Application of a Mobile Robot to Spatial Mapping of Radioactive Substances in Indoor Environment

Luis Fernando Piardi

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Engenharia Industrial.

Work oriented by:

Professor PhD José Luis Sousa Magalhães Lima

Professor PhD Paulo Costa

Professor PhD Marcos Bombacini

Bragança

2018

Application of a Mobile Robot to Spatial Mapping of Radioactive Substances in Indoor Environment

Luis Fernando Piardi

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Engenharia Industrial.

Work oriented by:

Professor PhD José Luis Sousa Magalhães Lima

Professor PhD Paulo Costa

Professor PhD Marcos Bombacini

Bragança

2018

Acknowledgement

I would like to express my thanks to Professor PhD José Luis Lima, supervisor of this work, for the dedication, guidance, incentive and opportunity to develop this work, and articles, highlighting the challenge and support to develop the writing of the document in English.

To Professor PhD Paulo Costa, co-supervisor of this work, my gratitude for the guidance and willingness to share his vast knowledge in the area.

To Professor PhD Ana Isabel Pereira, for support, availability and supervision in work activities using evolutionary algorithms.

To Professor PhD Marcos Bombacini for having accepted to be the distance co-supervisor in this agreement of the Dual Diplomacy program.

To my friends and college colleagues, Dual Diplomacy, laboratory and calculus center which I shared eternal moments of learning.

I thank the IPB, UTFPR and INESC-TEC for the conditions provided.

Finally, a special thanks to my parents and family for having provided all the accompaniment and support necessary to carry out this work.

Abstract

Nuclear medicine requires the use of radioactive substances that can contaminate critical areas (dangerous or hazardous) where the presence of a human must be reduced or avoided. The present work uses a mobile robot in real environment and 3D simulation to develop a method to realize spatial mapping of radioactive substances. The robot should visit all the waypoints arranged in a grid of connectivity that represents the environment. The work presents the methodology to perform the path planning, control and estimation of the robot location. For path planning two methods are approached, one a heuristic method based on observation of problem and another one was carried out an adaptation in the operations of the genetic algorithm. The control of the actuators was based on two methodologies, being the first to follow points and the second to follow trajectories. To locate the real mobile robot, the extended Kalman filter was used to fuse an ultra-wide band sensor with odometry, thus estimating the position and orientation of the mobile agent. The validation of the obtained results occurred using a low cost system with a laser range finder.

Keywords: Mobile Robot; Spatial Mapping Radioactive Substances; Path Planning; Genetic Algorithm; Kalman Filter; Ultra-wide Band; Laser Range Finder.

Resumo

A medicina nuclear requer o uso de substâncias radioativas que pode vir a contaminar áreas críticas, onde a presença de um ser humano deve ser reduzida ou evitada. O presente trabalho utiliza um robô móvel em ambiente real e em simulação 3D para desenvolver um método para o mapeamento espacial de substâncias radioativas. O robô deve visitar todos os waypoinst dispostos em uma grelha de conectividade que representa o ambiente. O trabalho apresenta a metodologia para realizar o planejamento de rota, controle e estimação da localização do robô. Para o planejamento de rota são abordados dois métodos, um baseado na heurística ao observar o problema e ou outro foi realizado uma adaptação nas operações do algoritmo genético. O controle dos atuadores foi baseado em duas metodologias, sendo a primeira para seguir de pontos e a segunda seguir trajetórias. Para localizar o robô móvel real foi utilizado o filtro de Kalman extendido para a fusão entre um sensor ultra-wide band e odometria, estimando assim a posição e orientação do agente móvel. A validação dos resultados obtidos ocorreu utilizando um sistema de baixo custo com um *laser range finder*.

Palavras-chave: Robô móvel; Mapeamento Espacial de Substâncias Radioativas; Planejamento de Caminho; Algoritmo Genético; Controle; Filtro de Kalman; Ultra-wide Band, Laser Range Finder.

Contents

List of Tables

List of Figures

Acronyms

EKF Extended Kalman Filter.

GA Genetic Algorithm.

IDE Integrated Development Environment.

LRF Laser Range Finder.

MCU Micro-controller Unit.

NUMDAB Nuclear Medicine Database.

ODE Open Dynamics Engine.

PET Positron Emission Tomography.

PRM Probabilistic Roadmap.

RF Radio Frequency.

RFID Radio Frequency Identification.

RRT Rapidly Random Tree.

RSS Received Signal Strength.

SIAR Sewer Inspection Autonomous Robot.

TCP Transmission Control Protocol.

ToF Time of Flight.

TSP Travelling Salesman Problem.

UDP User Datagram Protocol.

UWB Ultra-wideband.

WMR Wheeled Mobile Robots.

Chapter 1

Introduction

This work deals with the study and development of three essential areas of wheeled mobile robot, being: path planning, location estimation and control. The intention is to develop methods to enable a mobile robot to perform spatial radiation map of an indoor environment contaminated by harmful radioactive substances to humans. As a basis for the studies, a simulation platform will be used to test the algorithms. After the simulation step, a real mobile robot will be used to validate the methodology for the environment scan.

1.1 Motivation and Framework

Nuclear medicine requires the use of radioactive substances to observe the physiological condition of human tissues in a minimally invasive manner. To diagnose diseases such as cancer and its metabolism, methods such as Positron Emission Tomography (PET) are applied. Basically the patient ingests a radioactive substance analogous to glucose, where cells that have an accelerated reproduction tend to consume it [1]. However the substance is not fully absorbed, resulting in a concentration of radiation in the region of rapid cell reproduction (usually regions where malignant cancer cells are developing). The patient then undergoes a procedure that performs a scan on his body to locate regions where the radioactive emissions are concentrated as shown in Figure 1.1a. Figure 1.1b shows the

image resulting from the diagnoses where regions of the body with abnormal cell behavior are highlighted.

(a) PET scanner. (b) Diagnostic.

Figure 1.1: PET procedure [1].

The administration of nuclear radio pharmaceutical components to the patient must be carefully done by specialists. Unfortunately, the patient can contaminate the environment with physiologic needs. Moreover, environment and the patient should be isolated by a period of time regarding the decay of nuclear properties. The inspection of the clearance of the environment is mainly made by human beings that are exposure to the ionizing radiation that may cause the damage in the organs and tissues. The scanning and measurement of the radiation can be done resorting to a mobile robot that performs the acquisition based on a Geiger counter. The spatial radiation map performed by the mobile robot should guarantee that the complete scan is performed and ensure the environment is clean and technicians can enter the room.

According to Nuclear Medicine Database (NUMDAB), there are 1490 nuclear medicine institutions in the world, of which 1288 are active. Actually, 690 thousand PET and PET-CT (Computed Tomography) annual examinations are registered in the world [2].

The presented work addresses a path planning, localization and control method that scans the desired environment. The approach that will be developed has the potential to be adapted and applied to other solutions indoors using mobile robots.

1.2 Objectives

The main objective of this work is to develop and test a solution with low cost mobile robots to perform the spatial mapping of radioactive substances in indoor environments without human intervention. Consequently the other objectives also make up this work:

- Develop an interface between man and robot for remote access in real time.
- Development of methods for planning routes in a connectivity grid, visiting all available points.
- Implement a smooth curve for the robot path reference.
- Development of controllers for the robot to perform the trajectory.
- Implement the methodology developed in SimTwo simulation environment.
- Implement the methodology developed in real environment using a mobile robot provided, a location system based on UWB, and a ground truth of low cost.

1.3 Document Structure

This document is divided into 6 chapters, to describe the work carried out throughout the dissertation.

The introduction is described in Chapter 1, which presents the proposal of the work as well as the contamination of indoor environments by patients undergoing PET exams, and a mobile robot is proposed to perform the spatial mapping of the radioactive substance from nuclear medicine activities.

Chapter 2 presents real cases demonstrating the usefulness of mobile robots in various areas. Then a theoretical review of aspects such as locomotion, navigation and location, path planning as the foundation of mobile robotics is carried out. The chapter is then finalized presenting concepts of travelling salesman problem and genetic algorithm.

Chapter 3 presents the system architecture used in this work, demonstrating its structure used to address the problem and presenting and describing the software, hardware and simulation environment.

Chapter 4 presents the Extended Kalman Filter to address the errors and uncertainties in the robot location, followed by the description of the heuristic and genetic algorithm for path planning. It ends with a description of two methods for controlling the robot: points following and segment following.

Chapter 5 contains the results obtained during the tests to validate the methodology and is finished with the practical results.

Finally, chapter 6 presents the expanded conclusions of ideas and suggestions for future work.

Chapter 2

Related Work

In this section it will be analyzed the related work in the area of wheeled mobile robots such as, real applications, geometries, localization and path planning. Bearing in mind the application developed during this work, and taking into account the current context where each day becomes more common the coexistence with the different types of robots, this chapter tries to elaborate, based on a scientific research the state of the art.

2.1 Wheeled Mobile Robot Applied in Monitoring and Surveillance

The first robots have emerged to operate in a cell isolated environment, most often within warehouses or industries. There is no universally recognized definition of robotics, however Russel and Norving present a possible definition as:"Robots are physical agents that perform tasks by manipulating the physical world. To do so, effector they are equipped with effectors such as legs, wheels, joints, and grippers. Robots are also equipped with sensors, which allow them to perceive their environment" [3].

In general, robots have been developed to perform tasks that the human being for reasons of unhealthiness, incapacity or disinterest does not execute. The developed robotic systems are usually implemented when, for the same work, the robot performs with a

higher quality and speed or at a lower cost compared to human work [4]. Citing Tzafestas: "The robots contribute in one or the other way to human, industrial, agricultural, technical, and social life improvements"[5].

The class of robot used in this work is of the mobile category with wheels. A Wheeled Mobile Robots (WMR) is applied to problems related to operations in complex environments such as hazardous and dangerous environments, unknown environments with dynamic obstacles, or planetary explorations and others. It will be presented the concept and advantages of applying WMR for inspection, monitoring and surveillance, and then will be exposed some examples and practical applications well respected.

Inspection and monitoring of controlled or hazardous environments is an important issue for the sustainability, maintenance and use of these sites. These environments can be characterized due to the presence of dust, humidity, large temperature variations, fire risks, biological agents or even toxic and radioactive substances, which is the case addressed in the present work.

The classical approach to environment inspection and monitoring activities utilizes human workers, i.e., this activity may present safety and health hazards to human inspectors. For this reason, there is a need to avoid and replace the presence of humans in hazardous areas by robots. Solutions with WMR have been an ideal alternative for inspection and monitoring because they have a relatively low cost, avoid worker exposure, are more accurate and perform in less time when compared to the classical approach [6].

To operate in this type of environment the WMR must be able to acquire and use knowledge about the environment, estimate a position, control wheel speed, plan the route, have the ability to recognize obstacles and respond in real time to the different situations that may occur. All these features should work together [7]. In the remainder of this section we will comment and cite some papers demonstrating applications of mobile robots for monitoring and inspection that contemplate this set of functions in a robotic system.

2.1.1 Quince Robot

Quince WMR was employed to conduct inspections and monitoring of the nuclear power plant accident at Fukushima Daiichi in Japan. Due to an earthquake with a magnitude of 9 on the Richter scale and a tsunami that hit East Japanese, an accident occurred with the reactor nuclear, causing explosions of hydrogen where radioactive materials contaminated the environment. The area was very dangerous for humans to inspect the damage due to exposure to radioactive materials. Therefore, the Quince were adapted with components able to withstand the radiation present in the environment to carry out the inspection and monitoring of the environment [8], [9]. The robots applied in the inspection can be seen in Figure 2.1.

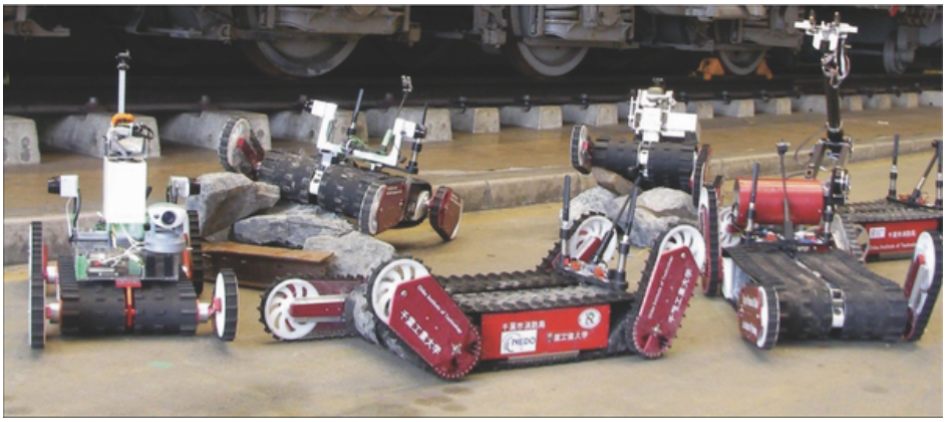


Figure 2.1: Robots used for inspection at Fukushima plant after disaster [8].

2.1.2 SIAR Robot

Sewer Inspection Autonomous Robot (SIAR) is a Portuguese project of IDMind in the development phase of autonomous WMR to inspect sewer systems with the least human intervention, presented in Figure 2.2. The prototype has already carried out tests in the sewage of Barcelona demonstrating its capacity of locomotion, navigation and perception. The robot is composed of six independent wheels to overcome obstacles efficiently. It has seven sensors of RGB-D cameras for its perception, with which it analyzes the data for autonomous navigation [10].

Figure 2.2: SIAR robot conducting an inspection in a sewer[10].

2.1.3 SACI Robot

SACI is a teleoperated WMR designed through wireless control and video cameras to act in fire fighting. It is also suitable for underground areas, for local assessment, measurement of toxic and flammable gases, as well as identification of the existence of victims for rescue. The Figure2.3 shows the robot in a demonstration of its operation. It is equipped with water cannons capable of emitting a mist or jet of water to combat the fire [11].

Figure 2.3: SACI robot emitting a jet of water to combat the fire [11].

2.1.4 Soryu Robot

The WMR Soryu developed by IRS and professor Hirose Shigeo of the Tokyo Institute of Technology. It is a rescue robot to help relief efforts in the aftermath of earthquakes

and other disasters by navigating through wreckage (Figure 2.4) that is too dangerous for people to enter and by gathering information on missing persons and the surrounding conditions. The WMR is equipped with a camera incorporating a CCD (Charge-Coupled Device) that converts light into electric signals, a thermo-graphic camera, the robot can locate disaster victims even when they are covered in debris, thanks to its ability to detect body heat [12].

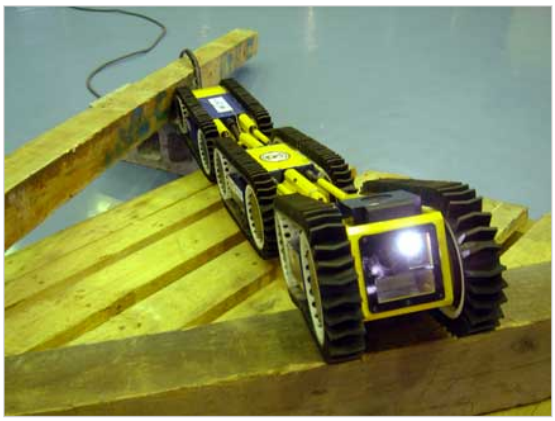


Figure 2.4: Soryu robot navigating through wreckage in test environment [12].

2.1.5 Roomba Robot

The Roomba was developed by iRobot. Although it is not a mobile robot applied in monitoring and surveillance it has an elaborate system of functioning and very widespread nowadays. This is a robotic floor vacuum, showed at Figure 2.5, capable of moving around home and sweeping up dirt while moving. It performs three types of cleaning via two rotating brushes that sweep the floor, a vacuum sucking dust and particles off the floor, and side sweeping brushes to clean baseboards and walls [5].

2.1.6 Curiosity Robot

One of the most famous WMR today is the Curiosity shown in Figure 2.6. It landed on Mars in August 2012 to accomplish his mission. The Curiosity is an unmanned vehicle with the goal of space exploration to collect images and samples rock, soil and air of Mars

Figure 2.5: The Roomba discovery vacuum [5].

for on board analysis to verify the existence of life on the planet. Curiosity has tools including 17 cameras, a laser to vaporize and study small pinpoint spots of rocks at a distance, and a drill to collect powdered rock samples. The Curiosity looks for special rocks and signs of water or organic life. [13].

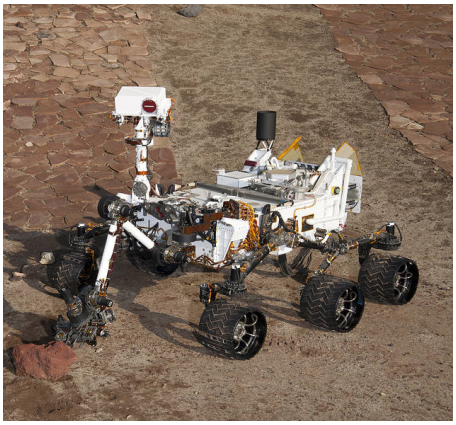


Figure 2.6: Curiosity - Robot for exploration of the planet Mars (NASA). (Adapted from:[14]).

To finalize this section where some WMR to inspection and monitoring was observed, it should be clear the complexity of these systems. Mobile robotics is a result of the integration of several areas of knowledge in order to achieve significant results. To solve problems of locomotion requires the understanding of kinematic, dynamics and control theory. For the understanding of robust systems of perception, it requires the analysis and processing of signals and artificial vision. Localization and navigation require knowledge

of computational algorithms, information theory, artificial intelligence, and probability theory [4]. Research in the area of mobile robotics is a challenge to be faced with enthusiasm to obtain a know how in the areas of scientific knowledge.

2.2 Locomotion of Wheeled Mobile Robots

The WMR have particular characteristics that make them suitable for the successful accomplishment of certain tasks. In other words, it is the task itself that determines, in a first instance, the structure of the mobile robot. In this work, it will not be mentioning legged, aerial, or submarine mobile robots, which are out of scope of this work. Therefore, from this section onwards, when mentioned robots or mobile robots will directly indicate WMR.

In this section the main types of locomotion developed for mobile robots with wheels will be presented. It is worth mentioning that for each specific application a different type of locomotion is used, which fits the desired objectives of the robotic application.

2.2.1 Ackerman Steering Geometry

This geometry in question is characterized by the fact that all wheels have their axes arranged as a radius of a circle, having a central point in common. As the rear wheels are fixed, this center point has to be defined from an extended line running through the rear axle. In order to intercept the axes of the front wheels, this line requires that the inner front wheel, on a change of direction, has a turning angle superior to the one of the outer front wheel. This system is shown Figure 2.7.

The greatest advantage of this system is the possibility of the lateral forces being passively neutralized through the slip restrictions, reducing the effort made by the motor. In addition, it has a small complexity of mechanical structure and control [4]. This system is the preferred mechanism for vehicles and it implies a minimum radius of curvature, making it difficult to maneuver [15]. Due to such difficulties presented in this system our work does not adopt this type of geometry for the robot.

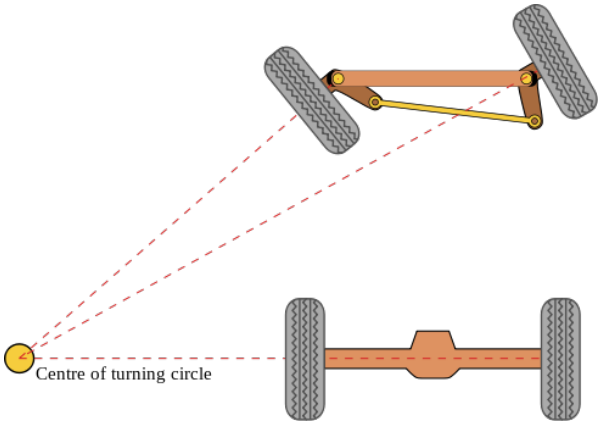


Figure 2.7: Geometry of robotic locomotion Ackerman steering.

2.2.2 Omnidirectional Geometry

Robots with omnidirectional traction have wide acceptance in the academic environment, as well as diverse applications in robotic competitions and industrial implementations. This configuration can have three or four wheels [16], each actuated by one motor, and has as its main characteristic the ability of the robot to move in all directions. The Figure 2.8 shows a prototype of a three wheels omnidirectional robot.

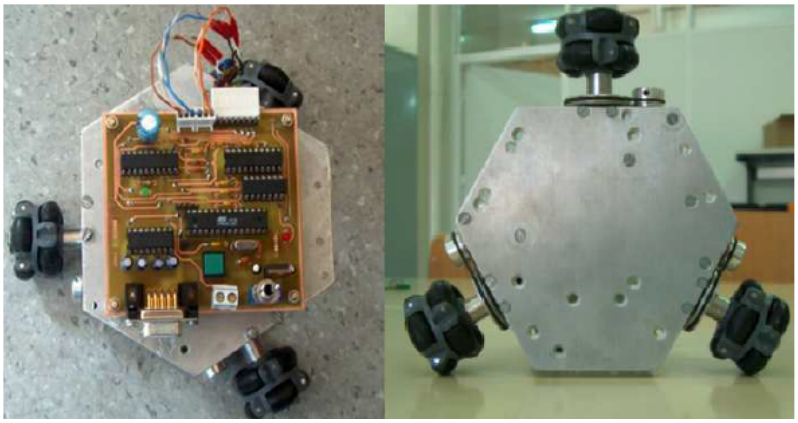


Figure 2.8: Geometry of robotic locomotion omnidirectional [17].

For its movement does not have nonholonomic restrictions, it is necessary that the robot uses special wheels as shown in Figure 2.9, which are different from conventional wheels because in its axis of rotation, this wheel grants the robot the ability to move laterally, resulting in movements in all directions. Several academic papers for the purpose

of research and robotic competitions with omnidirectional robots can be found in the literature, for example [18], [19].

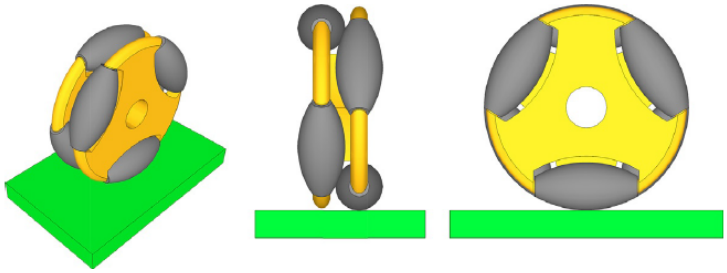


Figure 2.9: Omni wheel with six free rotating rollers mounted around the wheel rim [20].

However, for the application of this project, this robotic geometry presents wear on the wheels due to friction [21] and consequently the need for maintenance, since it is a necessity to reduce expenses and optimize the working time of the robot.

2.2.3 Differential Geometry

Robots with differential drive feature a simple drive mechanism. It is often used for applications in small robots. Usually robots with this geometry feature one or more castors wheels to support the vehicle and prevent tilting of the structure. The two main wheels are arranged on a common shaft controlled by independent motors [20].

This system (Figure 2.10) can change the direction of its trajectory by varying the speed of rotation of each of its wheels, however this configuration has nonholonomic restrictions of movement, which means that it does not allow displacement towards the axes of rotation.

This geometry was selected for this work due to mechanical and control simplicity, well its high robustness, compared to omnidirectional robots, the maintenance rates are smaller, making it an attractive for industrial applications. When using a common wheel, the calculation for odometry is more accurate when compared to wheels for omnidirectional robots.

Figure 2.10: Robot with differential drive (Pioneer P3-DX).

2.3 Localization and Navigation

For a WMR, it is extremely important to be able to locate, that is, to determine its position in the environment where it is situated, by interpreting the data obtained from sensors. In situations where the robot needs to reach a specific localization, it is necessary for the robot to possess or acquire through sensors, a model of the environment in which it is located. In this way it is able to plan a method to achieve its goal without taking great risks of getting lost along the trajectory [4]. The robot must be equipped with some mechanism to locate in relation to the environment that it is inserted. Since the beginning of research in the field of mobile robotics, one of the great challenges faced by researchers and developers of mobile robotic technology is the localization with the highest precision possible. Consequently, several localization techniques have been developed over the past two decades [22].

The design and development stage of a localization technique is extremely important for a robot to complete successfully its mission.

2.3.1 System Based on Wire Guidance

Wire guidance systems (or also known as inductive guidance) are performed by following a buried cable in the floor [23] as shown in Figure 2.11. The principle of operation is extremely simple, a sensor embedded in the WMR, has two coils arranged on the same axis. These coils detect the electromagnetic field generated by the sinusoidal current that runs through the wire that is on the ground. The difference in electric voltage between

the two coils will create the steering signal to the steering motor of the mobile robot [24]. The voltage difference will control the rotation of the motor and consequently the robot movement.

