PROJ-H405 - Project in Electromechanical Engineering

Optimal route planning and control of a mobile manipulator for material handling in construction

Author:

REKIEK Saad

Promotor:

GARONE Emanuele

Advisors:

ASANI Zemerart

BENHAMOUCHE Ouassim

2023-2024

Contents

Abstract

In construction site environments, it is a common need to carry multiple loads to different locations within the space continuously and to optimize the path to accomplish this as quickly as possible. Therefore, the idea of using a robot is very appealing for relatively heavy loads and for navigating through small spaces. However, on a construction site, there are both fixed and dynamic obstacles that need to be avoided. The objective of this project is to study the kinematics and dynamics of the robot to enable it to move through the workspace using its omniwheels. To be able to navigate in environment in presence of obstacles we will explore path plannings algorithms based on the Artificial Potential Field (APF) methods. APF is a path planning method that treats obstacles as repulsive forces and the target destination as an attractive force for the robot. Consequently, the robot is subjected to a force that guides it along a specific path to avoid obstacles while reaching the final destination. Unfortunately, APF suffers from a problem known as local minima, which can cause the robot to get stuck in certain situations where no force acts on it. Therefore, it is necessary to investigate other method that takes into account the global study of the workspace. A possible way to deal with this problem is the possibility to use global navigation functions instead. The global navigation function acts on the robot in such a way that it avoids the local minima problem inherent in the APF method. This global navigation field method, coupled with the local APF method, will allow the robot to avoid the local minima problem and take also into account dynamic obstacles on the construction site. As a result, the robot will be able to optimize its path and perform its tasks as efficiently as possible.

List of Figures

1 Introduction

The project involves delivering materials/equipment to human workers on a construction site. Different equipment/materials need to be transported from point A to point B, navigating through the work environment (the construction site). In this work environment, there will be humans moving around, which are considered as mobile obstacles, as well as fixed obstacles such as building beams. This project is carried out using a mobile robot that can carry loads and move around the construction site. The goal is to enable the coexistence of human workers and the robot in an unstructured environment such as a construction site. Additionally, there needs to be an order in which materials are delivered to a given position in the work environment. However, this project assumes that the ordering is already predefined (given) and we are concerned only with the navigation between point A and B.

1.1 Problem statement and assumptions

Construction sites are clustered environments with many obstacles. Some of these obstacles are static, such as machines, materials, etc. Other obstacles are dynamic(i.e., obstacles that move), such as humans or other mobile robots. The challenge for the robot is to avoid both static and dynamic obstacles that it encounters on the construction site. Another objective is determining the order in which the robot should visit various target points while optimizing for the shortest path and avoiding obstacles. Assuming the optimized shortest path is known, it is essential to ensure that the robot can reach each desired intermediate point while avoiding all obstacles. The project is divided into two parts: navigation while avoiding obstacles on the construction site, and study of the kinematics and dynamics of an omnidirectional mobile robot able to navigate in construction site using the proposed methods. The project involves a mobile manipulator capable of carrying equipment and equipped with omnidirectional wheels, allowing for instantaneous movement in x, y, and θ.

1.2 Methodology

Taking into account the assumptions and the problem statement, it is necessary to establish a methodology to advance the project step by step. For this, two approaches are studied. The first approach involves combining global and local planners. The local planner focuses on planning the path locally, meaning the robot is influenced in its direction only within a limited area around while the global planner plans the path globally, meaning the robot is influenced by the environment as a whole.

Then, the development of the global navigation field will be studied to address issues associated with local planning methods based on artificial potential fields. A major problem encountered with local planners, such as the artificial potential field method, is the local minima issue, where the robot can get stuck due to the geometry and arrangement of certain obstacles in the work environment. On the other hand, global planners like A* or RRT can provide the shortest path but do not account for dynamic obstacles. Therefore, combining both planners will help avoid local minima through the global planner and account for dynamic obstacles through the local planner.

A theoretical study will also be conducted to transition from a star-shaped space (which is typically the kind of shape found on construction sites) to a spherical space (where all obstacles are considered spherical) which simplifies and allows for global propreties. To achieve this, it is necessary to move from the star-shaped space to the basic theoretical spherical space, which can be accomplished through diffeomorphism. This property will enable the consideration of all types of star-shaped obstacles. These obstacles closely resemble real-world conditions and will allow the robot to avoid obstacles more effectively.

