry: A survey," *Robotics*, vol. 9, no. 4, 2020, ISSN: 2218-6581. DOI: 10.3390/robotics9040097. [Online]. Available: https://www.mdpi.com/2218-6581/9/4/97.

[13] K. A. Farley, K. H. Williford, K. M. Stack, R. Bhartia, A. Chen, M. de la Torre, K. Hand, Y. Goreva, C. D. Herd, R. Hueso, *et al.*, "Mars 2020 mission overview," *Space Science Reviews*, vol. 216, no. 8, pp. 1–41, 2020. DOI: 10.1007/s11214-020-00762-y.

[14] A. Bolu and Ö. Korçak, "Adaptive task planning for multi-robot smart warehouse," *IEEE Access*, vol. 9, pp. 27 346–27 358, 2021. DOI: 10.1109/ACCESS.2021.3058190.

[15] E. Darack, "Mars environmental dynamics analyzer: Advancing the study of the martian atmosphere," *Weatherwise*, vol. 74, no. 4, pp. 14–17, 2021.

[16] C. Chen, E. Demir, Y. Huang, and R. Qiu, "The adoption of self-driving delivery robots in last mile logistics," *Transportation Research Part E: Logistics and Transportation Review*, vol. 146, p. 102 214, 2021, ISSN: 1366-5545. DOI: https://doi.org/10.1016/j.tre.2020.102214. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554520308565.

[17] Y. Shen, D. Guo, F. Long, L. A. Mateos, H. Ding, Z. Xiu, R. B. Hellman, A. King, S. Chen, C. Zhang, and H. Tan, "Robots under covid-19 pandemic: A comprehensive survey," *IEEE Access*, vol. 9, pp. 1590–1615, 2021. DOI: 10.1109/ACCESS.2020.3045792.

[18] V. W. S. Tung and R. Law, "The potential for tourism and hospitality experience research in human-robot interactions," *International Journal of Contemporary Hospitality Management*, 2017. DOI: https://doi.org/10.1108/IJCHM-09-2016-0520.

[19] G. Castellano, B. Carolis, N. Macchiarulo, and G. Vessio, "Pepper4museum: Towards a human-like museum guide," Sep. 2020.

[20] S. Onofre, B. Caseiro, J. P. Pimentão, and P. Sousa, "Using fuzzy logic to improve ble indoor positioning system," in *Technological Innovation for Cyber-Physical Systems*, L. M. Camarinha-Matos, A. J. Falcão, N. Vafaei, and S. Najdi, Eds., Cham: Springer International Publishing, 2016, pp. 169–177, ISBN: 978-3-319-31165-4.

[21] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*, Citeseer, vol. 20, 2007.

[22] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016. DOI: 10.1109/TRO.2016.2624754.

[23] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University - Computer and Information Sciences*, 2021, ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2021.02.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821000550.

[24] Z. Su, X. Zhou, T. Cheng, H. Zhang, B. Xu, and W. Chen, "Global localization of a mobile robot using lidar and visual features," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2017, pp. 2377–2383. DOI: 10.1109/ROBIO.2017.8324775.

[25] R. Negenborn, "Robot localization and kalman filters," *Utrecht Univ., Utrecht, Netherlands, Master's thesis INF/SCR-0309*, 2003.

[26] P. Goebel, *ROS By Example*, Lulu, Ed. Lulu, 2014. [Online]. Available: %5Curl%7Bhttp://www.lulu.com/shop/r-patrick-goebel/ros-by-example-volume-2-indigo/ebook/product-22251662.html%7D.

[27] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 4, pp. 1–26, 2021.

[28] H. Zhang, C. Zhang, W. Yang, and C.-Y. Chen, "Localization and navigation using qr code for mobile robot in indoor environment," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, pp. 2501–2506. DOI: 10.1109/ROBIO.2015.7419715.

[29] A. R. Cantieri, M. A. Wehrmeister, A. S. Oliveira, J. Lima, M. Ferraz, and G. Szekir, "Proposal of an augmented reality tag UAV positioning system for power line tower inspection," in *Iberian Robotics conference*, Springer, 2019, pp. 87–98.