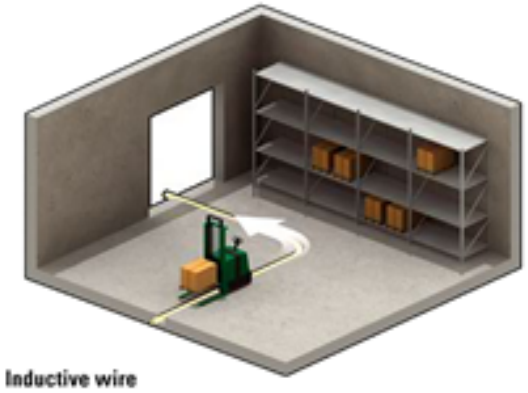


Figure 2.11: Localization and navigation system based on wire guidance [25].

It is a common method in industries due to its simplicity and robustness being these the positives of this method of localization. However, it is a system that provides a fixed path for the WMR, because as the wire guiding the robot is buried in the floor, a change in path means a high cost, so it is not used in industries that need to reconfigure their layout frequently [23]. For the correct implementation of this system, the speed of the robot must be low and limited, so that the embedded sensors never fail to capture the effect of the magnetic field.

2.3.2 Systems Based on Strip Guidance.

Another application widely used to robotics localization are systems based on strip. The robot trajectory is based on magnetic or colored strips arranged on the floor, line detection is performed by hall effect sensors, which detect the magnetic field in the case of using metallic tape, or optical sensor, when using colored bands. This method has lower implementation costs than the wire guidance system, since it is not necessary to cut the floor for the installation of this system, besides the reconfiguration of the system without much simpler, where only replacing the tape with the new desired layout [26]. Figure 2.12

presents a model of this system.

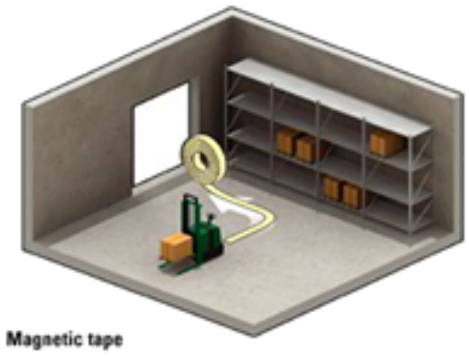


Figure 2.12: Localization and navigation system based on strip guidance [25].

However, n widely used environments,, this system reveals disadvantages in the durability of the implemented brands, since the tapes are damaged more easily and the painted lines are exposed to more dirt, potentially inhibiting their follow-up by the robot [23], [24], [26].

2.3.3 Systems Based on Marker

Marker based systems is another method for locating WMR in different environments. It presents markers embedded in the ground as shown in Figure 2.13. Markers can be magnetic labels, reflectors, passive RF, geometric shapes or even bar code [22].

Figure 2.13: Localization and navigation system based on marker [25].

Several markers should be spread at strategic localization throughout the room, where each will have its defined position. In this way, when the robot approaches a marker its reference position will be updated. In order to prevent the robot from getting lost by not locating a marker, a parallel system of odometry for localization is used.

Markers is a flexible solution for locating an AGV inside a factory, because there is the possibility of quickly changing the position of the markers, without wasting time or need to stop the production process [27].

Sobreira et al.[28] performs the development of a new localization system based on security lasers presents in most AGV robots for calculating the distance between landmark and robot. "An enhanced artificial beacons detection algorithm is applied with a combination of a Kalman filter and an outliers rejection method in order to increase the robustness and precision of the system. This new robust approach allows to implement such system in current AGVs" [28]. They highlight the peculiarity of the work because in many robots today there is the safety laser sensor, due to the need to prevent collisions against obstacles and especially against humans.

2.3.4 Systems Based on Trilateration and Triangulation

This method consists in detecting the localization of the robot through beacons usually arranged in high parts of the walls as presented in the Figure 2.14. To complete this localization system it is necessary to have a laser that performs a continuous rotating scanning to locate the reflectors, arranged on the robot.

To ensure the effectiveness of the method, it is necessary that the laser can detect and identify at least three reflectors without any obstacles between them. The laser obtains the distance of the robot to each of the reflectors, and thus analyzing the intersection between the ray formed by the laser and the sensors, it is possible to determine the position of the robot in the environment.

The main advantages of this method is that the robot is always analyzing its position within the room without the need to estimate its position with odometry, as in the method

Figure 2.14: Localization and navigation system based on trilateration and triangulation [25].

based on markers, besides having great robustness and flexibility. In contrast, the sensors used have a considerable cost, in addition to the need to prepare the space [27]. Below are some works that present an approach derived from triangulation to locate.

Márton et. al. also uses the trilateration method to define the position of the robot, however, instead of using laser it uses ultrasound sensors for the localization and a polynomial regression to estimate the localization of robot [29].

Ronzoni et. al. presents the global localization based on the distance of the reflectors. The purpose of this work is the AGV's self-localization based on the identification of landmarks taking into account false detections, very common in industrial environments. The position of the robot is calculated with a scanning using the laser located at the top of the robot, without any sensory fusion [30].

Gonçalves et. al. uses two laser sensors arranged in parallel on an omnidirectional robot inserted in a known environment. With the distance data obtained from the sensors it performs the triangulation that is fused with the odometry to estimate the robot's pose [31].

2.3.5 Perfect Match

Another method for locating mobile robots is the perfect match. This proposal has been used in competitions such as robot soccer [32] and industrial applications [33].

This approach promises a robust localization, accurate and computationally efficient able to be processed in real time. This method requires a priori knowledge of the navigation area. To improve the accuracy of localization, commonly is used mathematical tools to sensor fusion, such as Kalman Filter.

Consecutive observations are performed aboard robot sensor (e.g. camera). So is computed the robot position, processing the images to find the references that exist in the environment. The results optimization is a consequence of the errors minimization obtained by numerical methods described in [32]. To synthesize and conclude the central idea of this topic about robotic localization, it can be defined as an approach for the robot to localize itself in the environment it is in. For this, a method is used from the knowledge of the map of the environment, to interpret the data obtained from the sensors contained in the robot in order to determine the spatial configuration of the robot.

2.4 Path Planning

Much research in the mobile robotics community is focusing its efforts on developing navigation improvements and path planning for robots. A good path planning approach will define the efficiency and feasibility of the real application of a project using WMR to perform the most varied tasks that can be attributed to them.

The path planning is defined by Roland Siegwart, as: “Given a map and a goal localization, path planning involves identifying a trajectory that will cause the robot to reach the goal localization when executed. Path planning is a strategic problem-solving competence, as the robot must decide what to do over the long term to achieve its goals.” [4]. Basically, path planning will select a path among several possible to run in a physical space, starting from the initial position to the objective position, avoiding any collisions with obstacles in the environment. It is a representation of the possible paths to be

executed from geometric representations with the use of mathematical tools.

Below, will be briefly presented some of the main strategies used to represent the robot environment, and then perform path planning.

2.4.1 Roadmap

The main idea of the Roadmap (map that contains roads) is to create nodes (vertex) and links between them (edges). Nodes represent certain positions. The links represent the paths between nodes. This approach also classifies the environment as free spaces or spaces with obstacle. The task of this path planning is to connect the start point and the goal point with an existing road connection in the map to find a connecting sequence of roads [20]. There are two principals approaches for this strategy: visibility graph and Voronoi graph.

Visibility Graph

The visibility graph method is one of the earliest path planning methods, therefore it is popular in mobile robotic. A visibility graph consists of all possible connections among any two vertices that lie entirely in the free space of the environment. This means that for each vertex connections are made to all the other vertices that can be seen from it. The start point and the goal point are treated as vertices. Connections are also made between neighboring vertices of the same polygon. Figure 2.15 illustrate this method.

Although the visibility graph is simple to implement, the number of connections increases with the number of obstacles present on the map, which increases complexity and makes it computationally inefficient. Another existing problem is the limits of the obstacles, which can result in collisions with the mobile robot. An example application can be found in [34], where cooperative approach is used for robots and visibility graph approach is used to generate a series of intermediate objectives which guarantee the robots to reach the final objective without local minima.

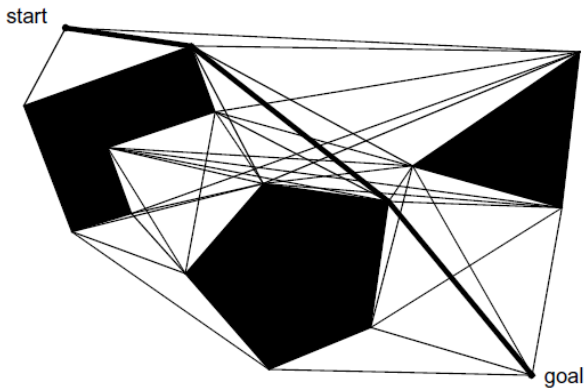


Figure 2.15: Visibility Graph: the highlighted line indicates the selected path for this example. [4]

Voronoi Diagram

Voronoi diagram is a complete road map that tends to maximize the distance between the robot and obstacles in the map [4]. Figure 2.16 show an example of the Voronoi diagram. It is formed by a set of points equidistant from obstacles. The resulting path of this method, which connects the start point to the end point, is the shortest path within this diagram. However, this is not the minimum path.

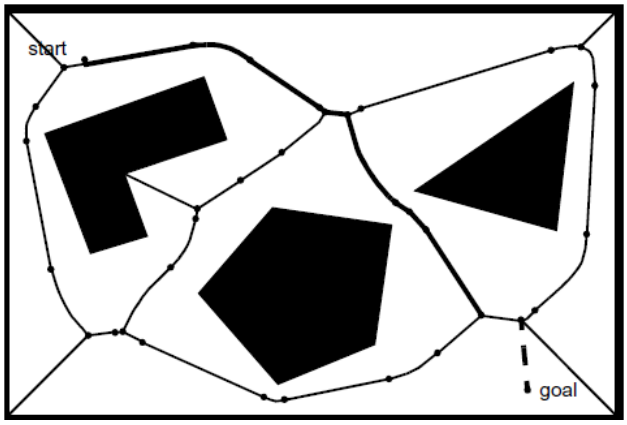


Figure 2.16: VoronoiGraph [4].

The advantage of this methodology is that it yields free paths which tend to maximize the clearance between the robot and the obstacles [35]. Driving on such a road minimizes the risk of colliding with obstacles which can be desired when the robot pose is known

with some uncertainty.

In [36] the Voronoi diagram is used together with the Dijkstra algorithm to obtain a shorter path in an obstacle environment, where it presents satisfactory results in the simulation environment. Already in [37] the Voronoi diagram is used for robotic football competitions, where there is great competitiveness and dynamics of the environment to avoid obstacles and collisions with other robots and perform the movement in the field.

2.4.2 Cell Decomposition

The main idea of cell decomposition is to perform a division of the environment into finite geometric areas or cells (see Figure 2.17). So that these regions are connected in an adjacent distribution.

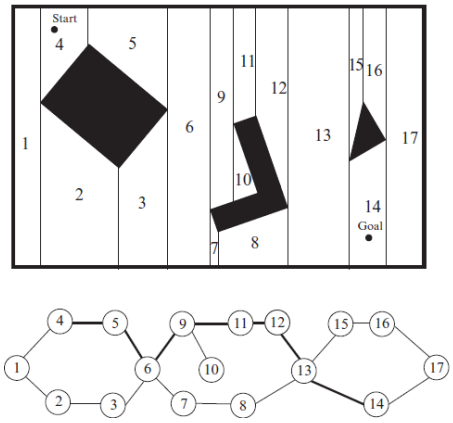


Figure 2.17: Cell decomposition apply vertical geometry (top) and adjacency graph (bottom) [35].

There must be a discrimination of cells between free areas and areas occupied by obstacles [35]. Assuming the decomposition is computed, path planning with a cell decomposition is usually done in two steps: first, the planner determines the cells that contain the start and goal, respectively, and then the planner searches for a path within the adjacency graph [38]. The work of Kloetzer, Mahulea, and Gonzalez proposes an optimization in this method, improving the points of passage of the robot in the free cell [39].

2.4.3 Sampling-Based Path Planning

The methods for path planning presented above, require a representation of the free space of the environment. When the environment is large, it can take a long time to process the algorithm. Sampling-based methods do not require calculation of free configuration space which is a time-consuming operation for complex obstacle shapes.

In sampling-based path planning, random robot configurations are sampled, then collision detection methods are applied in order to verify if these points belong to the free space. From a set of such sampled points and connections among them (connections must also lie in the free space) a path between the known start and the desired goal point is searched [20]. Sampling-Based path planning method can be divided into two types: Rapidly Random Tree (RRT) and Probabilistic Roadmap (PRM).

Rapidly Random Tree

The RRT is described in [40]. This approach has as its characteristic a single starting and ending point. It prioritizes the speed to locate the path. This algorithms normally focus only on parts of the environment that are more promising to find the solution. New points and connections to the graph are included at runtime until the solution is found. Figure 2.18 shows the behavior of this algorithm to find the desired path.

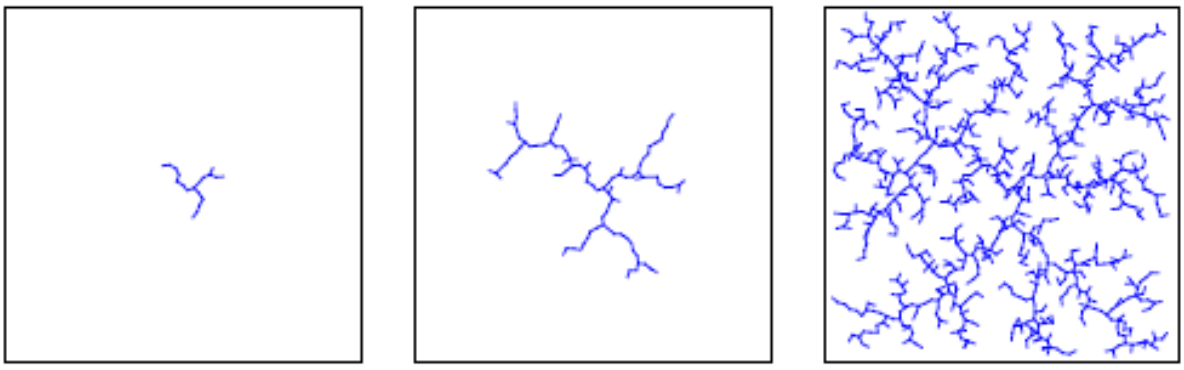


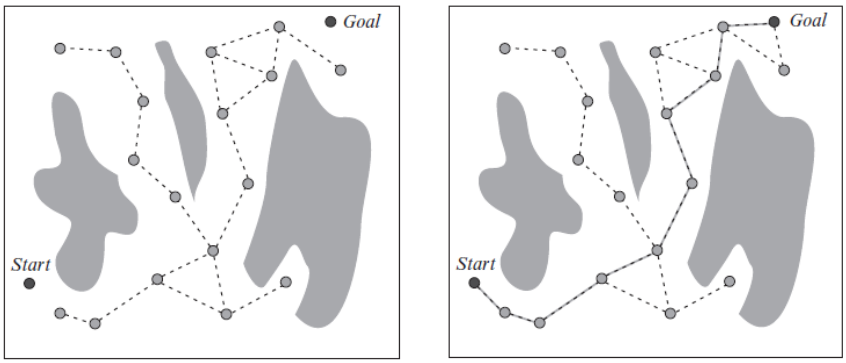
Figure 2.18: Progress (from the left to the right figure) of the RRT algorithm to find the path by connecting the desired points [20].

The trajectory planner used by Vaz is based on the RRT algorithms, which are based

on a unique search for kinematic and dynamic planning. The construction of random trees explore all possible trajectories while checking if the final goal was reached [41].

Probabilistic Roadmap

The PRM is described in [42]. PRM fully exploits the fact that it is cheap to check if a single robot configuration is in free-space or not. The algorithm is composed by two steps: First (see Figure 2.19a) PRM creates a roadmap in free-space. It uses rather coarse sampling to obtain the nodes of the roadmap and very fine sampling to obtain the roadmap edges, which are free paths between node configurations.



(a) Algorithm learning (b) Path searching

Figure 2.19: PRM algorithm [35].

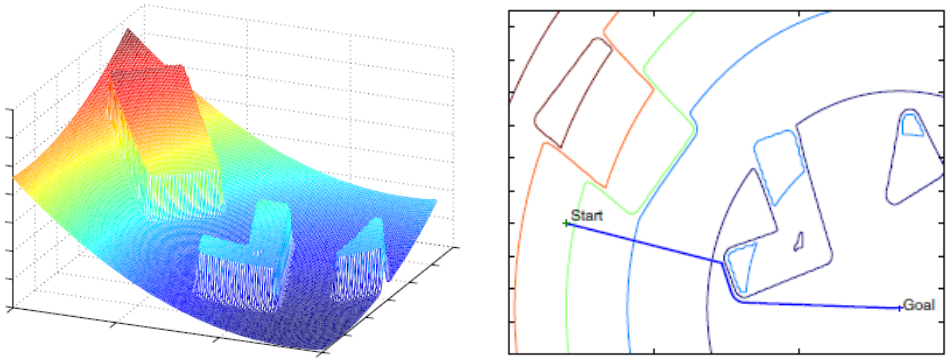
In the second step (see Figure 2.19b) the roadmap has been generated, planning queries can be answered by connecting the user-defined initial and goal configurations to the roadmap. Initially, node sampling in PRM was done using a uniform random distribution [38].

2.4.4 Potential Field

The techniques used above are based on methods for path planning from the classification in free space and space with object. The potential field uses a different methodology, basically this approach creates the map that the robot is inserted based on a potential field. This method treats the robot as a point load, on the influence of a potential field.

The forces act on the robot in the sense of taking it to its goal. The idea behind this approach is to make the robot be attracted to its goal while it is repulsed by the obstacles that are known [38].

The path planned by the robot will be in accordance with the overlapping gradients of the fields of attraction and repulsion that will act on it, show in Figure 2.20a. The field of attraction is in function of the goal point and the field of repulsion is due to the obstacles. The sum of these two fields represents the interaction between the potential field and the path the robot will take to reach its goal.



(a) Environment with three obstacles. (b) Resulting path.

Figure 2.20: Potential field algorithm [35].

Latombe, describe in his book that, "In comparison to other methods, potential field methods can be very efficient. However, they have a major drawback. Since they are essentially fastest descent optimization methods, they can get trapped into local minima of the potential function other than the goal configuration" [35]. An example of trapped path is showed in Figure 2.21.

The community of researchers who work with potential field for path planning, bring some extended approaches of this method seeking to optimize it, due to efficiency and simplicity of implementation that potential field presents. The paper of Guo, Gao, and Cui, presents an application to improve the speed of convergence and optimize the path of this method of path planning, besides mitigating the the local minimum problem, combining the strategies of artificial potential field and navigation potential field as can

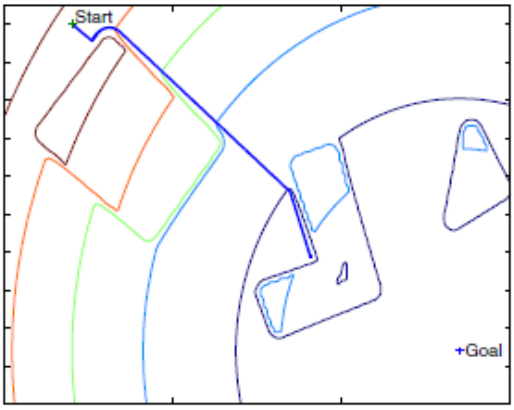


Figure 2.21: Local minima present in potential field, forming a trapped path [35].

be seen in [43]. Work [44] also presents a path planning based on derivation of potential field. It presents different ways to carry out a planetary exploration, where the potential field will be in function of the irregular terrain existing in these environment.

Above, the theoretical concepts of path planning were introduced and the most used approaches were listed. However, it is worth mentioning that there are many methods for robotic path planning. Such as Sariff and Buniyamin present in a review on the path planning algorithms for autonomous robots discussing its strengths and weaknesses [45].

2.4.5 A* Algorithm

In order to find a trajectory in a free space, considering a discrete map of the environment, taking the robot from an initial position to an end position, the algorithm A * may be a great approach to path planning. According to Pedro Costa "The algorithm is complete, optimal and a complexity of time and space depends on the heuristic" [46]. This algorithm for path planning is based on a cost function $f(n)$ as described in equation 2.1:

$$f(n) = g(n) + h(n) \tag{2.1}$$

The term $g(n)$ refers to the cost of moving a node to its neighbor n. The term $h(n)$ is the estimated cost of the n node for the target node. Figure 2.22 presents a simple example to understand the performance of algorithm A *, where it can be analyzed the cost function

makes clear the reason for choosing the trajectory by the algorithm.

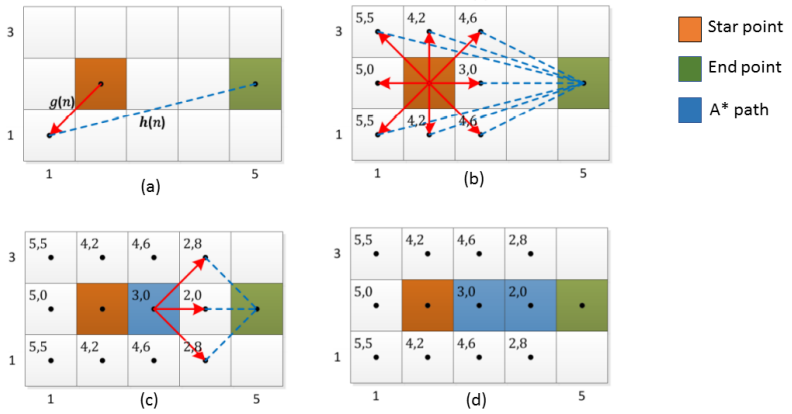


Figure 2.22: Example of a path generated by algorithm A* (adapted from [47]).

In [46] it performs improvements and modifications to the A * algorithm to increase its performance relative to search time and path planning. In this way the algorithm was implemented in autonomous robots for robotic soccer tournaments where there are several dynamic and unknown obstacles. The robot's trajectory must be re-planned in real time avoiding collisions and performing the desired tasks.

2.4.6 Genetic Algorithm

Within evolutionary computing, which encompasses an increasing number of paradigms and methods, Genetic Algorithm (GA) with random search controlled by probabilistic criteria are considered one of the most important. These methods have been used in solving optimization, scheduling and planning problems. In a population of possible solutions to a given problem, GA evolves according to probabilistic operators, so there is a tendency for individuals to represent increasingly better solutions as the evolutionary process continues.

According to Tuncer and Yildirim "Path planning for a mobile robot finds a feasible path from a starting node to a target node in an environment with obstacles. GA has been widely used to generate an optimal path by taking advantage of its strong optimization ability" [48]. The GA seeks to optimize trajectories with mutation and cross-over

operations. In [48] GA was used with search algorithm to carry out the path planning of starting point and end point avoiding obstacles and collisions in the environment. This environment being able to be static or dynamic, where it used an optimization in the mutation operator to optimize the path or seek a path near the optimum. Already in [49] applying crossover and mutations to search for an optimized path, using a connectivity grid to represent the plant where the robot is inserted, the objective is to find the lowest path between the start and end points, avoiding repeating cells along the way, simplifying the fitness function by analyzing path length.

GA-based methodologies have the problem of requiring a great deal of time to be able to converge to or near an optimal path. Therefore, it is not an attractive option to work on real-time planning, i.e., to update the trajectory initially established, because during the re-planning the robot would be in a waiting state without movements. However for problems in static environments with previously known obstacles is a good strategy to be adopted.

2.5 Travelling Salesman Problem

The well known Travelling Salesman Problem (TSP) is characterized by an optimization and combinatorial analysis of paths. To illustrate the problem, a salesman who wants to visit a set of cities or customers, passing exactly once each and returning to the starting city at the end of his journey. The objective is to minimize the total distance traveled by the salesman [50].

In advanced math, the TSP despite having a simple description is classified as an NP-complete problem, i.e, nondeterministic polynomial time complete. In other words this class (NP-complete) consists of problems that there are still no systematic methods or algorithms capable of finding a solution in polynomial time [51]. According to Assunção Gomes Costa, due to the complexity of this problem, in most real cases, the solutions for TSP are obtained by heuristic methods [52].

The formulation of this problem is not limited to finding better routes between different cities. It has a wide application in real situations. TSP forms the basis for many problems in logistics, finance and engineering [53]. In optimizations for route planning TSP is widely used, considering different optimization objectives such as: minimization of distance traveled [54], minimization of energy or fuel consumption [55] and minimization of travel time [56].