1.3 State of the art

Navigation and path planning is a very important element in robotics. Several studies exist that solve different problems or improve upon existing methods. The different existing methods will be listed along with their advantages and disadvantages.

1. Sampling-Based Methods

Sampling-based methods avoid explicitly constructing the configuration space (C-Space), which can be computationally expensive, especially in high-dimensional spaces. Different methods exist:

- Probabilistic Roadmaps (PRM): Samples random points in the C-Space and connects them to form a roadmap. This method is effective in high-dimensional spaces and environments with many obstacles [Ka96].
- Rapidly-exploring Random Trees (RRT): Builds a tree by randomly sampling the C-Space and expanding the tree towards the samples. It is particularly useful for single-query path planning in complex, high-dimensional spaces [LaVnd].

2. Graph-Based Methods

Graph-based methods model the environment as a graph, where nodes represent possible robot states and edges represent transitions between these states. Different methods exist:

- Dijkstra's Algorithm: Finds the shortest path between nodes in a weighted graph. Although it guarantees the shortest path, it can be computationally intensive for large graphs [Ven14].
- A*: An extension of Dijkstra's algorithm that uses heuristics to guide its search, significantly improving efficiency while still finding an optimal path [Che11].
- D*: Dynamic A* adapts to changes in the environment, making it suitable for scenarios where obstacles can appear or disappear [Ste94].

3. Artificial Intelligence-Based Methods

AI-based methods leverage machine learning and other AI techniques to improve path planning. Here is two methods existing for AI methods:

- Reinforcement Learning (RL): Enables robots to learn optimal paths through trial and error by maximizing cumulative rewards. Techniques like Deep Q-Networks (DQN) [Sew19] and Proximal Policy Optimization (PPO) [Sa17] have been successfully applied in robotic navigation.
- Neural Motion Planners: Use deep neural networks to learn and predict feasible paths from training data. These planners can generalize to new environments but require substantial training data and computational resources [Za21].

4. Optimization-Based Methods

Optimization-based methods formulate path planning as an optimization problem, seeking to minimize a cost function (path length, energy consumption). Prominent techniques include:

- Trajectory Optimization: Involves optimizing a trajectory directly, often using gradient-based methods. Approaches like CHOMP (Covariant Hamiltonian Optimization for Motion Planning) [Za13] and STOMP (Stochastic Trajectory Optimization for Motion Planning) [Ka11] are well-known in this category.

- Model Predictive Control (MPC): Plans paths by solving an optimization problem at each time step, considering future states and optimizing over a prediction horizon. It is well-suited for dynamic and real-time applications [ak17].

5. Hybrid Methods

Hybrid methods combine elements from different planning techniques to leverage their respective strengths. Here is two examples :

- PRM-RRT: Combines PRM and RRT to balance the benefits of roadmap construction and random tree exploration [Ba03].

- Learning-Based Sampling: Enhances sampling-based methods by incorporating learned models to guide the sampling process more effectively [BP17].

In conclusion, the field of robot path planning is dynamic and continually evolving. Each method has unique strengths suitable for different scenarios. Graph-based and sampling-based methods provide reliable solutions for static and high-dimensional environments. Optimization-based methods offer refined control over trajectory quality, while AI-based approaches introduce adaptability and learning capabilities. Hybrid methods aim to combine the best features of these techniques, pushing the boundaries of what is achievable in robotic path planning. As research progresses, further innovations and improvements are expected, leading to more efficient and versatile robotic navigation systems.

2 Kinematics and dynamics of the kuka youbot robot

This chapter will address the robot considered for the tasks, the assumptions made regarding this robot to facilitate its mathematical model. First, the kinematic model will be established and discussed with various simulations. Finally, the dynamic model will also be established.

2.1 Kuka Youbot

In recent years, the advancement of manufacturing has driven the development of autonomous mobile robots for handling and transporting workpieces and equipment.

For this purpose, the robot was chosen based on its ease of mobility in places such as a construction site, as required for the project. Finally, it should be as compact as possible to avoid cluttering the construction site space.