[30] K. Boudjit and C. Larbes, "Detection and target tracking with a quadrotor using fuzzy logic," in *8th IEEE International Conference on Modelling, Identification and Control (ICMIC)*, 2016, pp. 127–132.

[31] L. Piardi, V. C. Kalempa, M. Limeira, A. S. de Oliveira, and P. Leitão, "Arena-augmented reality to enhanced experimentation in smart warehouses," *Sensors*, vol. 19, no. 19, p. 4308, 2019.

[32] M. Limeira, L. Piardi, V. C. Kalempa, P. Leitão, and A. S. De Oliveira, "Depthlidar: Active segmentation of environment depth map into mobile sensors," *IEEE Sensors Journal*, 2021.

[33] A. Ernst, "Practical navigation: Autonomous behavior on inexpensive robots," Ph.D. dissertation, 2017.

[34] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006. DOI: 10.1109/MRA.2006.1638022.

[35] Y. K. Tee and Y. C. Han, "Lidar-based 2d slam for mobile robot in an indoor environment: A review," in *2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, 2021, pp. 1–7. DOI: 10.1109/GECOST52368.2021.9538731.

[36] T. Chong, X. Tang, C. Leng, M. Yogeswaran, O. Ng, and Y. Chong, "Sensor technologies and simultaneous localization and mapping (slam)," *Procedia Computer Science*, vol. 76, pp. 174–179, 2015, 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IEEE IRIS2015), ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.12.336. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915038375.

[37] M. Zaffar, S. Ehsan, R. Stolkin, and K. M. Maier, "Sensors, slam and long-term autonomy: A review," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2018, pp. 285–290. DOI: 10.1109/AHS.2018.8541483.

[38] S. Royo and M. Ballesta-Garcia, "An overview of lidar imaging systems for autonomous vehicles," *Applied Sciences*, vol. 9, no. 19, 2019, ISSN: 2076-3417. DOI: 10.3390/app9194093. [Online]. Available: https://www.mdpi.com/2076-3417/9/19/4093.

[39] B. Huang, J. Zhao, and J. Liu, "A survey of simultaneous localization and mapping," *arXiv preprint arXiv:1909.05214*, 2019.

[40] K. Madhira, J. Patel, D. Kothari, D. Panchal, and D. Patel, "A quantitative study of mapping and localization algorithms on ros based differential robot," in *2017 Nirma University International Conference on Engineering (NUiCONE)*, 2017, pp. 1–5. DOI: 10.1109/NUICONE.2017.8325609.

[41] S. Slamtec, *Rplidar a1: Introduction and datasheet*, 2016. [Online]. Available: https://www.generationrobots.com/media/rplidar-a1m8-360-degree-laser-scanner-development-kit-datasheet-1.pdf.

[42] D. T. Son, M. T. Anh, D. D. Tu, L. Van Chuong, T. H. Cuong, and H. S. Phuong, "The practice of mapping-based navigation system for indoor robot with rplidar and raspberry pi," in *2021 International Conference on System Science and Engineering (ICSSE)*, 2021, pp. 279–282. DOI: 10.1109/ICSSE52999.2021.9538474.

[43] M. T.R. and A. M., "Obstacle detection and obstacle avoidance algorithm based on 2-d rplidar," in *2019 International Conference on Computer Communication and Informatics (ICCCI)*, 2019, pp. 1–4. DOI: 10.1109/ICCCI.2019.8821803.

[44] R. Singh and K. S. Nagla, "Comparative analysis of range sensors for the robust autonomous navigation–a review," *Sensor Review*, 2019.

[45] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015, pp. 74–77. DOI: 10.1109/ICINIS.2015.35.

[46] Y. Kim and H. Bang, "Introduction to kalman filter and its applications," *Introduction and Implementations of the Kalman Filter, F. Govaers, Ed. IntechOpen*, 2019.

[47] C. Urrea and R. Agramonte, "Kalman filter: Historical overview and review of its use in robotics 60 years after its creation," *Journal of Sensors*, vol. 2021, 2021. DOI: 10.1155/2021/9674015.