The traveling salesman problem is easily observed in practical problems of logistics, such as transportation and garbage collection, transportation of students to their homes, transport and distribution of orders, visits of medical and nurses in patients' homes, among other activities [54]. In robotics, the concept of TSP is commonly used to solve problems of planning the best routes for robots, where we can mention the planning of better routes to optimize the time of tasks performed by manipulators [57], or in mobile robotics the sequence of points that the WMR must visit to optimize the distance on its path [58].

Figure 2.23 presents a simplistic illustration of the problem. It depicts four cities and six roads between them. The starting point is in the highlighted city, being necessary to visit all the other cities at least once to return to the city of origin. Table 2.1 presents possible solutions to the problem, where the costs of each trajectory varies according to each specific problem.

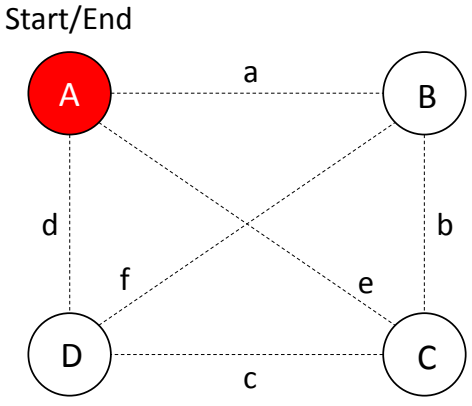


Figure 2.23: Example of a travelling salesman problem.

Table 2.1: Solution for the TSP exemplified in Figure 2.23.

TSP Solution	Path	Path Cost
1	A - B - C - D - A	a+b+c+d
2	A - B - D - C - A	a+f+c+e
3	A - C - B - D - A	e+b+f+d
4	A - C - D - B - A	e+c+f+a
5	A - D - B - C - A	d+f+b+e
6	A - D - C - B - A	d+c+b+a

Chapter 3

System Architecture

This chapter will describe the software and hardware components used in the proposed approach to perform the scan in an environment with the presence of radioactive or toxic substances. The Figure 3.1 presents the proposed structure and its data flow. The system consists of four main blocks: ControlApp, Real Robot, System Localization and SimTwo Simulator. This system has easy switching between the real environment and the simulation environment, since it share the same communication protocol.

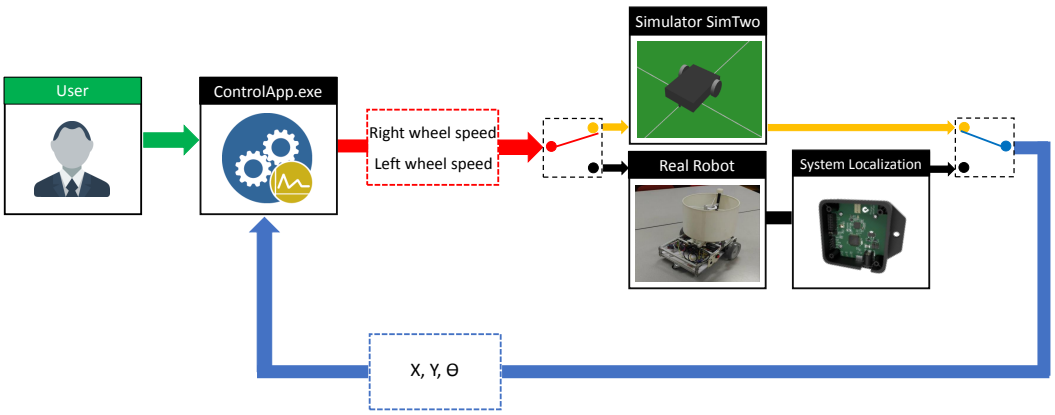


Figure 3.1: System architecture proposed for the interaction between the user and the robot.

The real robot used in this work was previously made by the supervisors for other projects that encompass mobile robotics.

3.1 ControlApp and Communication

ControlApp is a graphical application for remote access to the robot, developed in the Lazarus programming platform. ControlApp deals with real and simulation environment. This consists of an interface between man and the robot developed for the user to be able to obtain information related to the simulation and real environment such as, speed, position and orientation of the robot in real time. The Figure 3.2 presents the developed interface. The highlighted points A, B and C will be explained below.

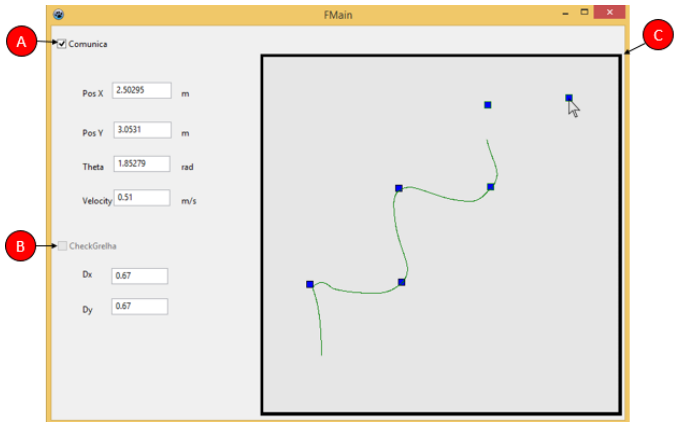


Figure 3.2: Interface between ControlApp and the user, highlighting its functionalities: A - Communication, B - Inserting Waypoints and C - Robot navigation area.

Point A displays the checkbox that enables and disables communication between ControlApp and real or simulated WMR. The communication protocol used is User Datagram Protocol (UDP) (More information about this protocol can be found in [59]). Each device (real robot, simulated robot and App control) has an IP to be identified. A packet encoding the robot position, orientation and distance sensor data is sent from robot to ControlApp, whereas a packet containing the right and left speed wheels is sent from ControlApp to robot.

Point B displays a checkbox to select the mode of insertion of the waypoints to construct the trajectory realizable by the robot. When the Checkbox is not selected, the user can interact with the robot through the mouse, indicating the sequence of waypoints desired for the trajectory. In the case where the checkbox is selected, a connectivity grid is

created through the values Dx and Dy inserted by the user, which represents the distance between waypoints in x and y. An example of a connectivity grid can be seen in Figure 3.3, where the environment has a dimension of 4 meters long by 4 meters wide and the waypoints are equidistant 0.67 meters each other.

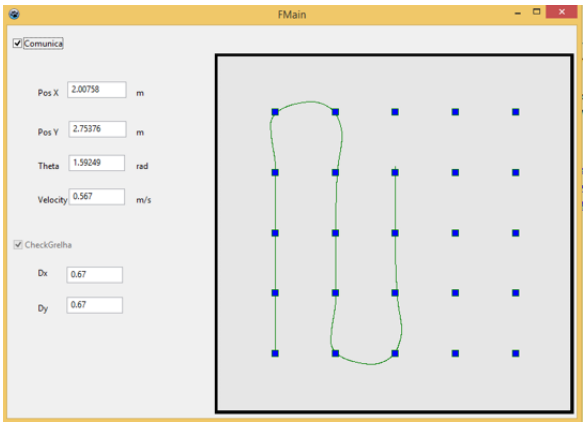


Figure 3.3: Connectivity grid where resolution of the grid is defined by the user.

Point C shows the robot's navigation area bounded by the existing square. The inner area of the square can be replaced by the 2D plant or even an top view image of environment that the robot will scan. The route that the robot executes is presented by the green line which is updated in real time.

Other relevant functionalities of the ControlApp are not presented in the interface are the calculations of the wheel speed control of the robot, the robot's path planning and the calculation of the trajectory. Note that this software runs on a computer external to the robot.

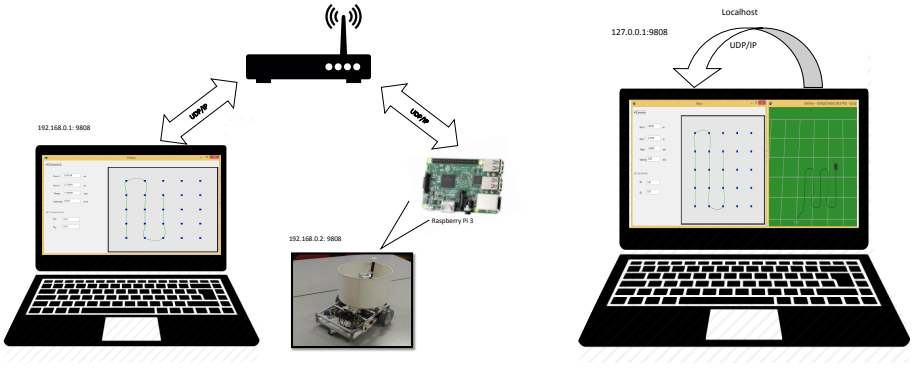
Communication Between ControlApp and Real Robot or Simulated Robot

Due the ControlApp runs on a computer external to the robot, it is necessary to implement a way of communication that allows the transfer of data. As will be discussed in section 3.2 the robot has integrated Wi-Fi, which makes it possible to establish a communications network between ControlApp and Robot.

The communication system must occur in real time and allow the exchange of different

types of data in two modes, loopback network for tests in simulation, and in a Wi-Fi network, for implementation in the robot prototype.

The adopted network protocol was UDP. Although it does not guarantee the reception of the data, in contrast to the Transmission Control Protocol (TCP) protocol that allows reception with reliability, which makes TCP slow, the UDP has high speed [60]. In this way, ControlApp must previously know the IP address of the robot that is connected in the same network, besides the port available for the communication and its identification establishing a system UDP/IP as described in Figure 3.4a where the communication port is 9808. Tests and simulations occur in a loopback network as shown in Figure 3.4b.



(a) Real robot. (b) Simulated robot.

Figure 3.4: Communication between robot and ControlApp.

3.2 Real Robot Description

A mobile robot prototype presented in Figure 3.5 was provided by the supervisors of this dissertation, having in mind the scanning of the toxic and radioactive field in a room. This prototype will prevent a human is exposed to an environment of radiation risks. This section will describe the features of the real robot. Initially it will be approached the structural part that makes up the body of the robot. Then, it will describe the cinematic model of differential robot. Further, it will be presented the robot's electronic schema and the section will conclude with the description of the software that operates on the

robot.

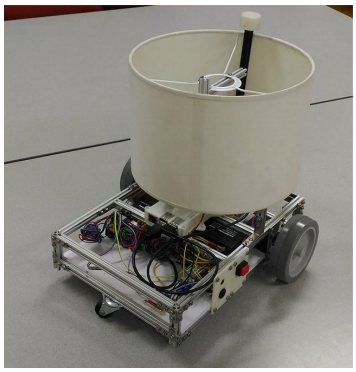


Figure 3.5: Prototype of real mobile robot.

3.2.1 Structural Constitution of the Robot

The WMR is designed to perform the scan moving in a plane environment. Its supporting structure is entirely composed of aluminum extrusion profile in X, as shown in Figure 3.6. This material was selected because of its high mechanical strength, its rigidity and its lightness, thus providing robustness and durability to the mobile robot. At the top, the robot has an object with a circumference of 25 cm in diameter, which has utility for the ground truth system of location that will be described later in this work in subsection 3.4.3.

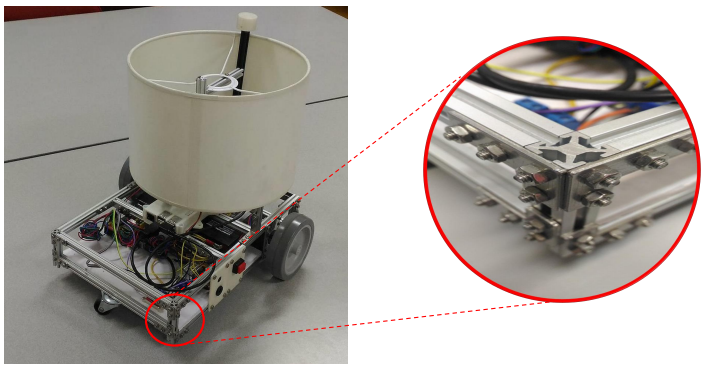


Figure 3.6: Detail of the support structure of the robot built with aluminum extrusion profile in X.

The robot has a differential wheel drive system. With this geometry the robot has three

degrees of freedom; $[x, y, \theta]^T$. The prototype has three wheels, two of which are coupled to independently stepper motors. The other is castor wheel which has the support function. This model of WMR is widely used because of the simplicity of construction and control [61].

The Table 3.1 presents the dimensions of the mobile robot structure:

Table 3.1: Dimensions of real WMR

Robot Description	Dimension	Unit
Width	0.28	m
Length	0.35	m
Wheel diameter	0.1	m
Wheel thickness	0.025	m
Robot mass	3.4	Kg

3.2.2 Kinematic Model of Differential Robot

According to Roland Siegwart "Kinematics is the most basic study of how mechanical systems behave. In mobile robotics, we need to understand the mechanical behavior of the robot both in order to design appropriate mobile robots for tasks and to understand how to create control software for an instance of mobile robot hardware" [4].

This subsection will be dedicated to present the mathematical notation that represents the kinematics involving the robot based on the literature in the area [4], [20]. It will be described how the movement of a differential robot occurs. For a better understanding, the kinematic model will be divided into three parts: pose, linear and angular velocity and the last part will present the non-holonomic constraint model.

Pose

The WMR has three degrees of freedom, and its pose in the 2D plane is defined by the state vector in a global coordinate frame (X_g, Y_g). That define an arbitrary inertial basis on the plane as illustrated in Figure 3.7 and represented by equation 3.1.

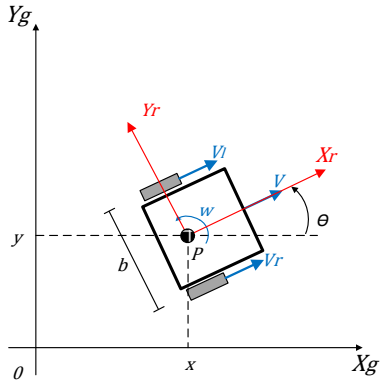


Figure 3.7: Robot in the plane representing the global robot reference frame.

$$\xi_g = \begin{bmatrix} x(t) & y(t) & \theta(t) \end{bmatrix}^T \tag{3.1}$$

The robot frame is in motion attached to the robot (X_r, Y_r) . The relation between the global and robot frame is defined by the rotation matrix $R(\theta(t))$ as show equation 3.2 and 3.3.

$$R(\theta(t)) = \begin{bmatrix} cos(\theta(t)) & sen(\theta(t)) & 0 \\ -sen(\theta(t)) & cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

$$\xi_r = R(\theta(t)) \cdot \xi_g \tag{3.3}$$

Linear and Angular Velocity

The velocity of each wheel is controlled by a step motor. Deriving the state of the robot in the global reference frame, the speed relation is obtained as shown in the equation 3.4:

$$\dot{\xi_g} = \begin{bmatrix} \dot{x}(t) & \dot{y}(t) & \dot{\theta(t)} \end{bmatrix} = \begin{bmatrix} cos(\theta(t)) & 0 \\ sen(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} V(t) \\ w(t) \end{bmatrix} \tag{3.4}$$

The input variables of this system are linear V and angular w velocities of the robot, obtained by the wheel speeds as shown by equations 3.5 and 3.6:

$$V(t) = \frac{V_r(t) + V_l(t)}{2} \tag{3.5}$$

$$w(t) = \frac{V_r(t) - V_l(t)}{b} \tag{3.6}$$

Where b is a constant that represents the distance between the traction wheels of the robot at its point of contact with the ground.

Non-Holonomic Restriction

For any state $[x(t), y(t), \theta(t)]^T$ of the robot, the motion must respect the constraint imposed by the equation 3.7 preventing sideways slippage, i.e, avoid movements perpendicular to the axis of the robot's linear velocity as shown in 3.8 in case of a lateral parking attempt.

$$\dot{y}(t) \cdot cos(\theta(t)) = \dot{x}(t) \cdot sen(\theta(t)) \tag{3.7}$$

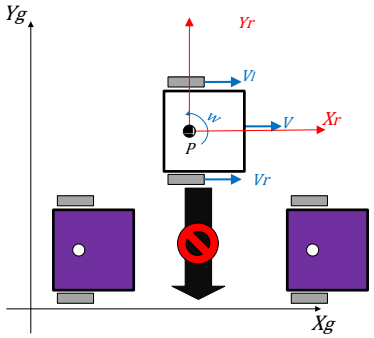


Figure 3.8: Example of a restricted motion for a mobile robot with differential drive.

3.2.3 Electronic Hardware

The mobile robot, in which the practical tests were carried out, has an elaborate electronic structure to move through the room independently. For that, the WMR has embedded on its base some electronic components, battery, sensors, Raspberry computes and Arduino shields. In this section will present these components. It will not address detailed working concepts of the components, because it is out of the main scope of work. The block diagram of Figure 3.9 shows the high-level component block. Note that the electronic architecture on the robot is arranged which has been designed to be compact to fit over the structure and with low consumption for battery operation.

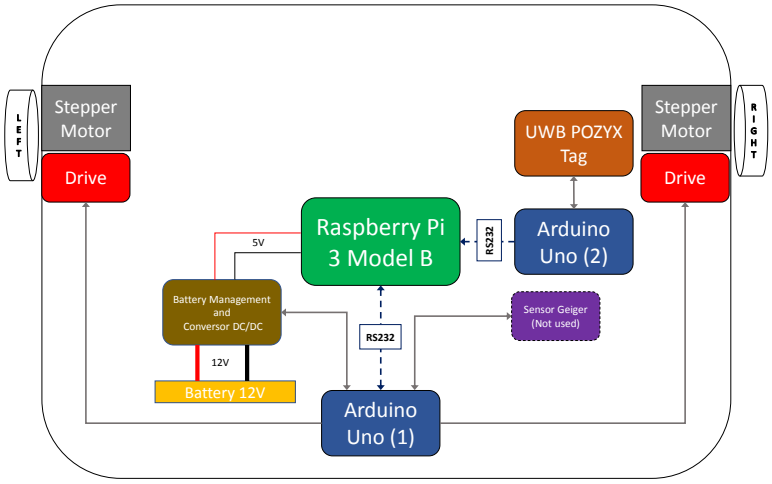


Figure 3.9: Mobile robot electronic architecture.

The following subsections will be devoted to the description of the components that compose the robot.

Raspberry Pi 3 Model B - The Robot's Brain

The electronic hardware for the mobile robot base has as its main component a raspberry pi model 3B. This is a microcomputer of low cost and small dimensions. The microcomputer has 1 GB of RAM, a Processor BCM2837 64bit ARMv8 Cortex-A53 Quad-Core processor with clock of 1.2 GHz. It comes with 4 USB 2.0 ports and a microSD card slot. It has integrated Wi-Fi avoiding the use of adapters. Figure 3.10 shows a raspberry pi.

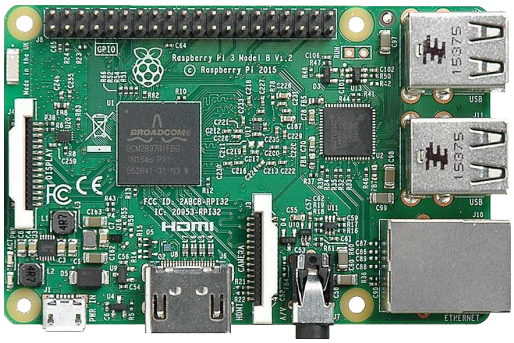


Figure 3.10: Raspberry Pi model B.

The microcomputer is powered by a step-down DC/DC converter which has a 12V battery voltage input and provides a 5V to Raspberry Pi. The operating system used is raspbian. The microcomputer run an application that is responsible for performing Kalman Filter calculations, communicating with the ControlApp over Wi-Fi and communicating with two Arduinos via USB (emulated serial port). It is responsible for processing the localization data of the Pozyx sensor obtained through the Arduino (2). It also sends and receives data to the Arduino (1) which matches the speed motors and voltage and current of the battery.

Arduino Uno (1) and (2)

The Arduino Uno (see Figure 3.11) is a development board that features an 8-bit ATmega328 microcontroller with a 5V operating voltage. It has 14 analog ports and 6 digital ports. Its clock capacity is 16 MHz and its flash memory has 32 KB. The real robot has in its architecture two Arduinos Uno, due to its ease of programming and wide connectivity with diverse shields.

The Arduino (1) is directly connected to the CNC shield V3 and to the Allegro MicroSystems-a4988 chip composing the power drive and control of the stepper motor which will be explained in sub-subsection 3.2.3. In addition to motor control, the development board is also responsible for calculations of odometry and has connection with shields to obtain values of current, voltage, power management and the Gayer sensor module, the latter being not implemented in this work.

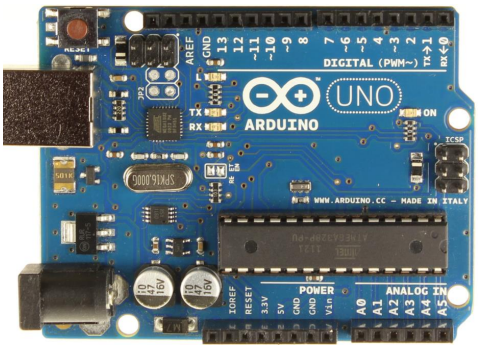


Figure 3.11: Arduino Uno.

Arduino (2) is directly connected to the Ultra-wideband (UWB) Pozyx tag, as shown in Figure 3.12. Thus the development board together with the Pozyx tag receive the data from the four Pozyx anchors scattered around the environment where the robot will scan by calculating the location (x, y) of the robot. More details about the operation of Pozyx will be presented in subsection 3.4.1.

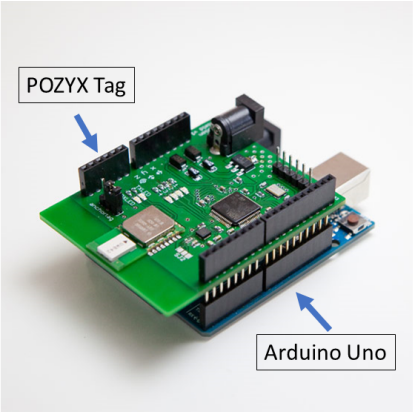


Figure 3.12: Connection between Pozyx tag and Arduino (adapted from [62]).

The data flow and change of information between the Arduino and the Raspberry Pi 3B microcomputer is through serial communication with USB (emulated serial port) protocol.

Drive and Step Motor

The robot uses two NEMA 17HS16 step motors. Each coupled to one of the traction wheels. Each step of the motor has a resolution of 1.8 degree providing considerable precision for the odometer calculations.

It uses a CNC shield V3 as interface between the Arduino and two Allegro MicroSystems drivers - A4989, each one responsible for one motor. In this way the stepper motor receives a sequence of logical levels of voltage that will feed its coil, where it will pass current, resulting in an electromechanical transformation, moving the wheels of the robot. Figure 3.13 shows how the Arduino is coupled together with shield CNC V3 and Allegro MicoSystem chip a4989.

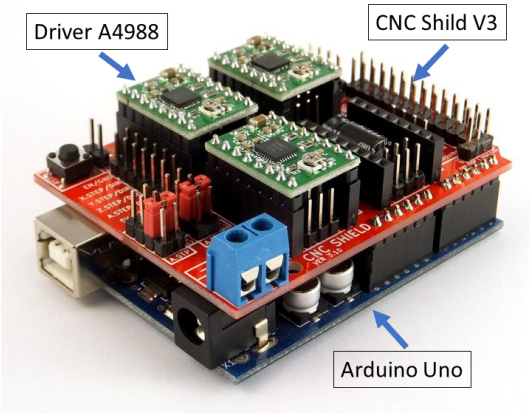


Figure 3.13: Connection between Arduino, CNC shild V3 and chip Allegro MicroSystems A4989.

UWB Pozyx Tag

The task of scanning an environment, it is necessary to obtain the precise localization of the robot. As the localization occurs in an indoor environment, the positioning system Pozyx based on the technology UWB is used. The system has a total of five components, of which four are immobile anchors scattered around the environment and a mobile tag which is in WMR. In this sub-subsection will be presented only the tag, because it integrates the robot. The Pozyx system will be described in subsection 3.4.1 in more detail.

According to the manufacturer, the Pozyx tag is an Arduino compatible shield that provides accurate positioning and motion information. The Pozyx tag connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top [62], as shown the Figure 3.14.

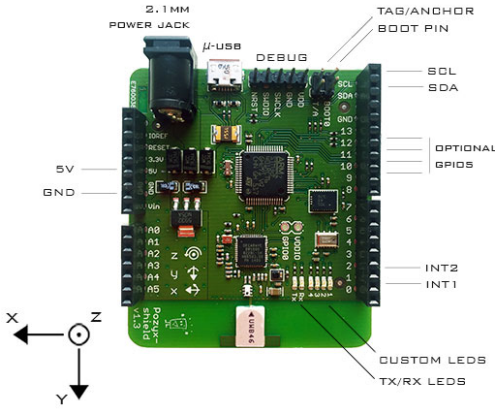


Figure 3.14: Interface of Pozyx system tag to connect at Arduino [62].