The Kuka YouBot mobile manipulator is robot that is available in the SAAS lab, therefore is studied for our use-case. This robot allows the mobility that one needs to navigate easily in clustered environments. This robot meets the requirements for testing our proposed methods in lab settings. In addition, it can be used in construction sites where the first layer has been already well done, so the terrain is not rough. However, for future work and for implementation in construction sites, robots such as Husky from clearpath [Roba] can be used as they are made specifically for rough terrains and heavy loads. The proposed methods are general and can be used with any mobile robot. This Kuka Youbot robot features an arm mounted above its mobile base. The arm has 5 degrees of freedom to transport various loads necessary on the site and move them to another location. It also has special wheels called omniwheels [Sa].

Regular wheels experience significant friction when turning as they need to drive slightly sideways during the turn. In contrast, omni-wheels are equipped with built-in rollers that can rotate sideways, allowing them to turn with minimal resistance.

These wheels offer strong driving force and easy control performance, which can enhance the

Figure 1: Kuka Youbot with the 5-DOF arm and 4 omniwheels [Sa]

utilization of space on the construction site and the efficiency of transportation within the environment.

2.2 Assumptions

Several assumptions [YD19] have been made to simplify the mathematical aspect of the robot's mechanics:

- Between roller and ground there is pure rolling.
- Contact between roller and ground does not slip
- Slippage between wheels and ground is a disadvantage because there is a loss of velocity and it affects the positionning accuracy.

2.3 Kinematics

In this section dedicated to the robot's kinematics, a mathematical development ([YD19],[Sic08]) will be conducted to model the kinematics of an omniwheel based on intrinsic parameters of the robot, which will be defined. Subsequently, a generalization will be made to 4 omniwheels, as is the case with Kuka Youbot.

In figure 2, a diagram illustrating the different coordinate systems: a fixed coordinate system, a rotating coordinate system, and a last one for the rollers attached to the wheel.

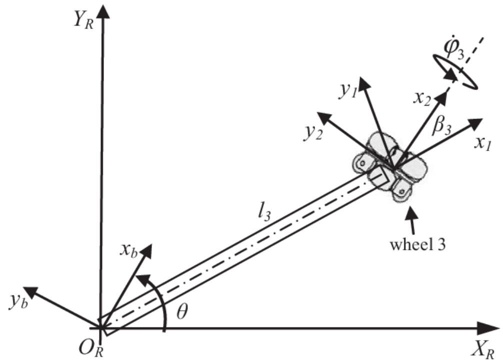

Figure 2: Omniwheel with a fixed coordinate system $O_R X_R Y_R$, a rotating coordinate system $O_R x_b y_b$, a system coordinate fixed for wheel with $x_l and y_l$ and a rotating coordinate system for wheel with $x_2 and y_2$. θ is the angle of rotation of the robot. β is the angle of rotation of the wheel. [Ma]

Thanks to the assumptions made in the previous chapter, it is possible to write two constraint equations for an omniwheel:

$$\dot{x}sin(\alpha_i + \beta_i) - \dot{y}cos(\alpha_i + \beta_i) - l_i\dot{\theta}cos(\beta_i) = r_w\phi_i - v_g cos(\gamma_i), \tag{2.1}$$

$$\dot{x}cos(\alpha_i + \beta_i) + \dot{y}sin(\alpha_i + \beta_i) + l_i\dot{\theta}sin(\beta_i) = -v_g sin(\gamma_i), \tag{2.2}$$

where :

- $\dot{x}$ and $\dot{y}$ represent the relative rotational speeds of the omniwheel along the x-axis and y-axis respectively.
- l_i is the distance between the center of the omniwheel and the center of one of its rollers.
- α_i is the angle between the inward X-axis of the omniwheel and one of its rollers.
- β_i is the angle between the perpendicular axis of a roller and the axis containing the center of the omniwheel and the roller.
- ϕ_i is the relative rotation angle of a roller.
- r_w represents the radius of the omniwheel.
- v_g is the speed of the roller.