[48] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, 1999, 1322–1328 vol.2. DOI: 10.1109/ROBOT.1999.772544.

[49] J. Azevedo, "Automatic parameter tuning of algorithms using optimization," Ph.D. dissertation, Master's thesis, Instituto Superior Técnico Lisboa-Lisbon, Portugal, 2017.

[50] C. Camargo, J. Gonçalves, M. Á. Conde, F. J. Rodríguez-Sedano, P. Costa, and F. J. García-Peñalvo, "Systematic literature review of realistic simulators applied in educational robotics context," *Sensors*, vol. 21, no. 12, 2021, ISSN: 1424-8220. DOI: 10.3390/s21124031. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/4031.

[51] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326. DOI: 10.1109/IROS.2013.6696520.

[52] C. Robotics, *Robot simulator coppeliasim: Create, compose, simulate, any robot.* [Online]. Available: https://www.coppeliarobotics.com/, Accessed: Oct. 2021.

[53] V. Leite, Â. P. Ferreira, and J. Batista, "Improving the storage capability of a microgrid with a vehicle-to-grid interface," in *II Congreso Iberoamericano Sobre Microrredes con Generación Distribuida de Renovables*, 2014.

[54] C. Fairchild and T. L. Harman, *ROS robotics by example*. Packt Publishing Ltd, 2016.

[55] P. Goebel, *ROS By Example*, Lulu, Ed. Lulu, 2013. [Online]. Available: %5Curl%7Bhttp://www.lulu.com/shop/r-patrick-goebel/ros-by-example-indigo-volume-1/ebook/product-22015937.html%7D.

[56] B. K. LUKNANTO, "A review of 2d slam algorithms on ros," 2020.

[57] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd, 2018.

[58] A. Mahtani, L. Sanchez, E. Fernández, and A. Martinez, *Effective robotics programming with ROS*. Packt Publishing Ltd, 2016.

[59] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System*. " O'Reilly Media, Inc.", 2015.

[60] R. Wiki, *Move_base.* [Online]. Available: http://wiki.ros.org/move_base, Accessed: Oct. 2021.

[61] S. Niekum and Saito, "Ar track alvar ros package," 2013. [Online]. Available: http://wiki.ros.org/ar_track_alvar.

[62] P. Jin, P. Matikainen, and S. S. Srinivasa, "Sensor fusion for fiducial tags: Highly robust pose estimation from single frame rgbd," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5770–5776. DOI: 10.1109/IROS.2017.8206468.

[63] R. Wiki, *Running ros across multiple machines.* [Online]. Available: http://wiki.ros.org/ROS/Tutorials/MultipleMachines, Accessed: Oct. 2021.

[64] U. Robotics, *Magni silver mobile based robot.* [Online]. Available: https://www.ubiquityrobotics.com, Accessed: Oct. 2021.

[65] R. Agrawal, *Ubiquity robotics repository.* [Online]. Available: https://github.com/ubiquityrobotics, Accessed: Oct. 2021.

[66] R. Pi, *Raspberry pi 4 model b: Datasheet*, 2019. [Online]. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf.

[67] Hokuyo, *Ust-10lx specification*, 2015. [Online]. Available: https://www.hokuyo-aut.jp/dl/UST-10LX_Specification.pdf.

[68] T. Baltovski, *Urg_node.* [Online]. Available: http://wiki.ros.org/urg_node, Accessed: Oct. 2021.

[69] RealSense, *Intel realsense d400 series product family: Datasheet*, 2019. [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf.

[70] Doronhi, *Ros wrapper for intel realsense devices.* [Online]. Available: https://github.com/IntelRealSense/realsense-ros, Accessed: Oct. 2021.

Appendix A

Publications

A. Oliveira, L. Piardi, E. Bertogna, and P. Leitão. "Improving the Mobile Robots Indoor Localization System by Combining SLAM with Fiducial Markers." In 18th IEEE Latin American Robotics Symposium, 2021.

A. Oliveira, L. Piardi, and P. Leitão. "Implementation of a navigation system for a mobile robot in a dynamic environment using AR tags to increase localization accuracy." In 1st Symposium of Applied Science for Young Researches, 2021.