3.2.4 Robot Software

The robot sofware is an application developed in Lazarus which is responsible for various operations, such as localization and movement. This application runs on the Raspberry Pi integrated into the robot and manages communications with Arduino (1) and (2) as well as ControlApp. The flow of the operations performed by the application was modeled by a petri net to represent the system in a discrete mode conduct by events which can be verified in Appendix B.

The application is responsible for calculations of odometry, which together with the data obtained from Pozyx performs the sensory fusion through the Kalman filter, estimating the robot's pose in the environment. The pose value is then encoded and transmitted by UDP/IP to the ControlApp which will return the wheel speed information. The software, when decoding this information, transmits it to the stepper motors that will act on the wheels.

3.3 Description of the Simulation Model

To test the algorithms and methodology used to perform the scan of an environment with a mobile robot was used the SimTwo Simulator [63]. According to its own developer Paulo Costa, the SimTwo is "a realistic simulation system that can support several types of robots. Its main purpose is the simulation of mobile robots that can have wheels or legs, although industrial robots, conveyor belts and lighter-than-air vehicles can also be defined" [64]. By using this simulator it was possible to validate the approach in a 3D virtual environment before performing the tests for practical validation. Figure 3.15 displays the workspace of this tool. This section will preset the model performed for the simulation describing the simulated robot, the scenario and method used to insert the uncertainty in the Pozyx system. Further details of the functions and tools of the simulator can be found in [64], [65].

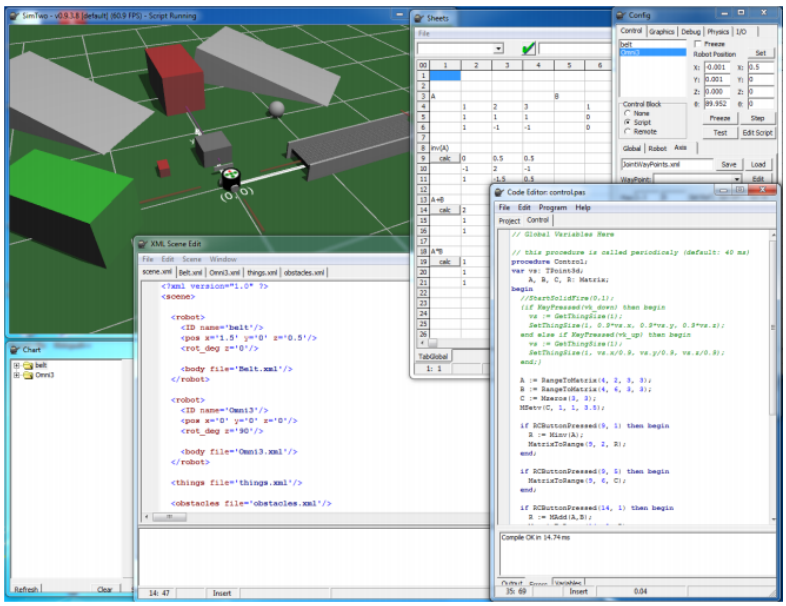


Figure 3.15: Overview of SimTwo simulator tabs and tools [64].

The modeled and simulated prototype is presented in Figure 3.5. The dimensions of the virtual robot are faithful to that of the real robot as shown in the Table 3.1. The simulator represents the dynamic and kinematic behavior of the robot through the Open Dynamics Engine (ODE) (an open source library) which helps the avoidance of the need

to handle the equations directly [66]. The structure of the simulated robot is defined using boxes, cylinders and joints (connections between bodies and wheels), which are arranged to form a structure similar to that of the real robot, as can be seen in Figure 3.16.

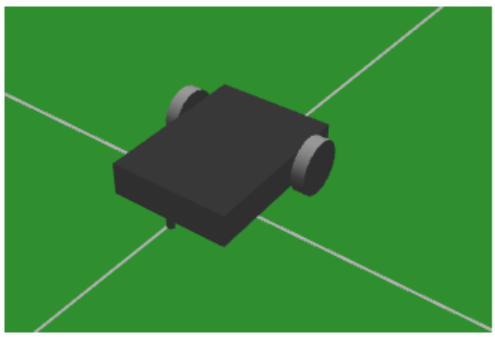


Figure 3.16: Simulated robot.

The simulator has a tab named "code editor" which represents the Integrated Development Environment (IDE) where a Pascal-based code is implemented, referring to the actions that the robot will perform inside the simulator, i.e. its control parameters [66]. In this tab it is also implemented the communication UDP/IP between the simulator and ControlApp already presented in section 3.2. The WMR exact pose (x, y, θ) in simulation environment is easily obtained with an existing function in the simulator. On the other hand, real localization system are subject to errors and uncertainties due to the real sensors. Previous work [67] addressed the estimation for the pose of the real robot using the Kalman filter to perform the fusion between the odometry and the UWB-ToF sensor. It also outputs the model of the covariance matrix errors for the positioning (equations 3.8, 3.9 and 3.10).

$$\overline{x}_{error} = -0.074 \tag{3.8}$$

$$\overline{y}_{error} = -0.013 \tag{3.9}$$

$$R = \begin{bmatrix} 0.0015 & 0.0002 \\ 0.0002 & 0.0005 \end{bmatrix} \tag{3.10}$$

The values obtained in $\overline{x}_{error}$ and $\overline{y}_{error}$ provide the average of the errors in relation to each coordinate. The R is the noise covariance matrix. These model was used to

estimation of the location of the robot in the simulator. The θ component of the robot pose is not influenced by noise in the simulation, due to the fact that the Pozyx sensor does not provide this component, only x and y.

3.4 Structure and Operation of Localization Hardware

In this section will describe the hardware used in the approach proposed for the low cost indoor location to WMR. Currently systems based on artificial vision with cameras have high-accuracy. However, these high-accuracy systems have a high price disadvantage, which can cost more than a thousand euros. Another disadvantage is the blind spots and the influence of the luminosity that can change in the environment. Due to these negative points an alternative approach to locating a mobile robot was adopted.

Various methods can be found for the location of the mobile robot, such as Radio Frequency (RF) technologies that are widely diffused because of their relatively low cost in addition to the low processing power required. Two approaches are well accepted by the academic community for localization using RF, which are: *(i)* The technique Received Signal Strength (RSS) which measures the distance between the sender and the receiver using signal power received at the receiver. *(i)* The technique Time of Flight (ToF) measures the difference between the instant of time at which the signal is transmitted and the time of arrival at the receiver. With the interval of time and the speed of wave propagation in the environment, the distance between the transmitter and the receiver is obtained.

Different solutions with RF are found, for example: Wi-Fi with an average error estimate of 3-5 m. Systems with Radio Frequency Identification (RFID) have an accuracy with average errors of 2 m. Bluetooth solutions has 1.5 m associated average error in the location. Ultra-wideband systems UWB are capable of decreasing the mean errors to a range of 10 to 20 cm [68]. This way UWB is an approach with great potential for indoor

location. For the present work, we selected Pozyx that uses the radio waves UWB to find the robot in real time in the task of performing a scan, presenting a price below one thousand euros.

In this section, it will be discussed the Pozyx system and its components, as well as the robot's odometry. These two tools provide data for the Kalman filter, where they are fused, optimizing the location. To evaluate the proposed system, a Laser Range Finder (LRF) is used as ground truth which will also be detailed.

3.4.1 Pozyx UWB Based

UWB has received some attention within the robotics community. It is considered one of the most promising indoor positioning technologies currently available, especially due to its fine time resolution [67]. Due to its increased bandwidth it not only avoids interference with other types of RF signals but also deals with obstacles and walls since the signal can go through them [69], being a solution for cases of Non-Line-of-sights. Its principle is based essentially on sending radio signals, which travel at the speed of light, from a mobile transceiver (tag) to a set of known anchors, measuring the Time of Flight (ToF) and consequently the distances to this set of points [69].

Throughout this work the Pozyx system [62] will be used as the main hardware for the location in a 2D environment. It is a low-cost time-of-flight technology with ultra-wideband localizing in 2D e 3D. The hardware used consists of five elements. Being a tag that is incorporated in the robot (Figure 3.12) and four anchors fixed and scattered in known places, used as points of reference of the environment to be scanned.

The Figure 3.17 presents in a block diagram the components in the Pozyx Tag. Note that the Pozyx anchors are similar but without the motion sensors as they are assumed to be immobile. The main component is Micro-controller Unit (MCU) that provides all the Pozyx functionalities. It communicates with all tag components as well as being responsible for localization algorithms. Its operation is in real time, thus obtaining a better performance. The MCU also provides an interface to any external device (such as

the Arduino). The Pozyx tag is also equipped with an UWB chip (Decawave DW1000 [70]) that provides the wireless capability of the device [62].

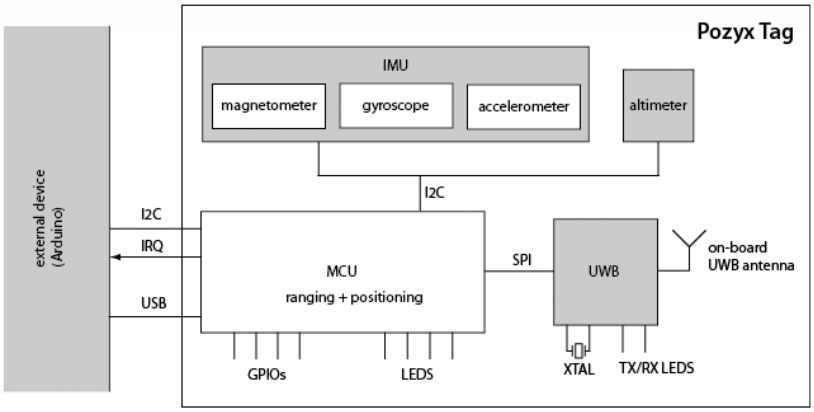


Figure 3.17: Pozyx tag high-level component blocks [62].

Since the signal UWB has a large bandwidth (in the Pozyx case 500 MHz), the power of its signal must be very low so as not to interfere with other transmissions that operating in the same frequency band of RF and which are regulated. Consequently the Pozyx system has a low power consumption. The tag is powered by the robot's own battery. Anchors can be powered by power banks. In this work the power banks used have a capacity of 2000 mAh. Figure 3.18 shows one of the four anchors used in the work.

Figure 3.18: Pozyx anchor using a power bank as energy source.

To locate the WMR in a 2D plane it is necessary to use three or more anchors. In our work four anchors were scattered in the environment to perform the scan. Its positions

must be static and defined. According to its developers, the arrangement of the anchors should not be in line, i.e, side by side due the fact of decreasing the accuracy of the data obtained. Figure 3.19 shows the method used in this work for the arrangement of the four anchors, which are located near the corners of the environment.

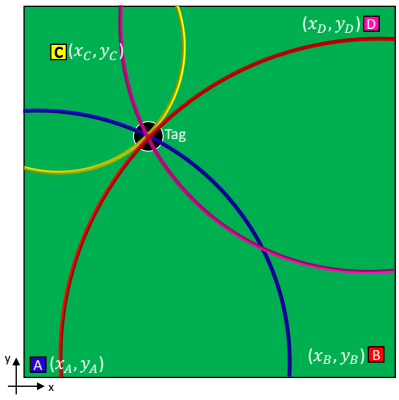


Figure 3.19: Arrangement of Pozyx anchors and tag location radius.

The principle of operation of the Pozyx system occurs with the transmission and reception of the signal RF UWB. Knowing the propagation time of the wave in the environment, for this case the speed of light (c about $3{\cdot}10^8$ [m/s]), and the ToF required to transmit a signal from the tag to the anchor or vice-versa, the absolute distance (equation 3.11) between the devices[68].

$$d = c \cdot ToF \quad [m] \tag{3.11}$$

.

3.4.2 Odometry

Odometry is a simple and widely adopted method for the current localization of mobile robots [71]. It is based on previous information about position and orientation, that is, the robot pose (x, y, θ) is estimated cyclically from odometry information. This technique consists in integrating the incremental information of the linear movement of the driving wheels over time, that is, in measuring the rotation of the driving wheels, in order to

determine the distance traveled by the robot, its speed and the direction of movement. As a relative localization method, its use in practical applications almost always aims to provide real-time actualization of position and orientation of the vehicle during the time intervals in which the absolute location method (Pozyx) is periodically executed [24].

The main advantages of odometry are associated with its low cost, the high sampling rate allowed and the good accuracy achieved. However, the main restriction concerns the unlimited accumulation of errors along the navigation, which is due to the integration of the errors with which the measures of the movement of the wheels are affected. This fact makes that the estimation of the pose of the robot provided by this method departs from its real values, making it impossible for this method to be used to locate the robot, being necessary the fusion through mathematical tools with an absolute location method [24].

In practical cases the errors accumulated in the location by the odometry can be classified as systematic errors and non systematic errors [24], [71]. Systematic errors can be as follows:

- Wheels with different diameters;
- Real wheel distance different from that measured and used in calculations;
- Wheels off-axis, i.e. ,out of alignment;
- Deformation of the wheels, due to manufacturing defects or excess weight;
- Variation of the diameter of the wheels with the load or distribution of the load in the robot;

Non-systematic errors occur due to the interaction between the robot and the floor of the environment that the robot will perform its task, and the following cases may occur:

- Sliding or skidding of the robot;
- Failure to execute the motor steps, ie, motor slip due to acceleration;
- Irregular soil

- Execution of fast curves;
- Lack of contact between wheels and ground;

Due to this set of errors, it is necessary to perform treatments and calibrations to optimize the estimation of the pose performed by odometry. Information about these optimizations can be found in [24], [72].

Mathematical Model of Odometry

Here will present the mathematical model to estimate the location of the mobile robot with differential geometry based on the kinematic model presented in subsection 3.2.2. Figure 3.7 presents the robot in the global coordinate system X_gY_g. Expanding the equation 3.4 results in the equation 3.12:

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} V(t) \cdot cos(\theta(t)) \\ V(t) \cdot sen(\theta(t)) \\ w(t) \end{bmatrix} \tag{3.12}$$

Model 3.12 can be written in discrete form using Euler integration and evaluated at discrete time instants t = kT_s, $k = 0, 1, 2, \ldots$ where T_s is the following sampling interval [20]:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + V(k) \cdot T_s \cdot cos(\theta(k)) \\ y(k) + V(k) \cdot T_s \cdot sen(\theta(k)) \\ \theta(k) + w(k) \cdot T_s \end{bmatrix} \tag{3.13}$$

3.4.3 Laser Range Finder

In order to evaluate the quality of the localization system composed by the sensors and algorithms proposed in this work, a ground truth based on a Hokuyo URG-04LX LRF was used (Figure 3.20). Using the laser together with an application in Lazarus, the system is able to determine the real position of the mobile robot in a 2D plane (x, y). This data

should be compared with the values obtained from the system localization (Pozyx) to validate the path performed by the robot.

Figure 3.20: Laser Ranger Finder Hokuyou URG-04LX.

The LRF scans the area around it to determine the distance of closest objects, providing a two-dimensional map of the environment with high accuracy. The principle of operation is based on the emission of a laser beam. A rotating mirror changes the beam's direction, then the laser hits the surface of an object ans is reflected. The direction of reflected light is changes again by a rotating mirror, and captured by the photo diode. The phases of the emitted and received light are compared and the distance between the sensor and the object is calculated [73]. The relevant specifications obtained from the manufacturer for the Hokuyo URG-04LX can be checked in Table 3.2.

Table 3.2: Sensor laser rangefinder URG-04LX specification [74]

Specification	URG-04LX	Unit
Measuring area	20 to 5,600	mm
Accuracy	60 to 1,000 :±30 or 1,000 to 5,600 ±3%	mm
Angular resolution	0.36º (360º/1,024)	deg
Scanning time	100	ms/scan
Power Consumption	4.0	W
Weight	0.16	Kg

Figure 3.21 shows an illustration of the ground truth approach. The LRF is installed in the origin of the Cartesian plane that represents the environment that the robot is

inserted, i.e., the laser is fixed in position $(0, 0)$ of the 2D plane where a polar mapping with a radius of 5.6 m and angle comprised of from -30 degree to 210 degree. Three regions are highlighted (blue, red and yellow). The blue region represents the area with a precision of 30 mm of location, the red region is delimited to the work area where the academic community normally realizes models and characterization with the Hokuyo [24], [73], [75] . The area in yellow includes the region of measuring stipulated by the manufacturer [74]. Within the measurable area, LRF locates a circle with a diameter of 25 cm and returns the position (x, y) of its center point. Note that at the top of the real robot there is an object with a diameter of 25 cm (Figure 3.5), so LRF returns the position (x, y) of the robot.

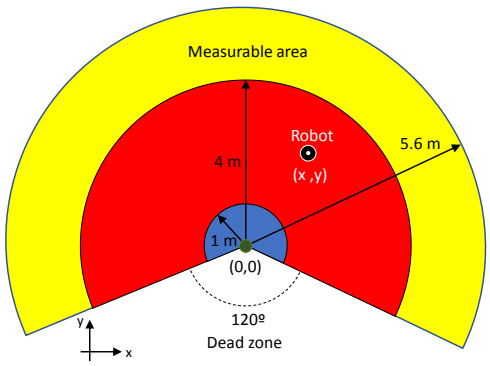


Figure 3.21: Laser operation and location of the robot in the ground truth approach.

It is worth mentioning that the operating principle of the Hokuyo URG-04LX varies according to the measuring angles, incidence of light, material that makes up the object and also the distance of the object from the senors. Some works such as models and characterization of this sensor [73], [75], showing imprecision in the order of mm, being within the resolution required for this project, validating the ground truth system adopted.

Chapter 4

Localization, Trajectory Planning and Control

This chapter will present the tools used and the implementation method for a mobile robot to be able to sweep an indoor environment. Thus, three major research areas of mobile robotics will be aborted, being: localization, path planning and mobile robot control.

4.1 Localization

It is a fact that the systems and sensors for acquiring position have intrinsic noises, and consequently, it produces errors in the measurement of the robot position. In this section will use the concept of probabilistic estimation, fundamental to the problem of estimating the location and position of an WMR in indoor environments. Even with sensors conditioning the uncertainty in their data, it is possible to obtain an estimation of the localization of the robot justified [20].

This section is dedicated to present the mathematics tool used to reduce the uncertainty in the location of the robot, where is used for this the fusion of the odometry with the data obtained from the UWB sensor.

4.1.1 Sensory Fusion (Extended Kalman Filter)

Kalman filter has good results for linear systems. Its use is limited to the field of mobile robotics, where it has state transitions or equations described by non-linear functions. However, to limit the nonlinearity of the systems, the EKF was developed. The EKF is the version of the Kalman Filter that overcomes the non-linearity of some system dynamics. It proceeds by continually updating a linearization around the previous state estimation and by approximating the state densities by Gaussian densities. [69].

EKF implements the sensor data fusion as show Figure 4.1. The median based gate filters the position provided by the Pozyx system ($x'_p and y'_p$) due to high intrinsic noise of UWB. The EKF implements the sensor fusion task. It receives x_p and y_p (position) from the median based gate and V_l and V_r (left and right wheel speeds) from the odometry system. The output of the filter is composed by x, y, θ that is the robot position and orientation [67].

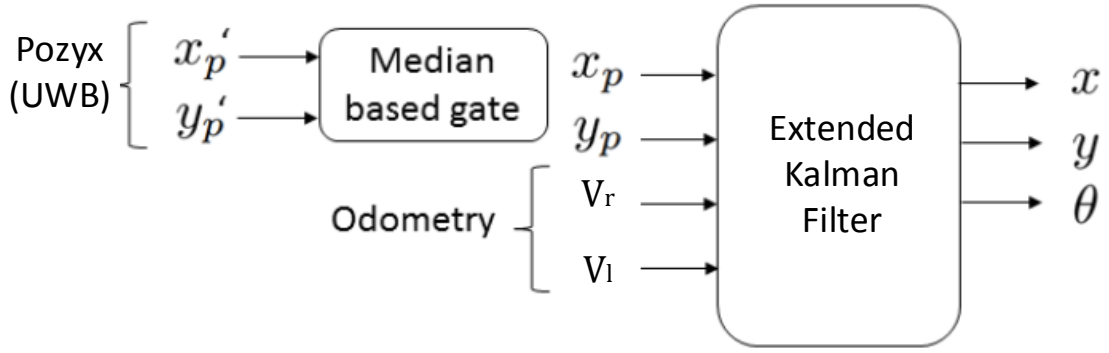


Figure 4.1: Data fusion using information of Pozyx sensor and odometry (adapted from [67]).

The filtered values (x_p and y_p values to set the center of the gate that eliminates unlikely measures) are the input for EKF.

$$X_k = [x, y, \theta]^T \tag{4.1}$$

$$\dot{X} = f(X, u) \tag{4.2}$$

where the variable u is:

$$u = [Vw]^T \tag{4.3}$$

The observation estimate, Z:

$$Z = h(X) \tag{4.4}$$

F_k is the state transition model which is applied to the previous state X_{k-1}, expressed in equation 4.5 where Δt is the time step for the acquisition.

$$F_k = \begin{bmatrix} 1 & 0 & -\Delta t \cdot sin(\theta) \\ 0 & 1 & \Delta t \cdot cos(\theta) \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

The Jacobian matrix is:

$$H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.6}$$

The measurement from Pozyx after median filter procedure is:

$$Z_k = \begin{bmatrix} x_p \\ y_p \end{bmatrix} \tag{4.7}$$

Time Update or Predict

The predict state estimate the actual pose of robot by the transition model until k instant:

$$P_k^- = F_{k-1} \cdot P_{k-1} \cdot {F_{k-1}}' + Q_{k-1} \tag{4.8}$$

Where F_k is the state transition model which is applied to the previous state x_{k-1} and Q_{k-1} is the process noise covariance. The Q_{k-1} matrix can be written as equation below:

$$Q_{k-1} = \begin{bmatrix} cov(v_x, v_x) & cov(v_x, v_y) & cov(v_x, w) \\ cov(v_x, v_y) & cov(v_y, v_y) & cov(v_y, w) \\ cov(v_x, w) & cov(v_y, w) & cov(w, w) \end{bmatrix} \tag{4.9}$$

Instead of working with v_x and vy, it is possible to perform a rotation R to work with v and v_n (where v is the linear velocity whereas v_n is the normal velocity) it can assume $v_n = 0$ and $cov(v_n, v_n) = \sigma_n^2$.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

Working with v and v_n, $Q_{k-1} = R \cdot Q - k - 1' \cdot R^T$ is presented in equation 4.11, it is possible to reach the ratio between $cov(v, v)$ and $cov(w, w)$ presented in equation 4.14 assuming that V_l and V_r errors follow a normal distribution centered in zero with a standard deviation of σ_o. Equation 4.13 and 4.12 present the distribution of v and w

$$Q_{k-1} = \begin{bmatrix} cov(v, v) & 0 & 0 \\ 0 & cov(v_n, v_n) & 0 \\ 0 & 0 & cov(w, w) \end{bmatrix} \tag{4.11}$$

$$v \to 2 \cdot \frac{1}{4} \cdot N(0, \sigma_o^2) \tag{4.12}$$

$$w \to 2 \cdot \frac{1}{b^2} \cdot N(0, \sigma_o^2) \tag{4.13}$$

$$cov(v, v) = \frac{4}{b^2} \cdot cov(w, w) \tag{4.14}$$

This methodology allows to tune the EKF by two constants (σ_o, σ_n) that can be found by performing a few experiences. The prediction state estimate, X_k^- is estimated by:

$$X_k^- = f^*(X_{k-1}, u_{k-1}) \tag{4.15}$$

Measurement Update

This phase correct the predicted state inputting the measurement of Pozyx data after the median filter procedure. The measurement residual is obtained by:

$$\widetilde{Y}_K = Z_K - Z \tag{4.16}$$

The innovation covariance:

$$S_k = H_k \cdot P_k^- \cdot H_k^{'} + Ro_k \tag{4.17}$$

where Ro_k is the observation noise covariance. The Kalman gain K_k :

$$K_k = P_k^- \cdot H_k^{'} \cdot S_k^{-1} \tag{4.18}$$

The update state estimate,X_k:

$$X_k = X_k^- + K_k \cdot \widetilde{Y}_K \tag{4.19}$$

The updated covariance estimate, P_k can be described:

$$P_k = (I - K_k \cdot H_k) \cdot P_k^- \tag{4.20}$$

4.2 Trajectory Planning

The concept of path planning was previously presented in section 2.4, where the main approaches used to identify a trajectory for the mobile robot were presented. In this section it will be detailed the formulation of the path planning problem for the robot to perform scan in an environment with toxic substances. The model used to represent the space where the scan will be done is described. Two methods to problem solving will be addressed. The first is a heuristic procedure developed through path planning analysis and the second is a genetic algorithm adapted to find an optimized solution.