Since the speed v_g is not directly controllable in practice, it will be eliminated from both equations (2.1, 2.2) :

$$\dot{x}cos(\alpha_i+\beta_i+\gamma_i)+\dot{y}sin(\alpha_i+\beta_i+\gamma_i)+l_i\dot{\theta}sin(\beta_i+\gamma_i)=-r_w\phi_i sin(\gamma_i). \tag{2.3}$$

To model the robot's kinematics, we need to have the rotational speed of the wheels in terms of global coordinates (the coordinates of the robot's working environment). To achieve this, we need to isolate the rotational speed of an omniwheel in the equation and express it in terms of global coordinates :

$$\phi_i=\frac{-1}{r_w sin(\gamma_i)}\begin{bmatrix}cos(\alpha_i+\beta_i+\gamma_i) & sin(\alpha_i+\beta_i+\gamma_i) & l_i sin(\beta_i+\gamma_i)\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}. \tag{2.4}$$

After isolating ϕ_i, we must transform the robot's relative coordinates $\dot{x}, \dot{y}, \dot{\theta}$ to the global coordinates of the construction site $\dot{x}_I, \dot{y}_I, \dot{\theta}_I$:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}cos(\theta) & sin(\theta) & 0\\ -sin(\theta) & cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}_I\\ \dot{y}_I\\ \dot{\theta}_I\end{bmatrix}. \tag{2.5}$$

Once this is done, we need to generalize the kinematic equation of a single wheel to a 4-wheel system, incorporating assumptions to facilitate calculations. In Figure 3, it shows the robot with the 4 wheels and the 2 coordinate systems:

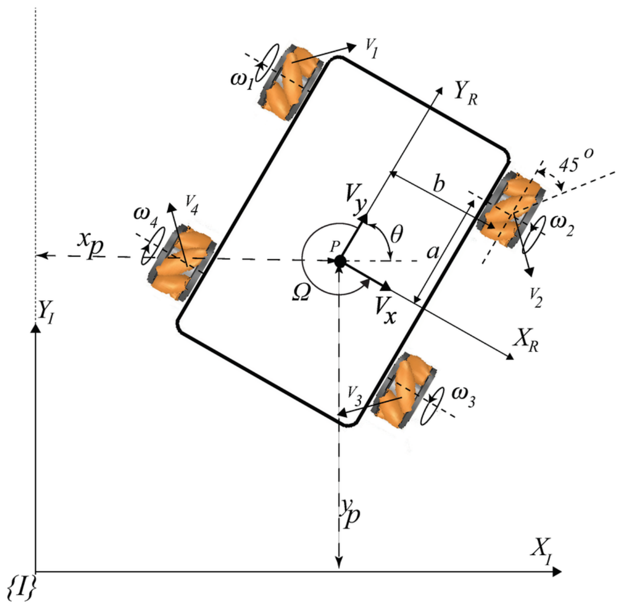

Figure 3: Coordinate system OX_IY_I is the global fixed system (environment coordinates). x_p and y_p are the coordinates of the center of the robot in the fixed coordinate system. $\omega_1, \omega_2, \omega_3$ and ω_4 are the rotational speeds of the wheels. X_RY_R is the relative coordinate system of the robot. a and b are length parameters. θ is the angle of rotation of the robot in the environment. [Aa]

The assumptions are made for each wheel. As it can be seen on Figure 4, the rollers have different angle disposition related to the base of the robot. We need to make assumptions on those different angles:

- $\varphi_1 = \varphi_3 = -45°$
- $\varphi_2 = \varphi_4 = 45°$
- $\alpha_i + \beta_i = 0$

The assumption of the sum of angles α and β equal to 0 will be used to derive the final equation for a single wheel :

$$\phi_i = \frac{-1}{r_w}\begin{bmatrix} cot(\gamma_i) & 1 & D - Hcot(\gamma_i) \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}. \tag{2.6}$$

Finally, by generalizing to all four wheels, the obtained equation is as follows :

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \frac{-1}{r_w}\begin{bmatrix} -1 & 1 & D+H \\ 1 & 1 & -(D+H) \\ -1 & 1 & -(D+H) \\ 1 & 1 & D+H \end{bmatrix}\begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{x}_l \\ \dot{y}_l \\ \dot{\theta}_l \end{bmatrix}, (2.7)$$

where H and D are lengths of the robot represented in the Figure 4.