4.2.1 Problem Formulation

Many approaches in path planning, seek only to make the shortest or most efficient path between two distinct points (start and target). However our problem has a different framework, similar to the classic traveling salesman's problem which was presented in section 2.5. By analogy the traveling salesman is the robot and the cities are the waypoints. The differences between our problem and the TSP is that the starting and ending points are different, i.e, the robot should not end the route at the same point it was started.

The challenge of path planning for robots is usually formulated as follows: given a mobile robot and a description of an environment need to plan a path between two specified locations, the start and the end point. During the execution of the path the robot can not collide with obstacles and the optimization criterion must be satisfied (for this case the shortest path) [76].

To simplify the path planning problem, it is necessary to make some assumptions. They are as follows:

- A path will be selected, always starting from a start point to a target point, as show in Figure 4.2.

- Known obstacles are mapped and represented in the connectivity grid.

- The proposed algorithm acts on a connectivity grid arranged in a two dimensional space (2D) or $\Re^2$ space.

- The robot does not perform movements in diagonal directions. It only moves between interconnected points in horizontal and vertical directions in the connectivity grid as show in Figure 4.2.

- The robot should visit all cells or points that are free of obstacles in the connectivity grid at least once.

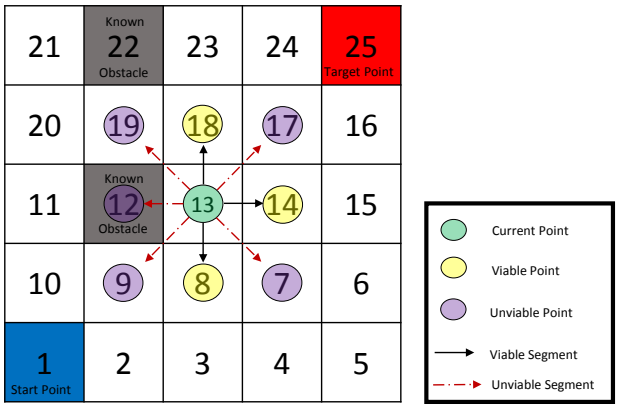


Figure 4.2: Connectivity grid (size: 5x5) and an example of possible points and segments. Without an obstacle the 12 point will be a reachable point.

4.2.2 Problem Space Representation

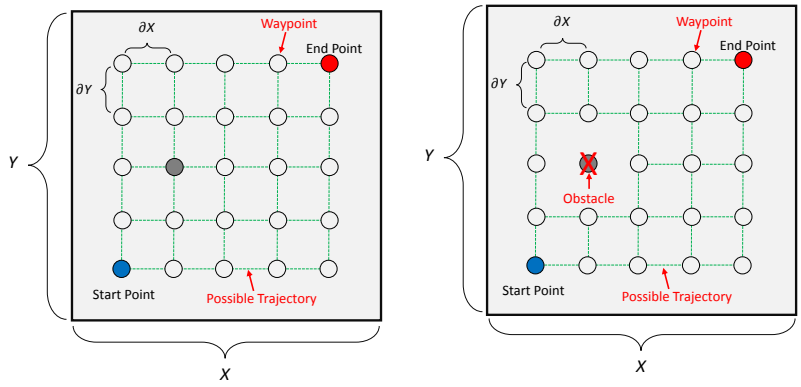
Many works developed in the area of path planning for mobile robots, use a graph of connectivity grid to represent the environment and obstacles. In the present work we use a similar approach to that presented in the papers [48], [77], where it modify the order of the values as shown in Figure 4.2. The dark color represent obstacles, while the lighter colors represent obstacle-free cells.

For a better illustration of the space representation and its possible trajectories the Figure 4.3a shows an environment limited by a square of X meters wide by Y meters long. The green lines represent the connections (possible paths) between different waypoints with a dimension $(\delta x, \delta y)$ meters. The previously mapped obstacles are indicated by the dark waypoints, which become unviable and isolated, i.e., without access as shown in Figure 4.3b.

4.2.3 Heuristic Method to Path Planning

This subsection will be dedicated to clarifying the heuristic strategy developed to carry out the path planning of the robot, aiming to be a simple and fast method, which is calculated before the robot start its movements through the environment.

The approach uses the Von Neumann neighborhood which consists of four orthogonal



(a) All possible trajectories.

(b) Isolated waypoint due to obstacle.

Figure 4.3: Representation of the connectivity grid.

cells around a central cell in a two-dimensional grid. This setting is shown in Figure 4.4. The center cell indicates the current waypoint or position of the path planning. The neighboring positions indicate the possible future positions to be selected to compose the path. The algorithm starts the path planning with the central cell of Von Neumann's

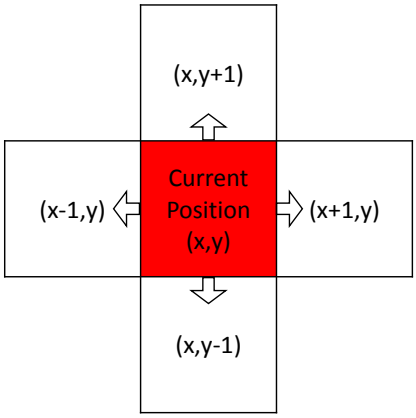


Figure 4.4: Von Neumann neighborhood with unitary radius.

geometry on the starting position of the robot (see Figure 4.2). Eight different direction priorities were set to guide the selection of the next cell for path planning. The path planning is calculated for all priorities, one at a time. The path generate by direction that has the smallest number of iterations and consequently the smallest trajectory will be selected to be executed by the robot. Figure 4.5 presents the eight priorities of direction

along with the logic of the heuristic method.

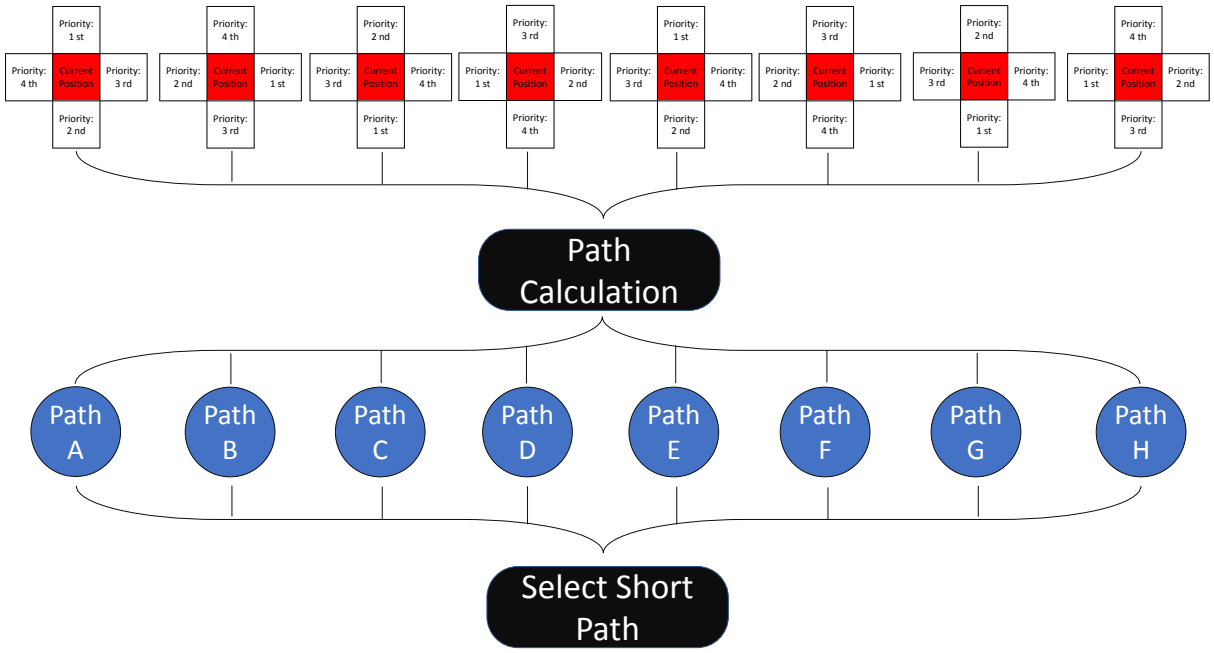


Figure 4.5: Heuristic method using eight direction priorities to generate eight paths and select the smallest.

The algorithm verifies which of the neighbors in the current position of the neighbor Von Neumann's has less number of visits. If only one neighbor was less visited, the algorithm selects this direction independent of the direction priority. When the smallest number of visits is verified in more than one neighbor, the choice will be according to the current position priority. Changing from one priority to another occurs when the algorithm identifies the final waypoint and all other waypoints have already been visited at least once. When the eight priorities are analyzed, the smallest path generated by the heuristic method will be selected. The flowchart describing the heuristic method can be verified in appendix C.

This methodology can be applied to dynamic trajectory planning, where unknown obstacles are encountered during the course, as developed in paper [78] during this work.

4.2.4 Genetic Algorithm to Path Planning

This section will present the genetic algorithm used to solve the path planning problem described in the subsection 4.2.1. To facilitate understanding, when referring to a gene, we are indicating a cell in the grid of connectivity. When a chromosome is pronounced, it indicates a set of cells that connect the start point to the end point.

Encoding Representation

The encoding method is one of the key steps in the design of a GA. The representation of the possible paths to be realized by the robot, is known as chromosome [48], [77]. The path is encoded in a sequence of adjacent cells. This sequence is started with the start cell (bottom left corner) and ended with the destination (upper right corner) cell. The path consists of a variable number of segments formed between two cells (waypoints). Each segment is a straight line which can be vertical or horizontal. Diagonal segments are invalid. Figure 4.6 shows a possible chromosome generated from the connectivity grid of Figure 4.2.

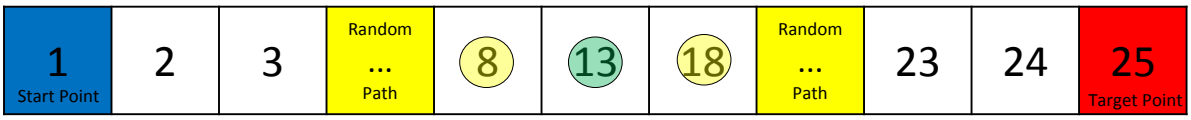


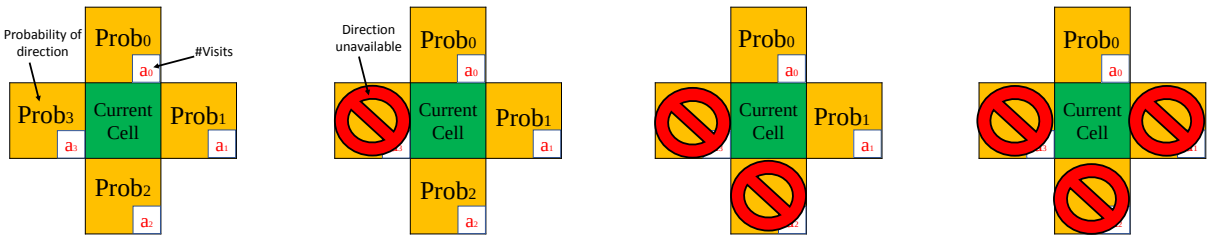
Figure 4.6: Possible chromosome that constituted the initial population of connectivity grid.

Initial Population

The initial population is generated in order to respect the criteria of horizontal and vertical movement allowed. The initial population is subjected to a random process, always starting at the start point and ending at the target point (see Figure 4.2 and 4.6). However all the initial randomly generated populations should visit all the points of the connectivity grid of at least once.

With the intention of improving the research capacity in addition to reducing the search time of the evolutionary algorithm, all the chromosomes generated by the initial population represent an executable path, as in the papers [48], [77], [79].

To generate a chromosome of the initial population, the algorithm apply a mask with Von Neumann Geometry (See Figure 4.4), where the center represents the current point and the extremity of the cross represent the directions allowed for the path. The choice of the direction taken is random and each direction has a probability of choice according to the amount of visits already undertaken. In other words, a point that was less visited is more likely to be visited compared to its neighbors that were most visited. In this way, all points have a probability of being chosen, guaranteeing a great diversity for the initial population. The mask can be seen in Figure 4.7a, where the possible configurations are also shown depending on the availability of neighboring points (unknown obstacles) or in case the mask is centered at the edge of grid.



(a) All directions available.

(b) Three directions available.

(c) Two directions available.

(d) One Directions Available.

Figure 4.7: Geometry of the mask for the generation of the initial population.

The probability of each direction $Prob_i$ for $i = 0, ..., 3$ is a function of the visit cell number (a_i) and n represents the available directions obtained by:

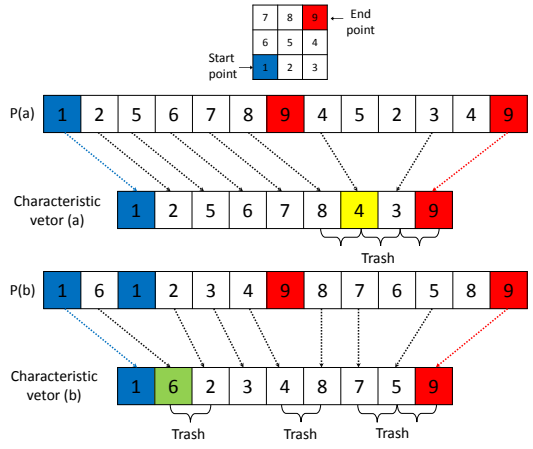
$$Prob_i = \begin{cases} \frac{\sum_{k=0}^{n-1} a_k - a_i}{(n-1)\cdot\sum_{k=0}^{n-1} a_k}, & for \quad n > 1 \\ 1, & for \quad n = 1 \end{cases} \tag{4.21}$$

Crossover Operation

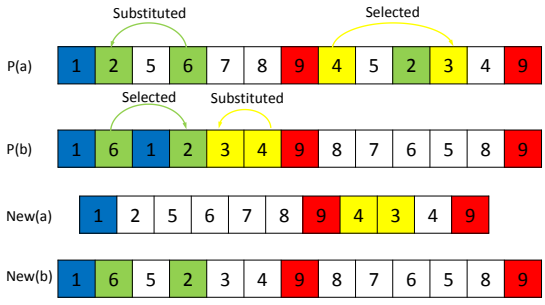
According to [80] crossover can be defined as: "Crossover is the process of taking two parent solutions and producing from them a child. After the selection (reproduction) process, the population is enriched with better individuals. Reproduction makes clones of good strings but does not create new ones. Crossover operator is applied to the mating pool with the hope that it creates a better offspring."

In the developed algorithm, the crossover consider two parents, selected randomly, to produce two children. The first step of this process is to generate the characteristic path of these two selected parents. The characteristic path is a sequence of cells that are visited for the first time in the path. In this way, individual and distinct information is stored for each parent randomly generated. Each characteristic path must begin in the starting point and end in the endpoint (even if the endpoint was previously visited).

Between two consecutive cells of the characteristic path there may be several cells in the parent chromosome, which were already visited. Figure 4.8a illustrate a situation where two characteristic paths are generated for two parents, where a 3X3 grid was used to facilitate the understanding of this operation.



(a) Characteristic path.



(b) Generation of new chromosomes.

Figure 4.8: Crossover operation details.

Only the garbage ranges can be used to generate the new chromosomes (offsprings). Therefore a random point of the characteristic path is selected (in yellow in (a) and green

in (b)), where between it and its subsequent point there is garbage to be replaced by the smaller interval in the opposite parents that generated the characteristic path. Figure 4.8b shows in detail the process of generating the new chromosomes.

Mutation Operation

According to [80] "Mutation prevents the algorithm from being trapped in a local minimum. Mutation is viewed as a background operator to maintain genetic diversity in the population. It introduces new genetic structures in the population by randomly modifying some of its building blocks. Mutation helps escape from local trap and maintains diversity in the population". All chromosomes are candidates to submitted the mutation process with a probability of $Prob_m$.

In order to prevent a mutation parent producing a unfeasible path, the developed algorithm for the mutation operator was established to avoid all such cases, i.e, all the paths generated after the mutation are feasible. As in the crossover operator, the first procedure performed is to generate a characteristic path. After this, a cell of the characteristic path is randomly chosen and its subsequent one where a random path between them will be inserted. The generated random path allows only horizontal and vertical movements. Figure 4.9 illustrates the operation for a 3X3 size connectivity grid.

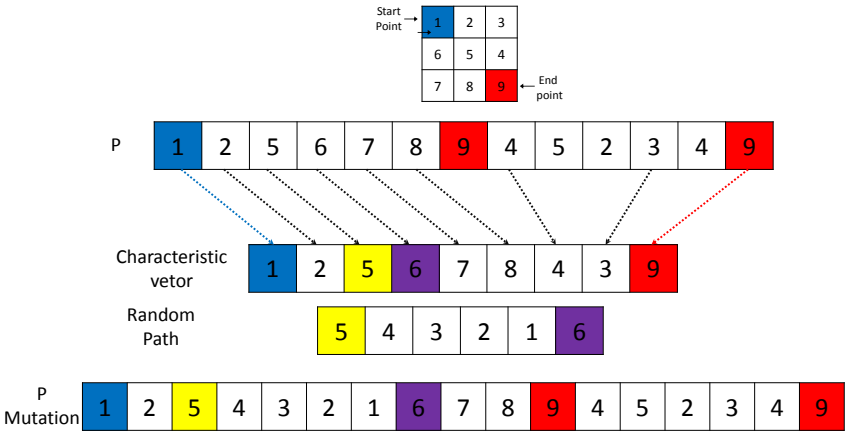


Figure 4.9: Detail of mutation operator.

4.2.5 Selection Process

In this work the goal is to obtain an optimal path, that means, the path with the shortest distance between the starting point and the end point by visiting all points of the connectivity grid. In order to evaluate the chromosomes, the amount of cells is analyzed. The best path is the one with less cells in its chromosome. When one of the chromosomes has $(i \times j)$ cells, i.e. when all points are visited only once to accomplish all the visits, the path is fully optimized for our problem. To get the optimized path, it is necessary that the smallest paths are maintained and transferred to the next generations. A selection process is proposed to obtain the best parents and yet guarantee the diversity of the new populations. This process consists of ordering all the chromosomes obtained in the current iteration of the algorithm, considering the results of crossover and mutation operations, and classifying them in ascending order. After the ordering, the new population is selected where 10% of the individuals are the ones with the smallest paths and the remaining 90% are selected randomly from the ordered chromosomes, thus generating the new population.

4.3 Trajectory Design by Spline curves

To convert the path planing method into feasible trajectories for the robot its used spline interpolations in a 2D plane. Thus, it is possible to represent in a compact way the smooth path, sampled the curve in several segments. In his work Sprunk, define spline as: "a piecewise polynomial parametric curve in the two-dimensional plane $Q(t) \in R^2$. All segments of the spline are required to be polynomials of the same degree n" [81]. In this work the spline construct a smooth curve between two subsequent waypoins resulting from path planning.

In order for the robot to be able to follow the planned path, it is necessary that its movement restrictions be respected. In the case of the robot with differential drive, the restrictions are non-holonomic [34], [82], in other words, the robot does not perform lateral movement since the driven wheels rotate only forward and backward as already presented

in subsection 3.2.2 and modeled by the equation 3.7 rewritten below:

$$\dot{y(t)} \cdot cos(\theta(t)) = \dot{x(t)} \cdot sen(\theta(t))$$

Therefore, it is necessary that the path be continuous in a straight line and continuous in curvature [81]. In addition, by implementing a smooth curve, the vibrations in the robot are attenuated along the trajectory, decreases the wear of the actuators and the errors of tracking [83].

The implemented algorithm calculates a cubic spline for the mathematical representation of the trajectories between two waypoints (calculations and mathematical proofs can be found in chapter 18 section 6 of the book [84]). In the developed application, the trajectory based on spline curves has the task of constructing a continuous and smooth path feasible by robot. At the same time, this trajectory must respect the non-holonomic constraints as explained above. To smooth the curves, we control the input and output angles of the waypoints represented by equation 4.22.

$$\theta_n = atan2(y_{n+1} - y_{n-1}, x_{n+1} - x_{n-1}) \tag{4.22}$$

The spline curve between two distinct waypoints is sampled and inserting k_s equidistant points. The k_s is an integer value, assigned according to the distance D between waypoints. The greater the distance between waypoints the greater amount of sampled points can be entered. Figure 4.10 represents two spline curves generated between three different waypoints.

4.4 Robot Control

After the configuration of the space, set the resolution of the scan, calculate the path planning resulting in the route of the robot and interconnecting all waypoints through spline interpolation, the robot is able to navigate through path. To do so, it is necessary to perform methods of controlling the robot's achievers performing a path coherent with

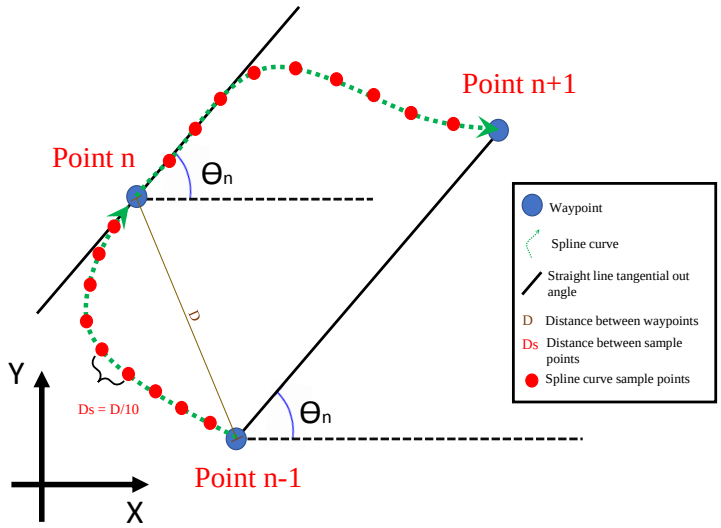


Figure 4.10: Spline curve generated between three distinct waypoints, sampled in smaller parts.

the planned. In this section will discuss two methods of controllers used in this work:

1. Points following: The goal of this controller is to make the robot go to a target point starting from its current position.

2. Segments following: Its purpose is to make the robot follow the trajectory sampled by the spline to the reference point.

According to [20] "In wheeled mobile robotics, usually kinematic models are sufficient to design locomotion strategie". Due to the small size of the simulated and real robot (see table 3.1) the control strategy does not involve the dynamics of the system. The proposed controllers were selected because they do not require dynamics considerations and do not require a high level of processing, which can be performed in simulation and real-time in real robot.

Both methods are based on feedback control. The real and simulated localization system is able to measure the variables involved in the control loop (position and orientation of the mobile robot with respect to global frame). The block diagram of Figure 4.11 shows the feedback control system. The feedback control methodology was adopted due to the open loop control not robust to modeling errors and that it cannot guarantee that the

mobile robot will move along the desired trajectory as planned [85]. Two variables are

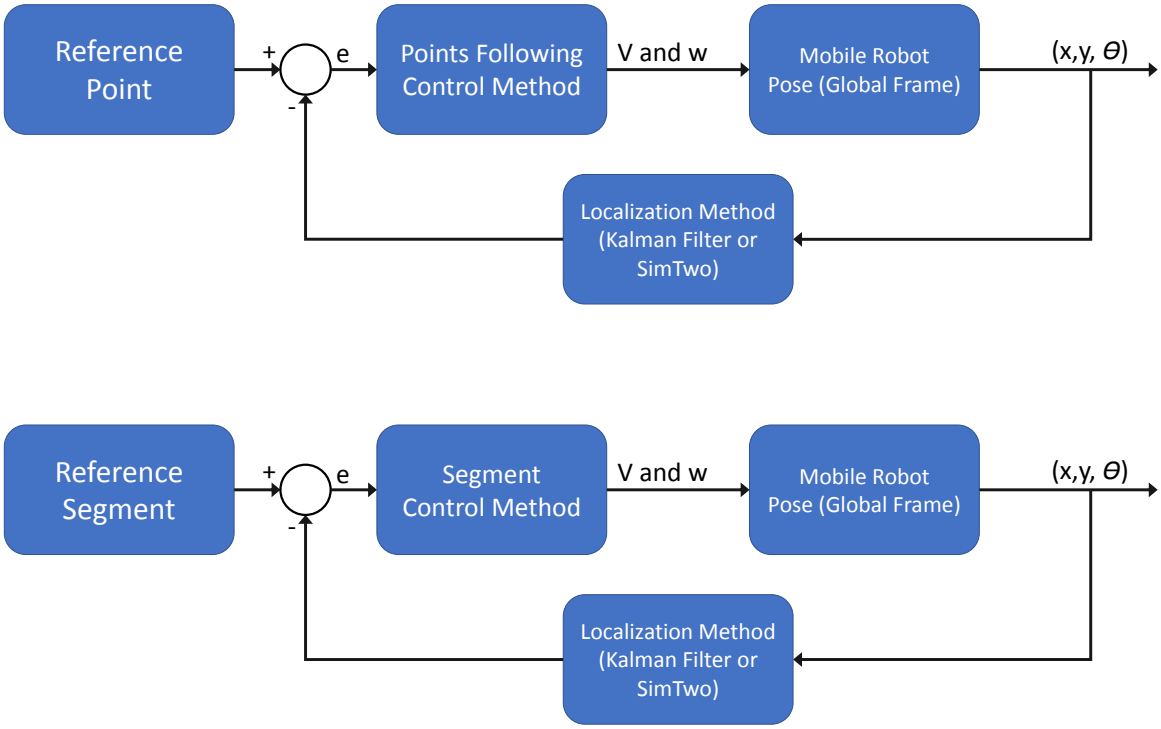


Figure 4.11: Feedback control system diagram: above points following and below segment following.

susceptible to be controlled: V (linear velocity) and W (angular velocity). V is always in the X_r direction. To respect the non-holonimic restriction the robot can not have velocity in the Y_r direction. w can be considered as the velocity of rotation of the ξ_r in relation to ξ_g. To recall the structure of the reference frames and the kinematics of the differential robot, take the subsection 3.2.2. The methods of control implemented will be described below.

4.4.1 Points Following

The following point controller is able to direct the WMR to a discrete position, that is, a point in the 2D space of the environment controlling the linear velocity V and the angular velocity w. The proposed path planning approaches organize the order of all waypoints that must be visited. The spline iterations inserts k_s points between each waypoint by

constructing a smooth path. Thus there is a set of P ordered points that form the robot's trajectory, where P is an integer representing the total number of points the robot should visit.

The idea of this controller is to establish a line λ_n between the position of the robot (x, y) and the position of the next point of the trajectory (x_n, y_n) where $n = \{1, 2, ..., P\}$. Then is calculate the difference θ_e between the orientation of the line λ_n and the orientation of the robot obtained by the equation 4.24 and as can be seen in Figure 4.12.

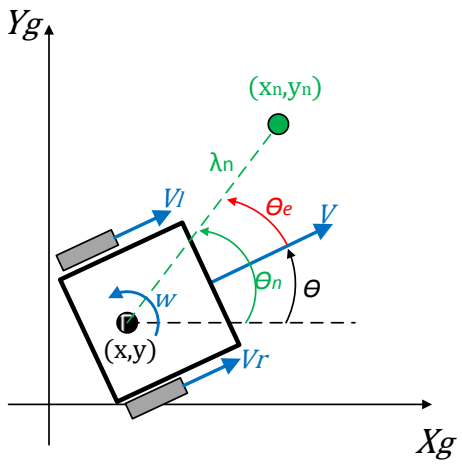


Figure 4.12: Considerations for robot control following points.

$$\theta_n = \arctan\left(\frac{x_n - x}{y_n - y}\right) \tag{4.23}$$

$$\theta_e = \theta_n - \theta \tag{4.24}$$

Starting from the equations 3.5 and 3.6 described above, the robot will be controlled to cancel out the value of θ_e. The value of angular velocity w must follow a reference w_{ref} in function of a constant of proportionality K_{prop} and θ_e represented by equation 4.25. The linear velocity V must be constant V_{ref} except when it is close to the last point of the trajectory (x_p, y_p).

$$V = \frac{V_l + V_r}{2} \qquad w = \frac{V_l - V_r}{b}$$

$$w_{ref} = K_{prop} \cdot \theta_e \tag{4.25}$$

In this way the velocity of the tractioned wheels are controlled as it presents the equations 4.26 and 4.27.

$$V_l = V_{ref} - \frac{w_{ref} \cdot b}{2} \tag{4.26}$$

$$V_r = V_{ref} + \frac{w_{ref} \cdot b}{2} \tag{4.27}$$

When approaching the last point of trajectory, more specifically at a distance $dist_{end}$ of the least point, the robot has a deceleration in the linear velocity V whose goal is to make the robot (x, y) stop near the end point (x_p, yp). Figure 4.13 shows V_{ref} as a function of the robot's proximity to the last trajectory point.

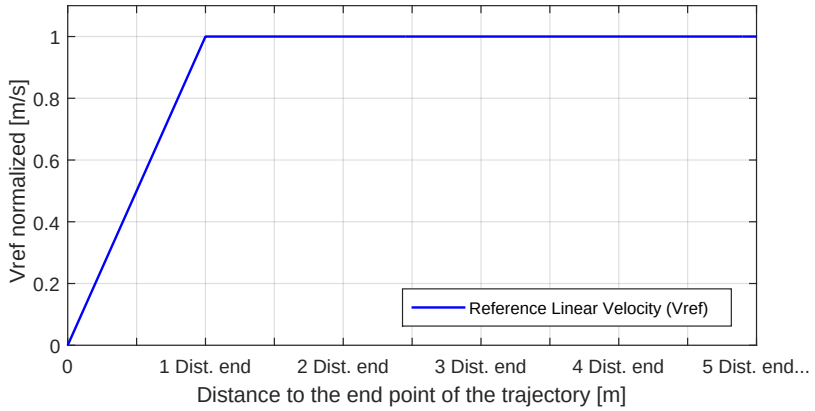


Figure 4.13: Reference velocity as a function of robot distance to the last trajectory point.

4.4.2 Segments following

Given a trajectory provided by the spline interpolation, this controller seeks to stabilize the robot on a sample of this trajectory minimizing the posture error. In other words, the

strategy proposed by this controller is to make the robot follow the sampled segments of the spline curve, illustrated in Figure 4.14 .

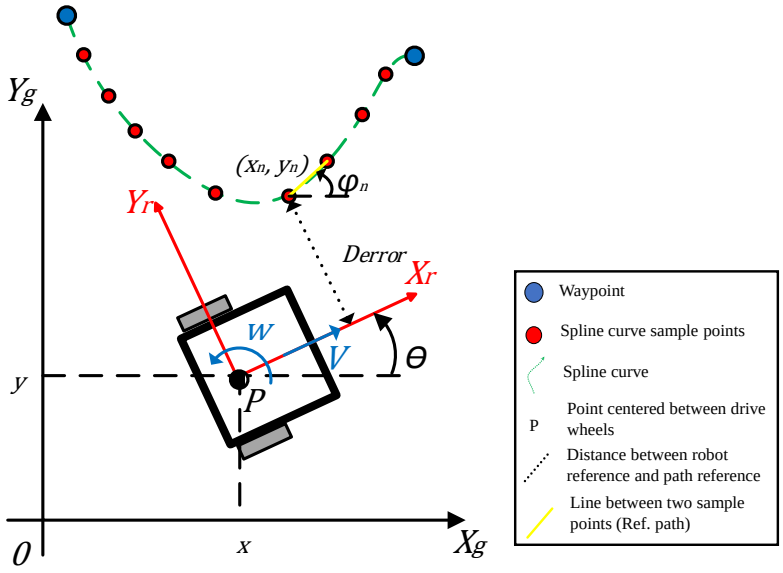


Figure 4.14: Considerations for robot control segments following.

A nonlinear kinematic model was used to represent the WMR movements in two frames, the global (X_g, Y_g) and the robot frame (X_r, Y_r), as show in Figure 4.14. The equations below demonstrate the procedure for controlling V and w where the *Speed* variable denotes the maximum linear velocity of the robot. When $\theta_{error} = 0$ and $D_{error} = 0$ the robot is aligned with the segment. In equation 4.29 is calculated the error between the orientation of the segment and the orientation of the robot. The controller tries to nullify this error.

$$\varphi_n = \arctan\left(\frac{x_{n+1} - x_n}{y_{n+1} - y_n}\right) \quad n = 1, 2, ..., P-1 \tag{4.28}$$

$$\theta_{error} = \theta - \varphi_n \tag{4.29}$$

In equation 4.30 the orthogonal shift of the robot with respect to the segment is calculated.

$$D_{error} = -(x - x_n) * sin(\varphi_n) + (y - y_n) * cos(\varphi_n) \tag{4.30}$$

The deviation of the robot in relation to the reference trajectory is obtained by θ_{error} and D_{error}. In equation 4.31 the robot's angular velocity is calculated to correct the trajectory error. W is in function of θ_{error} and D_{error} and their respective gains k_θ and k_{error} are constants and positive.

$$W = \begin{cases} -k_\theta \cdot \theta_{error} - k_{error} \cdot D_{error}, & for \quad Speed \geq 0 \\ -k_\theta \cdot \theta_{error} + k_{error} \cdot D_{error}, & for \quad Speed < 0 \end{cases} \tag{4.31}$$

The angular velocity is limited by:

$$W = \begin{cases} \pi/2, & for \quad W \geq \pi/2 \\ -\pi/2, & for \quad W \leq \pi/2 \end{cases} \tag{4.32}$$

In equation 4.33 the linear velocity is calculated. *Speed* indicates the maximum linear velocity the robot can achieve. V is in function of W and consequently of the trajectory deviation of the robot.

$$V = \begin{cases} Speed \cdot cos(W), & for \quad |W| < 4\pi/9 \\ Speed \cdot 0.2, & for \quad |W| \geq 4\pi/9 \end{cases} \tag{4.33}$$

The Figure 4.15 shows the relationship between the linear velocity and the angular velocity.

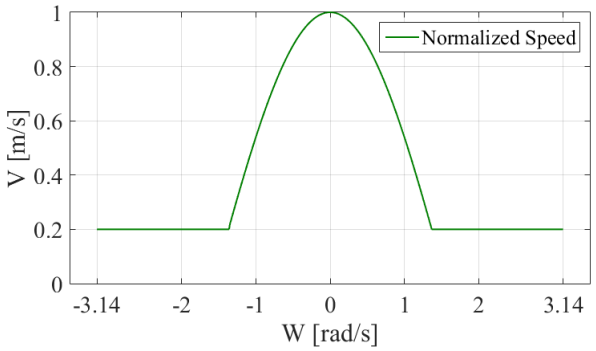


Figure 4.15: Linear velocity as a function of angular velocity.

Chapter 5

Results

This section will be dedicated to the presentation of results regarding the planning, control and localization activities using the methodologies proposed and presented in the previous chapters. In order to do so, experiments were carried out in SimTwo simulation environment, as well as practical experiments with the real robot. Initially, path planning will be analyzed, comparing the heuristic method with the GA method. So an approach to re-planning on the fly in simulation is presented. Further, the two approaches of control of the robot in a simulation environment will be analyzed. Then, finishing the work will be analyzed the localization of the robot in a real environment where was used different controls proposed for the robot to perform the scan of the environment. For all tests performed in this chapter static obstacle (known and unknown) will be considered.

5.1 Path Planning

This section will present the numerical results obtained for the planning algorithms, more specifically the paths generated from the heuristic method and from the proposed GA. The ideal path is one that begins from the starting point to the end and visits all the cells available once. Two cases will be presented. The first one case was performed with a connectivity grid with a resolution of 5X5. The second one case uses an 8x8 connectivity grid and will insert some known obstacles raising the complexity of path planning. The

table 5.1 show the data used in GA for all cases studied in this section.

Table 5.1: Data used in GA to path planning

GA data	Description	Value
Population Size	Number of chromosomes	50
Number of iterations	Maximum quantity iterations performed	100
Probability crossover	Probability to perform crossover	0.95
Probability mutation	Probability to perform mutation	0.2

5.1.1 Connectivity Grid with Resolution 5X5

This case presents a connectivity grid to represent the environment to be scanned with 25 cells, all available as shown in Figure 5.1. This same representation will be used for the real environment.

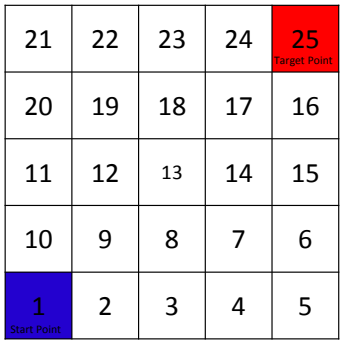


Figure 5.1: Connectivity Grid with 25 cell.

Figure 5.2a shows the sequence of cells visited using the heuristic method where each cell is visited only once. On the other hand, Figure 5.2b shows the sequence of cells visited using the GA method where points 8 and 7 were visited twice.

The table 5.2 shows the time required for the algorithms to converge for a path.

Table 5.2: Path planning time for a 5X5 connectivity grid

Method	Time to plan	unit
Heuristic	0.106	s
GA	125	s

1	10	11	20	21	22	19	12	9	2	3	8	13	18	23	24	17	14	7	4	5	6	15	16	25

(a) Result to Heuristic method

1	10	11	20	21	22	19	12	9	2	3	8	7	14	15	6	5	4	7	8	13	18	23	24	17	16	25

(b) Result to GA method

Figure 5.2: Path obtained for 5X5 connectivity grid.

It is possible to observe that in this case the heuristic method presents an efficient and faster behavior when compared with the evolutionary algorithm. Figure 5.3 shows the GA evolution and compares it the heuristic and ideal path.

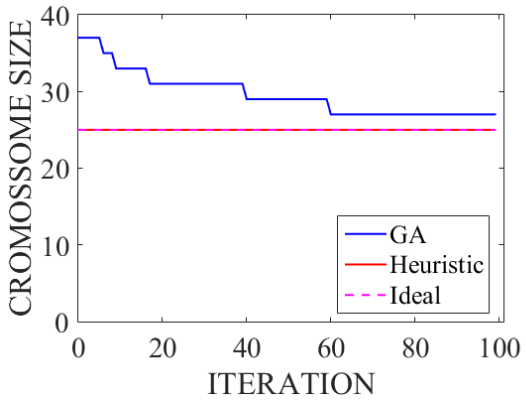


Figure 5.3: Performance of planning algorithms with a 5X5 connectivity grid.

The heuristic method has good behavior for cases where there are no obstacles and the path is trivial. The table 5.3 presents the number of visited cells.

Table 5.3: Number of waypoints visited by path planning algorithms considering a 5X5 connectivity grid.

Case Connectivity Grid 5X5	# Waypoints Visited
Ideal	25
Heuristic	25
Genetic Algorithm	27

5.1.2 Connectivity Grid with Resolution 8X8

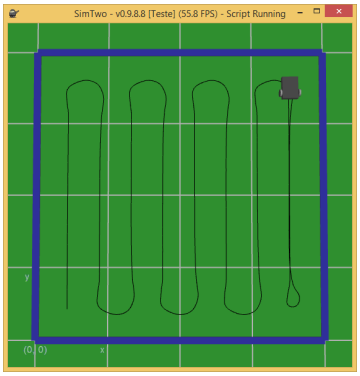
In this subsection it will be tested three different cases (A,B,C) with an 8x8 connectivity grid. The Situation A, where there are no obstacles and the optimal path is trivial and Situations B and C where there are some known obstacles and it is not obvious the optimal path. The solution to these latter cases is more complex than it appears.

Case A: No Obstacle

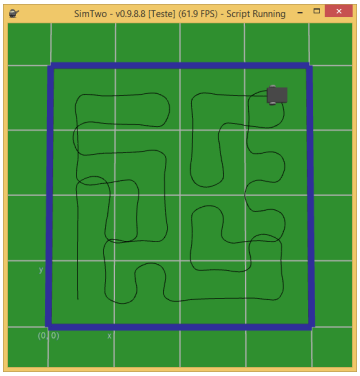
In this case all the cells are available and need to be visited. Figure 5.4a shows the connectivity grid. Figure 5.4b presents the path realized by the heuristic method and the Figure 5.4c presents the path accomplished by the GA.

64	63	62	61	60	59	58	57 Target Point
49	50	51	52	53	54	55	56
48	47	46	45	44	43	42	41
33	34	35	36	37	38	39	40
32	31	30	29	28	27	26	25
17	18	19	20	21	22	23	24
16	15	14	13	12	11	10	9
1 Start Point	2	3	4	5	6	7	8

(a) Conectivity grid



(b) Path by heuristic method



(c) Path by GA method

Figure 5.4: Case A: Connectivity grid 8x8 without obstacle.

It is a trivial case, however the two approaches did not find the optimal path. Therefore, considering the previous subsection 5.1.1 it is possible to conclude that the heuristic method only finds the best path for trivial cases with the presence of odd rows or columns. Figure 5.5 shows the performance of the GA during its iterations by comparing it with the heuristic method.

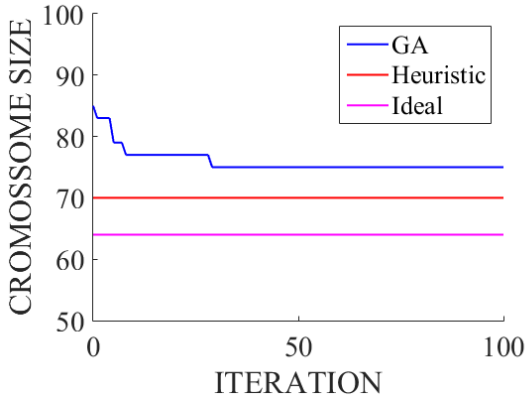


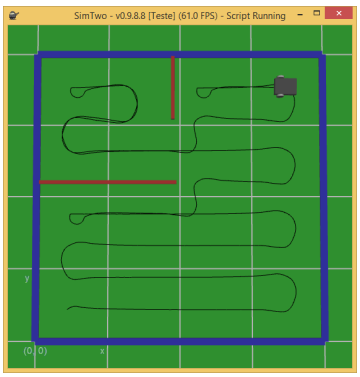
Figure 5.5: Case A: Performance of algorithms to path planning.

Case B: Two Obstacle

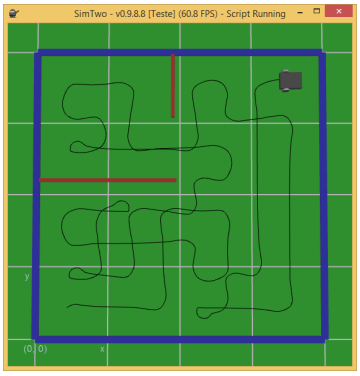
Figure 5.6a presents the case B where connectivity grid has two known obstacles. In this situation it becomes a complex task to carry out route planning. Wayopint 45 represents a unique path both for entering and leaving the sub-environment. The result with the heuristic method can be analyzed in figure 5.6b while the GA result can be analyzed in Figure 5.6c.

64	63	62	61	60	59	58	57 Target Point
49	50	51	52	53	54	55	56
48	47	46	45	44	43	42	41
33	34	35	36	37	38	39	40
32	31	30	29	28	27	26	25
17	18	19	20	21	22	23	24
16	15	14	13	12	11	10	9
1 Start Point	2	3	4	5	6	7	8

(a) Connectivity grid



(b) Path by heuristic method



(c) Path by GA method

Figure 5.6: Case B: 8x8 connectivity grid with two obstacle.

The performance of the GA is presented in Figure 5.7 where is possible to observe that, after some iterations, the Genetic Algorithm is capable to identify a better path when compared with the heuristic planning. Note that in this situation, once the ideal

path value is not trivial, the least possible was adopted, being attributed to this the sum of the number of available cells.

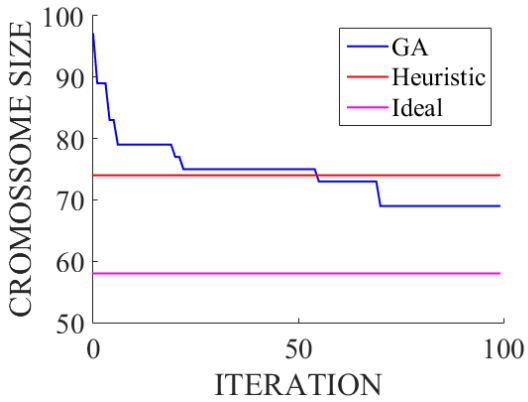


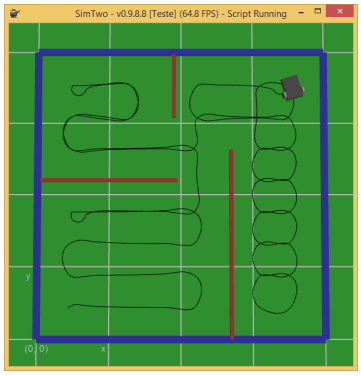
Figure 5.7: Case B: Performance of algorithms to path planning.

Case C: Three Obstacles

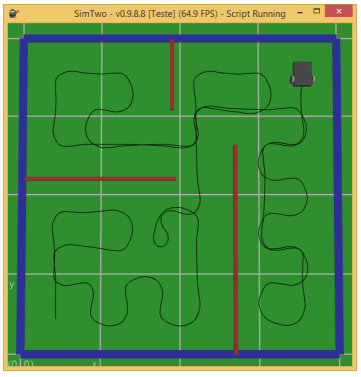
Figure 5.8a presents the Situation C where the connectivity grid has three known obstacles forming a complex scenario. The Waypoints 44, 45, 46 can be considered critical, since they will be visited more than once to satisfy the proposed problem. The result with the heuristic method can be analyzed in figure 5.8b while the GA result can be analyzed in Figure 5.8c.

64	63	62	61	60	59	58	57 Target Point
49	50	51	52	53	54	55	56
48	47	46	45	44	43	42	41
33	34	35	36	37	38	39	40
32	31	30	29	28	27	26	25
17	18	19	20	21	22	23	24
16	15	14	13	12	11	10	9
1 Start Point	2	3	4	5	6	7	8

(a) Conectivity grid.



(b) Path by heuristic method.



(c) Path by GA method.

Figure 5.8: Case C: 8x8 connectivity grid with three obstacle.

The performance of the GA is presented in Figure 5.9 where it is possible to observe that evolutionary algorithm obtains better solutions than the heuristic planning procedure starting from the initial iterations. Note: as in case B, in this situation as the ideal path value is not trivial, the procedure to evaluate the ideal path is the same.

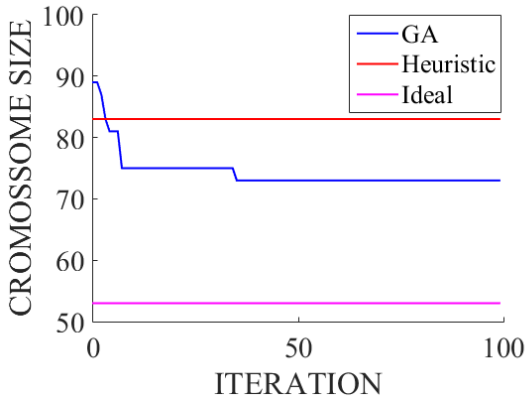


Figure 5.9: Case C: Performance of algorithms to path planning.

Case Analysis

The table 5.4 presents the number of visited cells. So, analyzing the genetic algorithm behavior in the Situations A, B and C it is possible to conclude that the GA method has better results in a more complex environment.

Table 5.4: Heuristic and GA method comparison (visited cells).

Case	Ideal	Heuristic	Genetic Algorithm
A	64	70	75
B	58	75	71
C	52	83	72

However, the time spent to perform the planning using the GA method is greater than the heuristic as shown in table 5.5. It is worth noting that the time spent to path planning in environments with known obstacles is not critical because it is performed before the mobile robot start the scan.

Table 5.5: Time consumed for the path planning algorithm to converge.

Case	Heuristic	Genetic Algorithm	Unit
A	0.127	190.23	s
B	0.182	315.98	s
C	0.193	362.54	s

5.2 Dynamic Path Planning

This section will be dedicated to presenting the re-planning of the trajectory when detecting an unknown obstacle in the environment. The implementation will be in simulation environment since the mobile robot is not equipped with a sensor capable of performing object detection in real time.

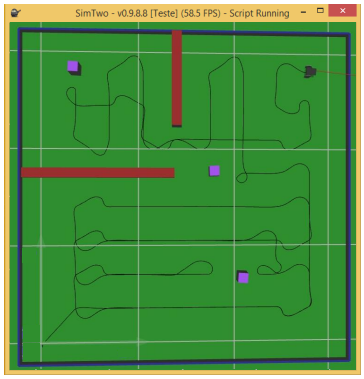
Due to the low time required for the developed heuristic algorithm to converge on a path where all waypoints are visited, it is possible to perform the re-planning of the robot's trajectory when an unknown obstacle is detected. As the genetic algorithm has a high time to converge, in this case it is impracticable for the re-planning of the route.

Figure 5.10a shows the connectivity grid where the gray cells are known obstacles whereas the purple cells are random unknown obstacles detected during the route. Figure 5.10b shows the path performed by the robot with in an environment with known and unknown obstacles. Although the robot does not collide with obstacles the resulting trajectory is not optimized. In future works it will be intended to optimize the algorithm for environments with unknown obstacles.