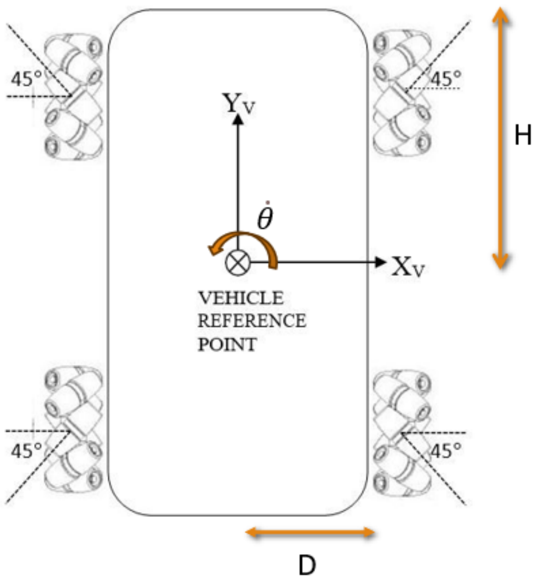

Figure 4: Length parameters H and D of the robot

The final equation has been obtained. The rotational speed is calculated based on a global matrix containing parameters specific to the robot and global coordinates that need to be integrated to obtain positions.

2.4 Dynamics

The Kuka Youbot robot can also be controlled in torque mode with both the mobile base and the arm [dev]. Here are the equations of the dynamic model of the robot with 4 omniwheels ([HR17]):

$$\begin{aligned}(m+\frac{4I_{rotwheel}}{r^2})[cos\theta(\ddot{x}_l+\dot{y}_l\dot{\theta})+sin\theta(\ddot{y}_l-\dot{x}_l\dot{\theta})] = \\ =\frac{1}{r}[\tau_1+\tau_2+\tau_3+\tau_4-\frac{G.f}{4}(sign(\dot{\phi}_1)+sign(\dot{\phi}_2)+sign(\dot{\phi}_3)+sign(\dot{\phi}_4))]\end{aligned} \tag{2.8}$$

$$\begin{aligned}(m+\frac{4I_{rotwheel}}{r^2})[cos\theta\ddot{y}_l-\dot{x}_l\dot{\theta})+sin\theta((\ddot{x}_l+\dot{y}_l\dot{\theta})] = \\ =\frac{-1}{r}[\tau_1-\tau_2-\tau_3+\tau_4-\frac{G.f}{4}(sign(\dot{\phi}_1)-sign(\dot{\phi}_2)-sign(\dot{\phi}_3)+sign(\dot{\phi}_4))]\end{aligned} \tag{2.9}$$

$$\begin{aligned}(I_{pc}+\frac{4I_{rotwheel}(H+D)^2}{r^2})\ddot{\theta} = \\ =\frac{-(H+D)}{r}[\tau_1-\tau_2+\tau_3-\tau_4-\frac{G.f}{4}(sign(\dot{\phi}_1)-sign(\dot{\phi}_2)+sign(\dot{\phi}_3)-sign(\dot{\phi}_4))]\end{aligned} \tag{2.10}$$

where :

- I_{pc} is the moment of inertia of the base plus the 4 omniwheels;
- G is the weight of one omniwheel;
- f is the coefficient of rolling friction;
- $I_{rotwheel}$ is the mass moment of inertia determined in relation to the axis of own rotation of the wheel;
- m is the mass of the robot;
- $\tau_1, \tau_2, \tau_3, \tau_4$ are the driving moment of the wheels;

2.5 Kinematics model validation

The kinematics model that is computed in section 2.3, is implemented in CoppeliaSim [Robb] and can be seen in Figure 5 :

```
function CalculationOmegaWheels(vx,vy,w,r,h,d)
    w1= -(vy-vx+(h+d)*w)/(r)
    w2= (vy+vx-(h+d)*w)/(r)
    w3= -(vy-vx-(h+d)*w)/(r)
    w4= (vy+vx+(h+d)*w)/(r)
end
```

Figure 5: Implementation of kinematics model with the 4 rotation speeds of the wheels

where :

- vx and vy are the speeds of the center of the robot
- h and d are lengths parameters of the robot
- w is the rotation speed of the robot
- r is the radius of the robot's wheel

When the implementation is done, we get the Coppelia simulation for different cases :