Whenever the laser ToF sensor (simulated) detects an obstacle, the robot will return to the last visited waypoint, then the waypoint that intercepts the obstacle will be deleted and the heuristic algorithm executed by tracing a new route. The petri net in the appendix D describes this approach.

64	63	62	61	60	59	58	57 Target Point
49	50	51	52	53	54	55	56
48	47	46	45	44	43	42	41
33	34	35	36	37	38	39	40
32	31	30	29	28	27	26	25
17	18	19	20	21	22	23	24
16	15	14	13	12	11	10	9
1 Start Point	2	3	4	5	6	7	8

(a) Connectivity grid



(b) Path by heuristic method

Figure 5.10: Path executed in environment with unknown obstacles.

5.3 Control

In this section it will be presented the results of the controllers proposed in this work. The obtained results will be presented considering the errors of measurement and uncertainties in the location of the robot in simulation environment as described in section 3.3. The dimensions of the test environment are describe in table 5.6 according to the representation of Figure 4.3a. The speed at which the robot travels through the environment must be reduced so that the measurement sensor is able to measure the radiation levels at each point to perform the scan.

Table 5.6: Dimension of the connectivity grid for controller tests in simulation.

Description of Connectivity Grid	Value	Unit
Size in X	4	m
Size in Y	4	m
Dx	0.67	m
Dy	0.67	m
# Waypoint	25	-

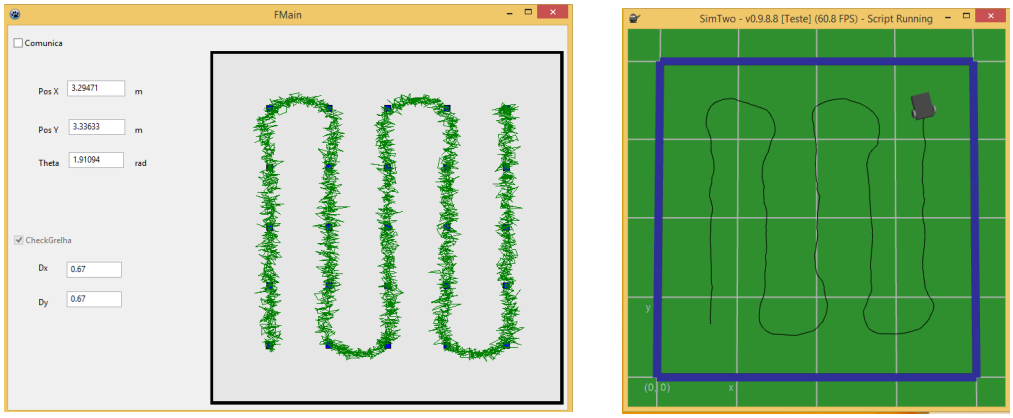
5.3.1 Waypoint Path

This subsection presents the simulation results obtained for the proposed controller in subsection 4.4.1. Table 5.7 presents the parameters of points following controller in simulation environment. The robot must have a reduced speed to be able to perform the scan, and the value of K_{prop} was obtained empirically by tests.

Table 5.7: Controller parameters waypoint path in simulation.

Controller Parameter	Value	Unit
V_{ref}	0.05	m/s
K_{prop}	30	-

Figure 5.11a shows the location data sent from the simulator to the ControllApp. The noises inserted in the data are in accordance with model represented in [67]. Figure 5.11b shows the path that the robot performed in the simulation environment.



(a) ControlApp. (b) Path taken by robot.

Figure 5.11: Control follow point with measurement noise and uncertainty of sensor Pozyx.

Figure 5.12 presents a comparison between the reference spline trajectory for the controller and the trajectory performed by the robot. The starting state of the robot is $[0.67, 0.67, \pi/2]^T$ and the state at the end of the path is $[3.3, 3.4, (11\pi)/18]^T$. When observing the trajectory of the simulated robot it is possible to notice that this controller did not obtain a performance of smooth the route.

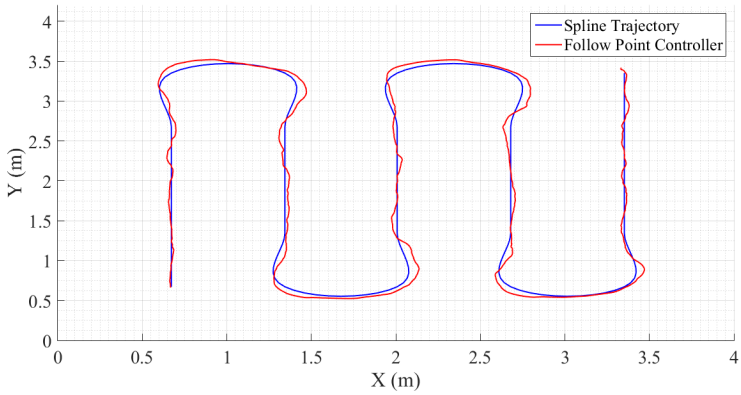


Figure 5.12: Comparison between spline trajectory and path realized by simulated robot with point following controller.

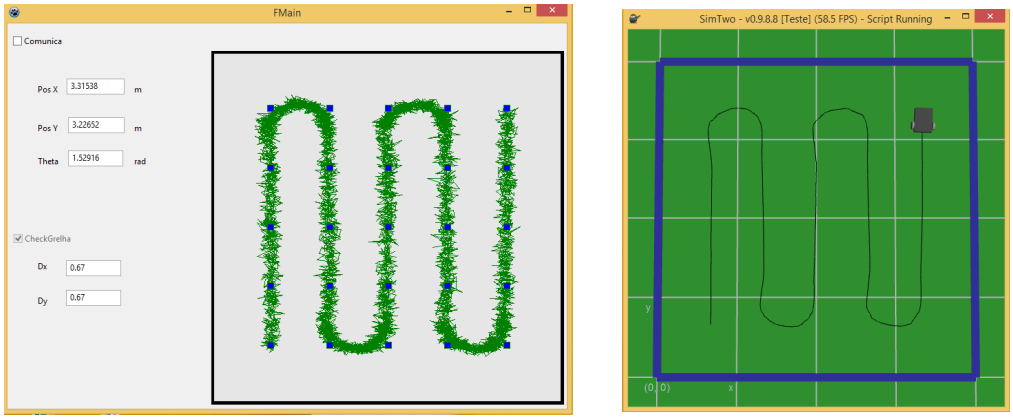
5.3.2 Segment Based Path

This subsection presents the simulation results obtained for the proposed controller in subsection 4.4.2. Table 5.8 presents the parameters of segments following controller in simulation environment. The values of K_θ and K_{error} was obtained empirically by tests.

Table 5.8: Controller parameters segment based path in simulation.

Controller Parameter	Value	Unit
$Speed$	0.05	m/s
K_θ	10	-
K_{error}	20	-

Figure 5.13a shows the location data sent from the simulator to the ControllApp. Figure 5.13b shows the path that the robot performed in the simulation environment. Compared with the results of the previous section, the following segment controller presents a smoother path, even with the presence of noises and uncertainties in the measurement of robot position.



(a) ControlApp. (b) Path taken by robot.

Figure 5.13: Segment following controller with measurement noise and uncertainty of sensor Pozyx.

Figure 5.14 presents a comparison between the reference spline trajectory for the controller and the trajectory performed by the robot. Although in this case the path is smooth, there are still errors, specially at the corners. The starting state of the robot is $[0.67, 0.67, \pi/2]^T$ and the state at the end of the path is $[3.31, 3.22, \pi/2]^T$.

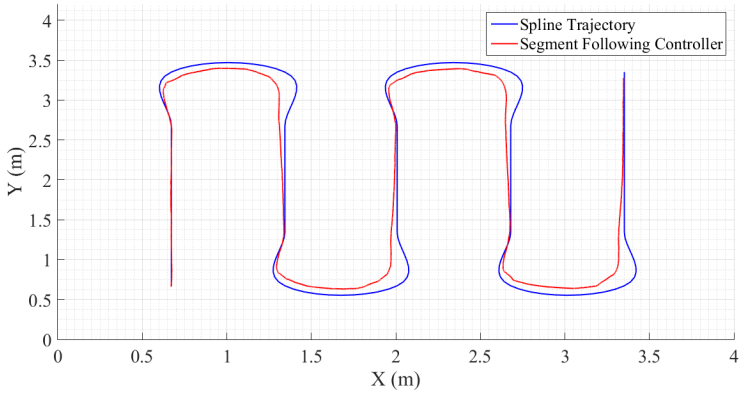


Figure 5.14: Comparison between spline trajectory and path realized by simulated robot with segment following controller.

5.3.3 Evaluation of Error Between the Spline and the Path Performed by Mobile Robot in Simulation Environment

With the visual results it was possible to notice which route is smoother or not and that the robot visited all the waypoints. However it is not possible to measure which controller caused the robot to perform the path with a smaller error. With this in mind, an evaluation of the two proposed controllers was performed based on the paths the robot performed and comparing the error with the reference spline curve.

Given the 25 waypoints of the connectivity grid, the spline interpolation inserted 24 sub-paths connecting the waypoints. The spline trajectory was sampled with 240 equidistant points distanced by 0.067 meters. For each of these points, the lowest Euclidean distance was found between it and the path executed by the robot and the value was attributed to $dist_i$, $i = (1, 2, ..., 240)$.

The quadratic error is obtained by equation 5.1 to measure the weighted deviation between the ideal path and the path executed by the robot.

$$Error = \sum_{i=1}^{240} dist_i^2 \tag{5.1}$$

Table 5.9 presents the quadratic errors obtained for the paths performed by the two controllers proposed in the simulation environment.

Table 5.9: Evaluation of controller errors in simulation environment

Control method	Quadratic *Error*
Point Following	0.1970
Segment Following	0.7231

Thus it is possible to note that even resulting in a smoother path the controlled segment following has a larger quadratic error compared to point following.

5.4 Localization in Real Indoor Environment

This section will be dedicated to present the practical results for the validation of the methodology proposed to carry out the scanning of an indoor environment.

5.4.1 Structure of the Environment

For the practical tests, an environment with a smooth, clean and plane surface was selected in order to mitigate the accumulated errors along the path through the odometry, conscious that the errors will still be present. Figure 5.15 presents the layout constructed in the classroom using adhesive tapes.

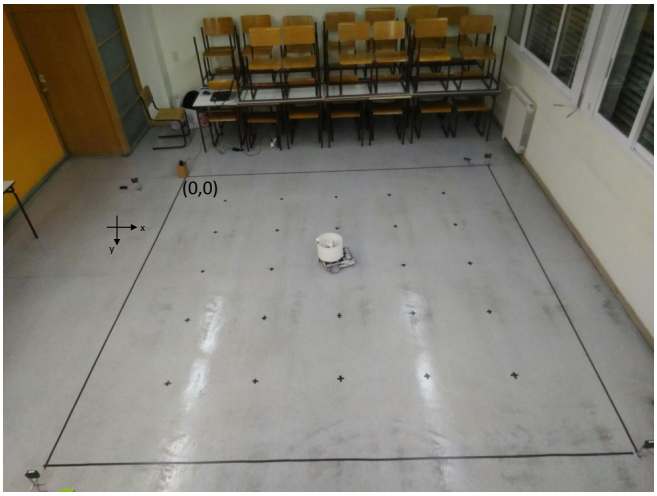


Figure 5.15: Indoor environment layout developed for testing with real robot.

Table 5.10 shows the size of the real test environment. The resolution of the connectivity grid was selected with an equidistant and centralized spacing between the waypoints.

Table 5.10: Dimension of real environment for tests with real robot.

Dimension	Value	Unit
Size in X	4	m
Size in Y	4	m
Dx	0.67	m
Dy	0.67	m
# Waypoint	25	-

5.4.2 Ground Truth and Pozyx System

The LRF Hokuyo URG-04LX was used as low cost ground truth. It is located at the source (0,0) of the environment. A region was delimited within its range resolution (see Figure 5.16) for the laser to localize the circular object embedded in the robot and consequently locate the robot, ignoring any object outside the delimited region.

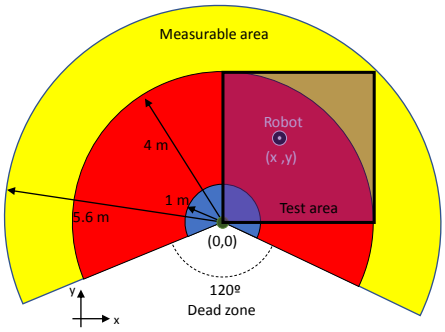


Figure 5.16: Region Labeling of ground truth.

The anchors of the Pozyx system are distributed close to the corners of the environment. Table 5.11 shows the localization of the anchors, complemented by Figure 5.17 detailing the layout of the real system.

Table 5.11: Location of Pozyx anchors.

Anchor I.D.	Location (x)	Location (y)	Unit
Anchor 0X6817	-0.5	0.3	m
Anchor 0X683F	4.0	0	m
Anchor 0X6837	0	4.300	m
Anchor 0X6830	4.0	4.0	m

5.4.3 Result Extended Kalman Filter and Control

This subsection will be devoted to presenting the results of the location of the mobile robot using data obtained from odometry and Pozyx fused by EKF. The two controllers

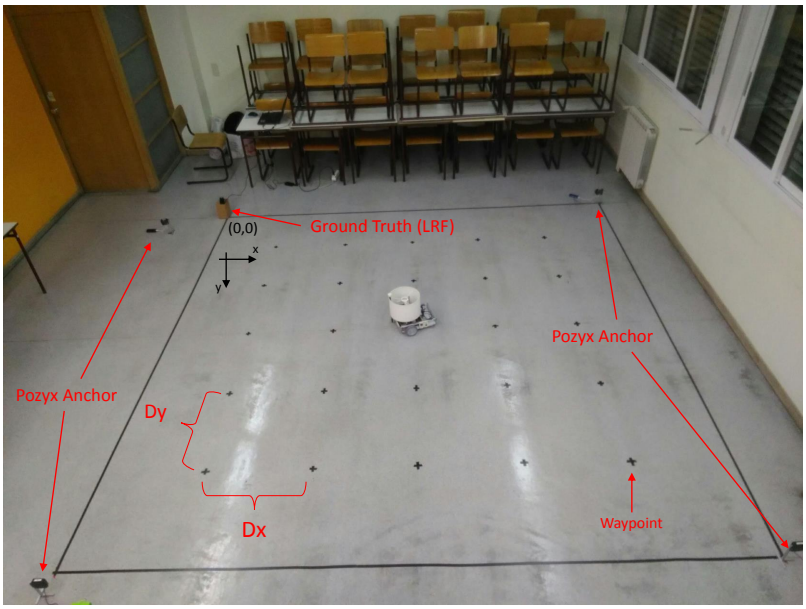


Figure 5.17: Ground truth and anchors distributed in real test environment.

will be tested and the actual path performed by the robot will be verified with data obtained from the ground truth.

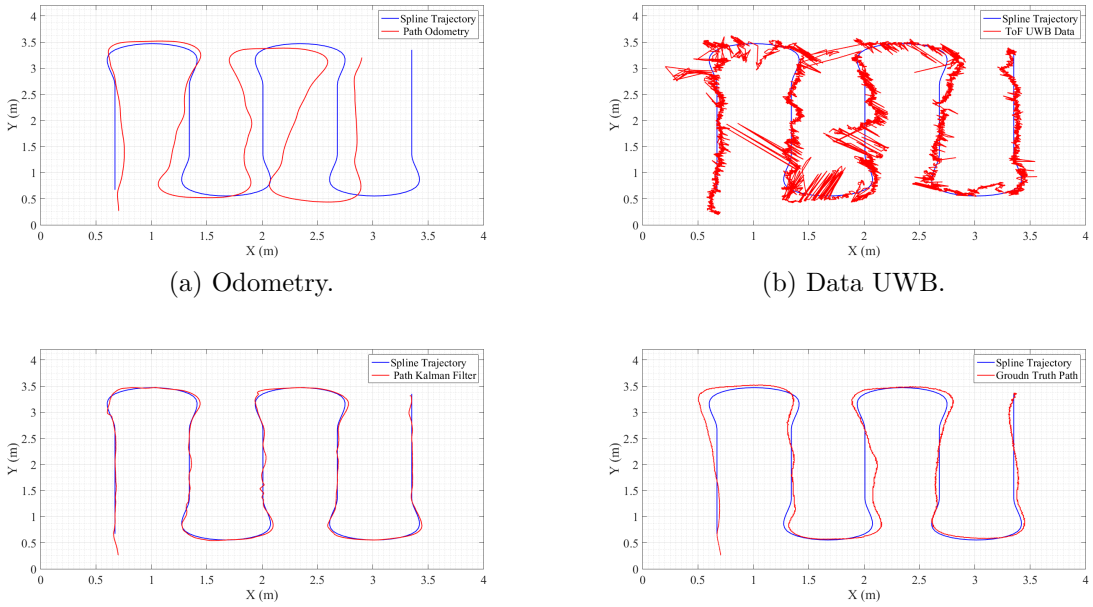
Waypoint Path in Real Environment

Table 5.12 presents the parameters of points following controller in realistic environment. As in simulation, the robot must have a reduced speed to be able to perform the scan, and the value of K_{prop} was obtained empirically by real tests.

Table 5.12: Waypoint path controller parameters in real environment

Controller Parameter	Value	Unit
V_{ref}	0.05	m/s
K_{prop}	30	-

Figure 5.18 shows the odometry, UWB, EKF and real path data compared to the reference spline curve. Figure 5.18a shows the data resulting from odometry. It is possible to observe that after finishing the first contour, the data starts to accumulate considerable errors as predicted. Figure 5.18b shows data obtained by the Pozyx sensor. Here two questions are raised. The first one refers to the increase in the noise level of the location when the robot increases its angular velocity w. The second question is the highest noise level near the region (2;2) of the environment. Figure 5.18c shows the data resultant of the fusion of data obtained by odometry and Pozyx. The actual trajectory executed by the robot is shown in Figure 5.18d obtained by the system's ground truth.



(a) Odometry.

(b) Data UWB.

(c) KalmanFilter Data.

(d) Ground Truth.

Figure 5.18: Waypoint path controller results in real environment.

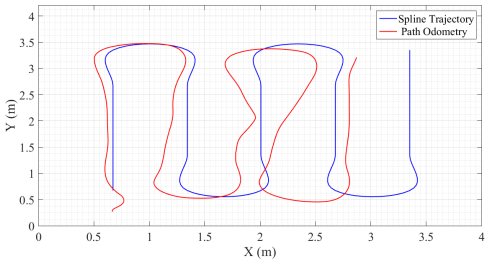
Segment Based Path in Real Environment

Table 5.13 presents the parameters of segments following controller in real environment. The values of K_θ and K_{error} was obtained empirically by real tests.

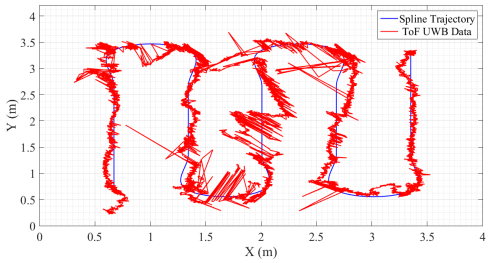
Table 5.13: Controller parameters segment based path in Real Environment .

Controller Parameter	Value	Unit
$Speed$	0.05	m/s
K_θ	6	-
K_{error}	22	-

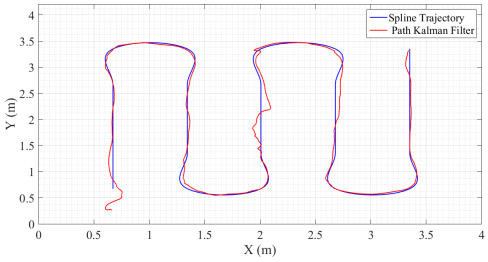
As in the previous subsection, Figure 5.19 compares the odometry, UWB, Kalman filter and real paht performed with the reference spline curve. The noise present in the region (2.2) due to the Pozyx sensor has a negative influence on this type of controller, however after passing this critical region the robot recovery its path as presented by the Figure 5.19d.



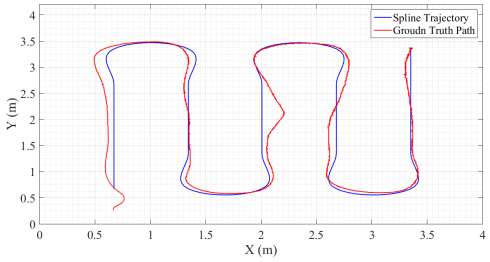
(a) Odometry.



(b) Data UWB.



(c) KalmanFilter Data.



(d) Ground Truth.

Figure 5.19: Segment based path controller results in real environment.

5.4.4 Evaluation of Error Between the Spline and the Path Performed by Mobile Robot in Real Environment

The same method used in subsection 5.3.3 using the equation 5.1 was used to evaluate the error between the paths performed by the robot and the reference path. Table 5.14 presents the result by submitting the ground truth data for the comparison.

Table 5.14: Evaluation of controller errors in real environment

Control method	Quadratic *Error*
Point Following	0.4245
Segment Following	0.7658

As in the simulation, the behavior for the following point was better. As expected, the errors in the real tests were higher than in the tests performed in simulation environments.

Chapter 6

Conclusion and Future Work

6.1 Developed Works

A methodology was developed that allows sweeping a critical (dangerous or hazardous) indoor environment for a mobile robot to perform spatial radiation map reducing or avoiding the presence that humans execute this task. Areas of mobile robotics such as path planning, control and localization were employed with the purpose of developing the methodology for spatial mapping. Two methodologies of path planning were compared, where the heuristic method proved to be more efficient for environments without obstacles and environments with unknown obstacles. The method of path planning using evolutionary algorithms is efficient when there are known obstacles and a solution is not intuitive. Two controllers were proposed, and the two guided the robot to visit all the waypoints of the connectivity grid. In real environments, the location using odometry and UWB were sufficient for the proposed work due to the Kalman filter used. If used separately it would not be possible to perform the navigation of the robot. A low cost ground truth system was used to validate the results obtained by the Extended Kalman filter.

6.2 Future Works

The present work has the potential to continue to develop. These will be described below in topics:

- Integrate the ControlApp into the mobile robot to make it autonomous.
- Embark a Geiger-Muller sensor and develop algorithms to process the data constructing the spatial radiation map.
- Improve the localization by performing corrections in the odometry with the help of QR-codes, cameras or another tool.
- Improve the path planning algorithms to environment with unknown obstacles.
- Equip the robot with sensors to locate unknown objects in the environment, to carry out the on-line re-planning of the trajectory.

Bibliography

[1] P. Suetens, *Fundamentals of medical imaging*. Cambridge university press, 2010.

[2] IAEA, *International atomic energy agency*, "Accessed on 13/12/2017 at 23:00 p.m.", 2017. [Online]. Available: `http://nucmedicine.iaea.org/default.asp`.

[3] P. N. Stuart Russell, *Artificial intelligence: A modern approach*, 3rd, ser. Prentice Hall Series in Artificial Intelligence. Prentice Hall, 2010, ISBN: 1292153970, 9781292153971.

[4] I. R. N. Roland Siegwart, *Introduction to autonomous mobile robots*, 1st. The MIT Press, 2004, ISBN: 9781417561810.

[5] S. T. (Auth.), *Introduction to mobile robot control*, 1st ed. Elsevier, 2014, ISBN: 978-0-12-417049-0.

[6] P. Rea and E. Ottaviano, "Design and development of an inspection robotic system for indoor applications," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 143–151, 2018.

[7] G. Pessin, "Estratégias inteligentes aplicadas em robôs móveis autônomos e em coordenação de grupos de robôs," PhD thesis, Universidade de São Paulo, 2013.

[8] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, *et al.*, "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.

[9] K. Nagatani, S. Kiribayashi, Y. Okada, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, and Y. Hada, "Redesign of rescue mobile robot quince," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011, pp. 13–18.

[10] I. L. Robotics, *Siar*, "Accessed on 05/01/2018 at 11:40 a.m.", 2017. [Online]. Available: http://siar.idmind.pt/.

[11] A. Macedo, A. Macedo, and J. Duarte, "Robótica aplicada ao combate a incidentes," *Revista TN Petróleo (53)*, pp. 108–113, 2007.

[12] T. in Japan, *Rescue robots: Machines play vital roles in disaster relief*, "Accessed on 05/01/2018 at 14:20 a.m.", 2017. [Online]. Available: http://http://web-japan.org/trends/09_sci-tech/sci100909.html.

[13] M. S. Laboratory, *Nasa*, https://mars.nasa.gov/msl/, "Accessed on 9/26/2017 at 21:00 p.m.".

[14] ——, *Nasa*, https://www.nasa.gov/sites/default/files/styles/full_width_feature/public/images/616880main_pia15277-full_full.jpg, "Accessed on 9/26/2017 at 22:00 p.m.".