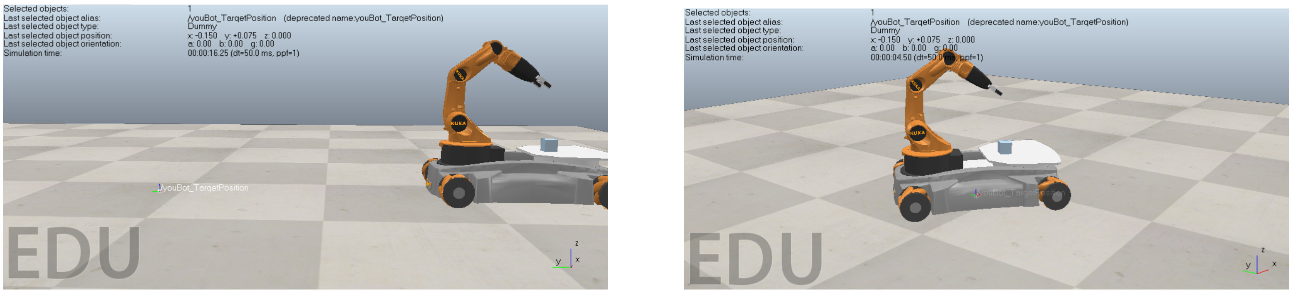

Figure 6: a) Translation of the kukaYoubot b) Rotation of the kukaYoubot

3 Artificial potential field methods

In this chapter, various path planning methods will be discussed along with their distinct characteristics. Subsequently, the Artificial Potential Field (APF) will be addressed and modeled, highlighting its advantages and drawbacks. Simulations of this method will be presented and elucidated. Finally, the major issue of the APF method, namely local minima, will be explained, and solutions will be proposed to overcome this problem.

3.1 Path planning methods

The goal of path planning in robotics is to find the optimal path for the robot that perfectly satisfies the constraints encountered in a specific environment, such as a construction site in this case.

To find this optimal path, it is necessary to create a model that enable understanding of the environment in which the robot will operate and guide it accurately. For this purpose, several methods exist to accomplish this task. In the present project scenario, where the environment is a construction site, path planning will focus on situations where the robot navigates in a known environment. This means that the model for path planning will be such that the generated path to be followed by the robot (knowing the positions of static obstacles and assuming that the position of dynamic obstacles can be mesured using camera system) is already known.

3.2 Artificial Potential Field

The chapter will focus on the Artificial Potential Field (APF) method. These sources were used in this chapter([CL05],[Rym],[A N22],[XL20],[Kha86],[PP23],[TN15]).

The APF method involves considering the robot as an object affected by an artificial potential field. The robot's destination point (target) and obstacles in the environment are treated as attractive potential fields for the robot's destination point and repulsive potential fields for the obstacles.

This means that the robot is drawn towards the destination point while being simultaneously repelled by the obstacles it needs to avoid. The sum of all potential functions defines the potential function of the free space.

The attractive potential is modeled as follows:

$$U_{att}(x) = \begin{cases} \frac{1}{2}k_{att}||x - x_{final}||^2 : ||x - x_{final}|| \leq d \\ dk_{att}||x - x_{final}|| - \frac{1}{2}k_{att}d^2 : ||x - x_{final}|| > d \end{cases}, \tag{3.1}$$

where :

- k_{att} is a parameter related to the attractiveness of the objective, meaning that the higher this parameter, the more the robot will be drawn towards the objective while neglecting the repulsiveness of obstacles
- $||x - x_{final}||$ is the norm of the distance between the current position of the robot and the final position to reach.

In the first case, when the distance between the current position of the robot and the final position to reach is smaller or equal to d, which is an arbitrarily chosen minimum distance, the attractive potential is proportional to the square of $||x - x_{final}||$. This can be interpreted by the fact that as the robot approaches within a distance smaller than d, the potential decreases with the square of $||x - x_{final}||$ until it reaches 0 when the robot has arrived at the final position.

In the second case, the robot is at a distance greater than d from the final position to reach. In this case, the potential includes a term proportional to $d||x - x_{final}||$ and a term proportional to d^2.

On the other hand, the repulsive potential due to the obstacles is defined as:

$$U_{rep}(x) = \begin{cases} \frac{1}{2}k_{rep}(\frac{1}{\rho(x)} - \frac{1}{\rho_0})^2 : \rho(x) \leq \rho_0 \\ 0 : \rho(x) > \rho_0 \end{cases}, \tag{3.2}$$

where :

- k_{rep} is a parameter related to the repulsiveness of obstacles, meaning that the higher this parameter, the more the robot will be repelled by the obstacle it faces.
- $\rho(x)$ is the shortest distance between the robot at a given position x and the obstacle.
- ρ_0 is an arbitrarily chosen minimum distance.