[15] G. D. M. Jenkin, *Computational principles of mobile robotics*, 2ed. Cambridge University Press, 2010, ISBN: 978-0-521-87157-0.

[16] H. F. P. de Oliveira, "Análise do desempenho e da dinâmica de robôs omnidireccionais de três e quatro rodas," Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2007.

[17] J. A. de Carvalho Gonçalves, "Modelação e simulação realista de sistemas no domínio da robótica móvel," PhD thesis, Faculdade de Engenharia da Universidade do Porto, 2013.

[18] A. S. Conceição, A. P. Moreira, and P. J. Costa, "Design of a mobile robot for robocup middle size league," in *2009 6th Latin American Robotics Symposium (LARS 2009)*, Oct. 2009, pp. 1–6. DOI: 10.1109/LARS.2009.5418331.

[19] P. J. Costa, N. Moreira, D. Campos, J. Gonçalves, J. Lima, and P. L. Costa, "Localization and navigation of an omnidirectional mobile robot: The robot factory case study," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 11, no. 1, pp. 1–9, Feb. 2016, ISSN: 1932-8540. DOI: 10.1109/RITA.2016.2518420.

[20] S. B. Gregor Klancar Andrej Zdesar and I. A. (Auth.), *Wheeled mobile robotics. from fundamentals towards autonomous systems*, 1st Edition. Butterworth-Heinemann, 2017, ISBN: 9780128042045.

[21] M. D. Correia, A. Gustavo, and S. Conceição, "Modeling of a three wheeled omnidirectional robot including friction models," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 7–12, 2012.

[22] M. Shneier and R. Bostelman, "Literature review of mobile robots for manufacturing," May 2015. [Online]. Available: http://dx.doi.org/10.6028/NIST.IR.8022.

[23] E. A. da Silva Santos, "Logística baseada em agv," Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2013.

[24] D. D. S. Lima, "Localização absoluta de robô móveis em ambientes industriais," Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2010.

[25] Transbotics, *Transbotics*, https://www.transbotics.com/learning-center/guidance-navigation, "Accessed on 9/30/2017 at 15:00 p.m.".

[26] T. F. Cordeiro, "Sistema de deteção e contorno de obstáculos para robótica móvel baseado em sensores kinect," Master's thesis, Instituto Politécnico de Bragança, 2014.

[27] A. dos Santos Granjo Oliveira, "Localização de um agv em ambiente industrial através de um laser de segurança e marcadores artificiais," Master's thesis, Instituto Politécnico de Bragança, 2014.

[28] H. Sobreira, A. P. Moreira, P. G. Costa, and J. Lima, "Robust mobile robot localization based on security laser scanner," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, Apr. 2015, pp. 162–167. DOI: 10.1109/ICARSC.2015.28.

[29] L. Márton, C. Nagy, and Z. Biró-Ambrus, "Robust trilateration based indoor localization method for omnidirectional mobile robots," in *2016 European Control Conference (ECC)*, Jun. 2016, pp. 2547–2552. DOI: 10.1109/ECC.2016.7810673.

[30] D. Ronzoni, R. Olmi, C. Secchi, and C. Fantuzzi, "Agv global localization using indistinguishable artificial landmarks," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 287–292. DOI: 10.1109/ICRA.2011.5979759.

[31] J. Gonçalves, J. Lima, P. Malheiros, and P. Costa, "Code migration from a realistic simulator to a real robot," in *9th Conference on Autonomous Robot Systems and Competitions*, 2009.

[32] M. Lauer, S. Lange, and M. Riedmiller, "Calculating the perfect match: An efficient and accurate approach for robot self-localization," in *Robot Soccer World Cup*, Springer, 2005, pp. 142–153.

[33] H. Sobreira, M. Pinto, A. P. Moreira, P. G. Costa, and J. Lima, "Robust robot localization based on the perfect match algorithm," in *CONTROLO'2014–Proceedings of the 11th Portuguese Conference on Automatic Control*, Springer, 2015, pp. 607–616.

[34] Y. Ma, G. Zheng, and W. Perruquetti, "Cooperative path planning for mobile robots based on visibility graph," in *Proceedings of the 32nd Chinese Control Conference*, Jul. 2013, pp. 4915–4920.

[35] J.-C. Latombe, *Robot motion planning*, 1st, ser. The Springer International Series in Engineering and Computer Science 124. Springer US, 1991, ISBN: 978-0-7923-9206-4,978-1-4615-4022-9.

[36] H. Dong, W. Li, J. Zhu, and S. Duan, "The path planning for mobile robot based on voronoi diagram," in *2010 Third International Conference on Intelligent Networks and Intelligent Systems*, Nov. 2010, pp. 446–449. DOI: 10.1109/ICINIS.2010.105.

[37] X. Yang, Z. Zeng, J. Xiao, and Z. Zheng, "Trajectory planning for robocup msl mobile robots based on bézier curve and voronoi diagram," in *2015 IEEE International Conference on Information and Automation*, Aug. 2015, pp. 2552–2557. DOI: 10.1109/ICInfA.2015.7279715.

[38] H. Choset, K. Lynch, S. Hutchison, G. Kantor, W. Burgard, L. Kavraski, and S. Thrun, *Principles of robot motion: Theory, algorithms, and implementations*, ser. Intelligent Robotics and Autonomous Agents series. MIT, 2005, ISBN: 0-262-03327-5.

[39] M. Kloetzer, C. Mahulea, and R. Gonzalez, "Optimizing cell decomposition path planning for mobile robots using different metrics," in *2015 19th International Conference on System Theory, Control and Computing (ICSTCC)*, Oct. 2015, pp. 565–570. DOI: 10.1109/ICSTCC.2015.7321353.

[40] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[41] D. A. B. d. O. Vaz, "Planejamento de movimento cinemático-dinâmico para robôs móveis com rodas deslizantes," Master's thesis, Universidade de São Paulo, 2011.

[42] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[43] J. Guo, Y. Gao, and G. Cui, "Mobile robot path planning based on improved artificial potential field method," *Asian Network for Scientific Information(Information Technology Journal)*, 2013. DOI: 10.3923.

[44] S. Garrido, L. Moreno, F. Martin, and D. Alvarez, "Quality study of robot trajectories based on the anisotropic fast marching method," in *Iberian Robotics conference*, Springer, 2017, pp. 891–901.

[45] N. Sariff and N. Buniyamin, "An overview of autonomous mobile robot path planning algorithms," in *Research and Development, 2006. SCOReD 2006. 4th Student Conference on*, IEEE, 2006, pp. 183–188.

[46] P. L. C. G. d. Costa, "Planeamento cooperativo de tarefas e trajectórias em múltiplos robôs," PhD thesis, Faculdade de Enegenharia da Universidade do Porto, 2011.

[47] S. F. dos Reis Alves, "Plataforma de software para técnicas de navegação e colaboração de robôs móveis autônomos," Master's thesis, Universidade Estadual de Campinas, 2011.

[48] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012.

[49] A. Ismail, A. Sheta, and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *Journal of Computer Science*, vol. 4, no. 4, pp. 341–344, 2008.

[50] A. Larsen and O. B. Madsen, "The dynamic vehicle routing problem," PhD thesis, Technical University of DenmarkDanmarks Tekniske Universitet, Department of TransportInstitut for Transport, Logistics & ITSLogistik & ITS.

[51] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.

[52] C. S. de Assunção Gomes Costa, "Problema do caixeiro viajante - resolução e depuração," Master's thesis, Universidade de Aveiro, 2008.

[53] M. Burger, Z. Su, and B. De Schutter, "A node current-based 2-index formulation for the fixed-destination multi-depot travelling salesman problem," *European Journal of Operational Research*, vol. 265, no. 2, pp. 463–477, 2018.

[54] N. Christofides, A. Mingozzi, and P. Toth, "Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations," *Mathematical programming*, vol. 20, no. 1, pp. 255–282, 1981.

[55] Y. Kuo and C.-C. Wang, "Optimizing the vrp by minimizing fuel consumption," *Management of Environmental Quality: An International Journal*, vol. 22, no. 4, pp. 440–450, 2011.

[56] J. Li, "Vehicle routing problem with time windows for reducing fuel consumption," *Journal of Computers*, vol. 7, no. 12, pp. 3020–3027, 2012.

[57] P. T. Zacharia and N. Aspragathos, "Optimal robot task scheduling based on genetic algorithms," *Robotics and Computer-Integrated Manufacturing*, vol. 21, no. 1, pp. 67–79, 2005.

[58] N. Pereira, F. Ribeiro, G. Lopes, D. Whitney, and J. Lino, "Autonomous golf ball picking robot design and development," *Industrial Robot: An International Journal*, vol. 39, no. 6, pp. 541–550, 2012.

[59] J. Postel, "User datagram protocol," Tech. Rep., 1980.

[60] J. F. Kurose, *Computer networking: A top-down approach featuring the internet, 3/e*. Pearson Education India, 2005.

[61] P. M. d. S. P. Silva, "Movimentação autónoma de robôs móveis de baixo custo, com base no sistema nxt da lego," Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2010.

[62] Pozyx, *Pozyx*, https://www.pozyx.io/, "Accessed on 08/16/2017 at 15:00 p.m.".

[63] P. Costa, *Simtwo simulator*, "Accessed on 06/11/2017 at 1:20 p.m.", 2017. [Online]. Available: `https://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo`.

[64] P. Costa, J. Gonçalves, J. Lima, and P. Malheiros, "Simtwo realistic simulator: A tool for the development and validation of robot software," *Theory and Applications of Mathematics & Computer Science*, vol. 1, no. 1, p. 17, 2011.

[65] A. R. C. d. Andrade, "Sistema robótico autónomo para ambientes de terapia de iodo," Master's thesis, Instituto Politécnico de Bragança, 2015.

[66] A. G. Conceição, A. Moreira, P. Costa, P. Costa, and T. Nascimento, "Modeling omnidirectional mobile robots: An approach using simtwo," 2012.

[67] J. Lima and P. Costa, "Ultra-wideband time of flight based localization system and odometry fusion for a scanning 3 dof magnetic field autonomous robot," in *Iberian Robotics conference*, Springer, 2017, pp. 879–890.

[68] J. Defraye, "Determining the position of sporters using ultra-wideband indoor localization," Master's thesis, Faculty of Engineering and Architecture Ghent University, 2017.

[69] T. Conceição, F. N. dos Santos, P. Costa, and A. P. Moreira, "Robot localization system in a hard outdoor environment," in *Iberian Robotics conference*, Springer, 2017, pp. 215–227.

[70] Decawave, *Uwb transceiver*, "Accessed on 09/11/2017 at 22:00 p.m.", 2017. [Online]. Available: `http://www.decawave.com/`.

[71] H. S. P. Alves, "Navegação e controlo de um veículo móvel," Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2010.

[72] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on robotics and automation*, vol. 12, no. 6, pp. 869–880, 1996.

[73] J. Lima, J. Gonçalves, P. J. Costa, and A. Paulo Moreira, "Modeling and simulation of a laser scanner sensor: An industrial application case study," in *Advances in Sustainable and Competitive Manufacturing Systems: 23rd International Conference on Flexible Automation & Intelligent Manufacturing*, A. Azevedo, Ed. Heidelberg: Springer International Publishing, 2013, pp. 245–258, ISBN: 978-3-319-00557-7. DOI: `10.1007/978-3-319-00557-7_20`.

[74] URG-04LX, *Hokuyo*, `https://www.hokuyo-aut.jp/search/single.php?serial=166`, "Accessed on 10/18/2017 at 11:40 a.m.".

[75] Y. Okubo, C. Ye, and J. Borenstein, "Characterization of the hokuyo urg-04lx laser rangefinder for mobile robot obstacle negotiation," *International Society for Optics and Photonics in SPIE Defense, Security, and Sensing*, pp. 733 212–733 212, 2009.

[76] K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright, and H.-M. Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm," in *Evolutionary Computation, 2004. CEC2004. Congress on*, IEEE, vol. 2, 2004, pp. 1338–1345.

[77] S. Alnasser and H. Bennaceur, "An efficient genetic algorithm for the global robot path planning problem," in *Digital Information and Communication Technology and its Applications (DICTAP), 2016 sixth international conference on*, IEEE, 2016, pp. 97–102.

[78] L. Piardi, J. Lima, P. Costa, and T. Brito, "Development of a dynamic path for a toxic substances mapping mobile robot in industry environment," in *Iberian Robotics conference*, Springer, 2017, pp. 655–667.

[79] R. Moharam and E. Morsy, "Genetic algorithms to balanced tree structures in graphs," *Swarm and Evolutionary Computation*, vol. 32, pp. 132–139, 2017.

[80] S. Sivanandam and S. Deepa, *Introduction to genetic algorithms*. Springer Science & Business Media, 2007.

[81] C. Sprunk, "Planning motion trajectories for mobile robots using splines," Master's thesis, University of Freiburg, 2008.

[82] L. C. Figueiredo and F. G. Jota, "Introduçao ao controle de sistemas nao-holonômicos," *Sba: Controle & Automação Sociedade Brasileira de Automatica*, vol. 15, no. 3, pp. 243–268, 2004.

[83] W. R. B. Ferreira *et al.*, "Planejamento de trajetórias robóticas utilizando b-splines," Master's thesis, Universidade Federal de Uberlândia, 2011.

[84] R. P. C. Steven C. Chapra, *Numerical methods for engineers, 6th edition*. McGraw-Hill Higher Education, 2009, ISBN: 0073401064,9780073401065.

[85] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer Science & Business Media, 2008.

Appendix A

Publications

Piardi, L., Lima, J., Costa, P., & Brito, T. (2017, November). "Development of a Dynamic Path for a Toxic Substances Mapping Mobile Robot in Industry Environment". In Iberian Robotics conference (pp. 655-667). Springer, Cham.

Piardi, L., Lima, J., Brito, T., "Autonomous mobile robot for mapping toxic substances in industrial environment", 4o Encontro de Jovens Investigadores do IPB 2017. Bragança.

Paper Submitted and Accepted

Piardi, L., Lima, J., Costa, P.,Pereira, A. I., "Path Planning Optimization Method Based on Genetic Algorithm for Mapping Toxic Environment" In 8th International Conference on Bioinspired Optimization Methods and their Applications (BIOMA).

Papers Submitted and Awaiting Evaluation

Piardi, L., Lima, J., Costa, P, " Trajectory Control of a Toxic Substances Mapping Autonomous Robot With an Ultra-Wideband Time of Flight Localization System", In Emerging Technologies And Factory Automation (ETFA2018).

Piardi, L., Lima, J., Costa, P, "Low cost ground truth system with LRF to tracking mobile robot", In International Conference on Informatics in Control (ICINCO2018).

Appendix B

Petri Net: Software Application Embedded in the Robot

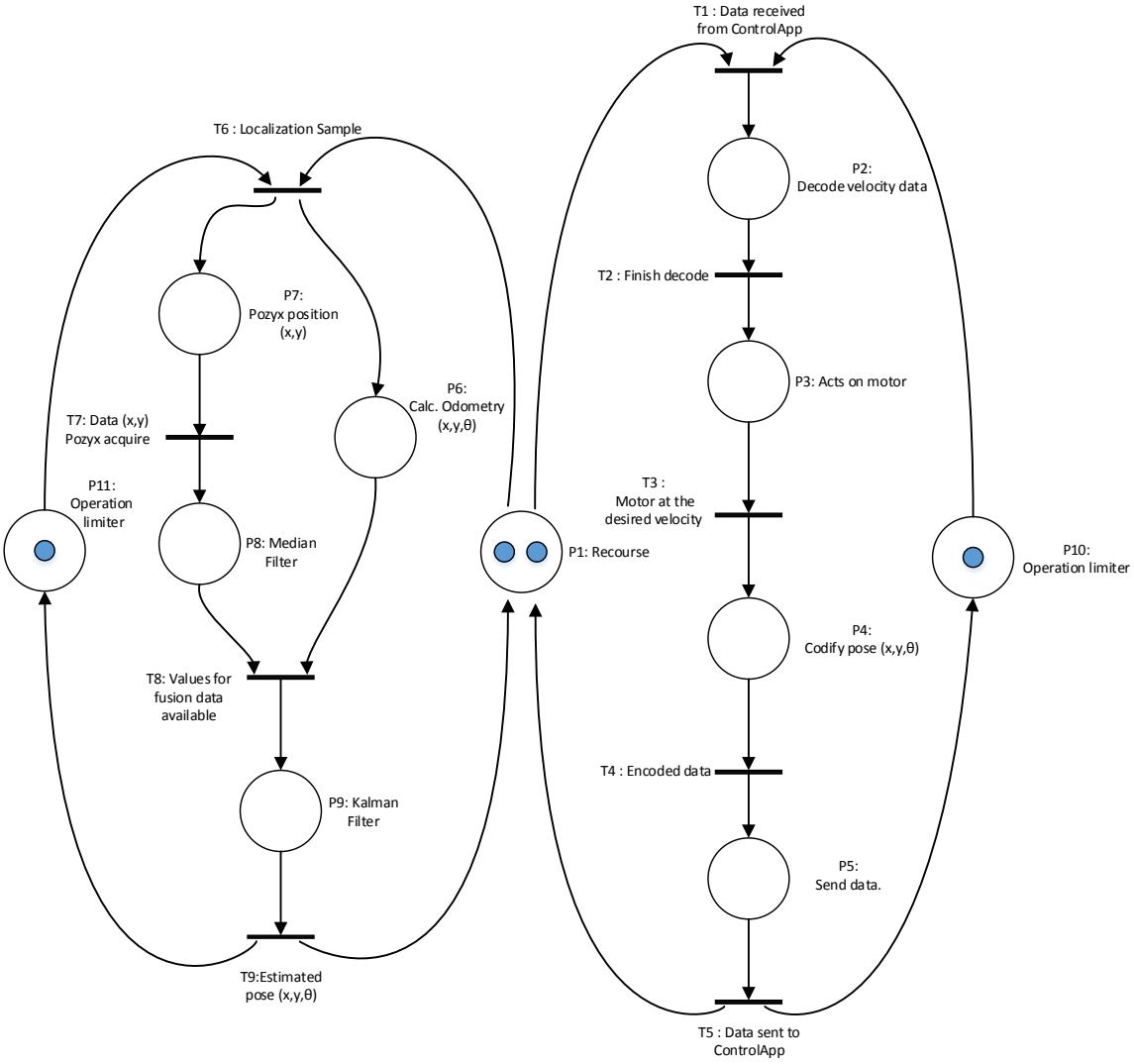


Figure B.1: Petri Net: software application embedded in the robot to act on the wheels, communicate with ControlApp and perform the location by the EKF.

Appendix C

Flowchart Heuristic Method

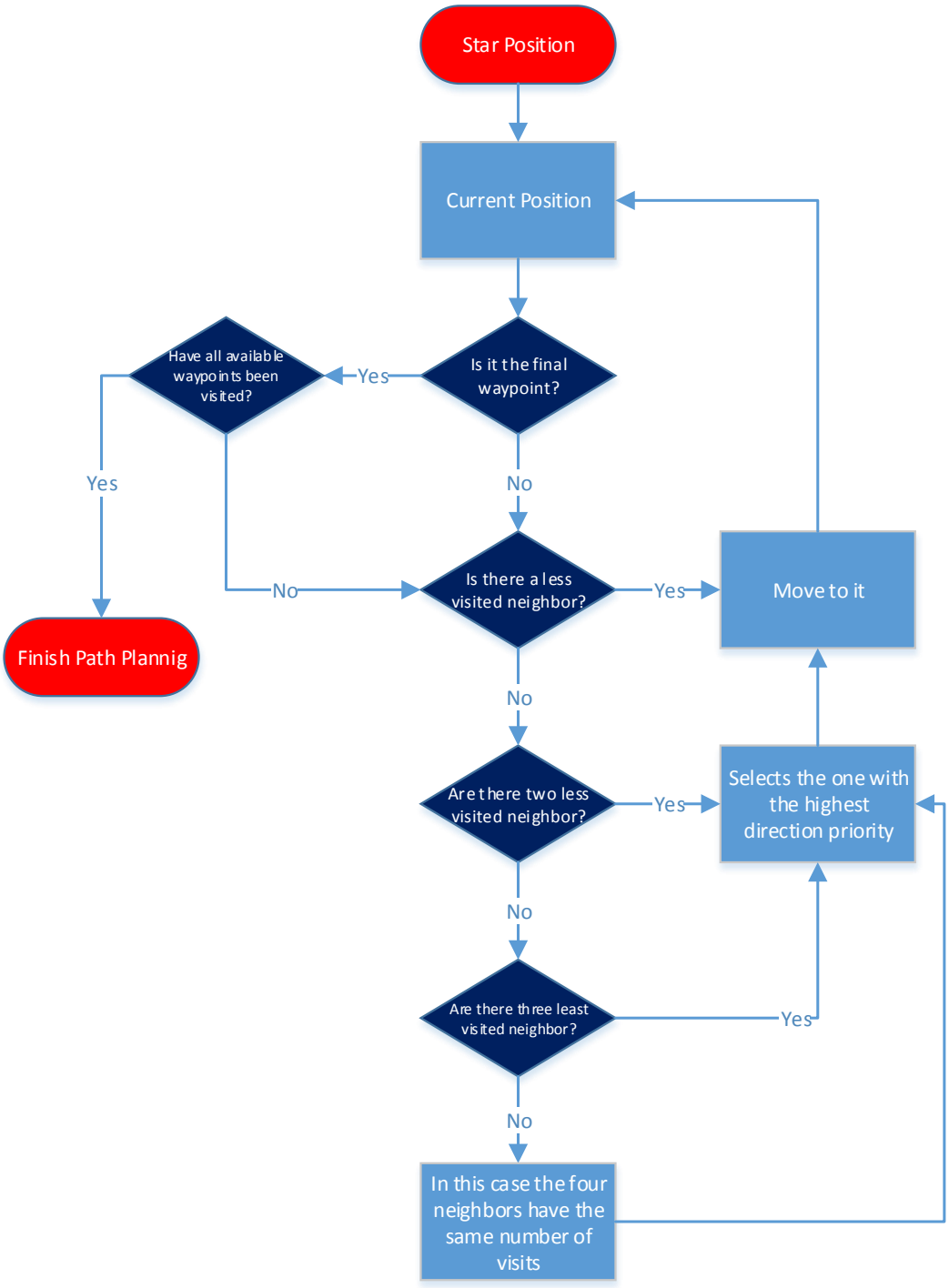


Figure C.1: Flowchart to accomplish path planning considering a direction priority. There are eight direction priorities, executed one at a time.

Appendix D

Petri Net: unknown obstacle detected and path re-planning.

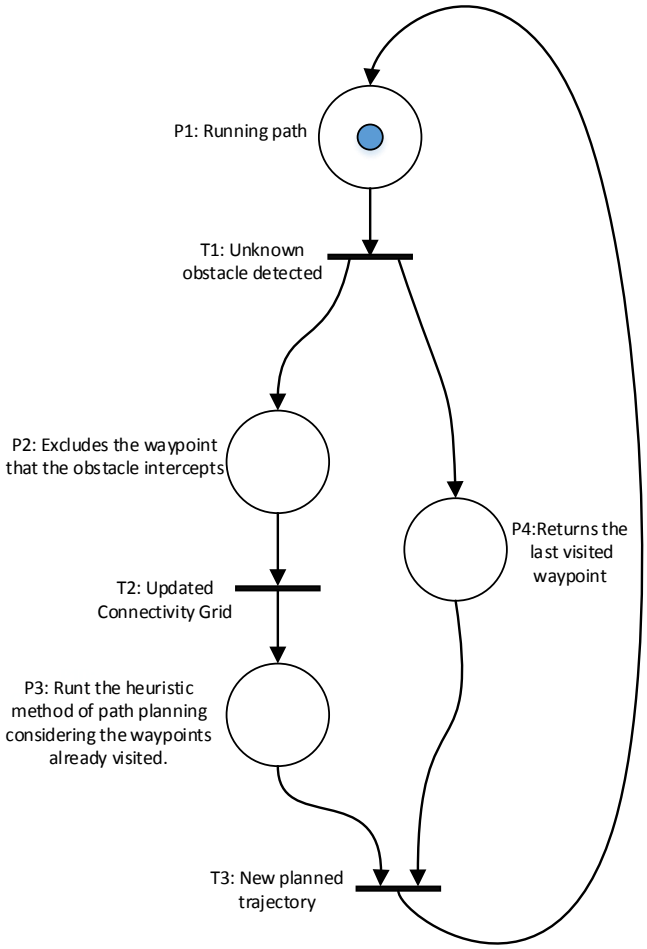


Figure D.1: Petri Net: unknown obstacle detected and path re-planning.