In the first case, when $\rho(x)$ is smaller or equal to ρ_0, indicating that the robot is relatively close to the obstacle at a distance less than or equal to ρ_0, then the repulsive potential is proportional to $(\frac{1}{\rho(x)} - \frac{1}{\rho_0})^2$. In this term, as $\rho(x)$ approaches 0, meaning it gets very close to the obstacle, $\frac{1}{\rho(x)}$ tends towards a very large value, implying that the repulsive potential will be very large. This is logical, as the obstacle will exert stronger repulsion as the robot gets closer to avoid collision.

In the second case, when $\rho(x)$ is greater than ρ_0, it means the robot is sufficiently far from the obstacle and therefore, the obstacle does not significantly affect the robot's field. The value of the potential at this moment is equal to 0.

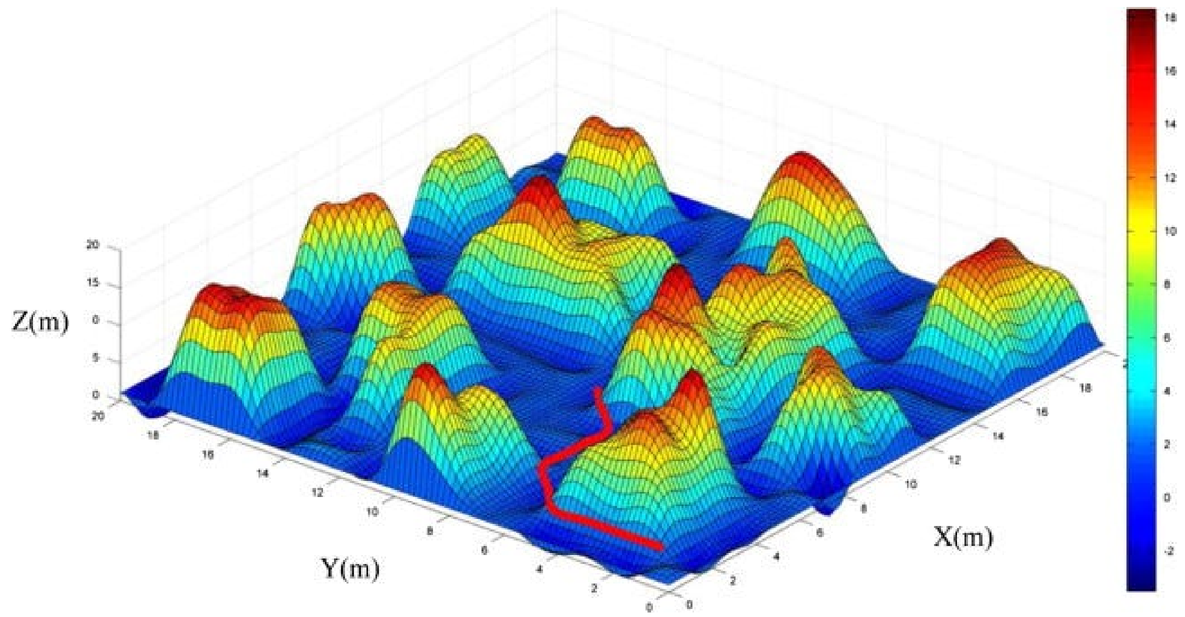

Figure 7: APF with path in red and Potential with different values close or far from obstacles [Rym]

Therefore, the robot is subjected to a potential function $U(x) = U_{att}(x) + U_{rep}(x)$. With this potential function, we can calculate the generalized force applied to the robot so that it can move in the environment towards its objective while avoiding obstacles. This generalized force is equal to the negative gradient of the potential function:

$$\nabla U_{att}(x) = \begin{cases} k_{att}(x - x_{final}) : ||x - x_{final}|| \le d \\ \frac{dk_{att}(x - x_{final})}{||x - x_{final}||} : ||x - x_{final}|| > d \end{cases}, \tag{3.3}$$

$$\nabla U_{rep}(x) = \begin{cases} -k_{rep}(\frac{1}{\rho(x)} - \frac{1}{\rho_0})\frac{\nabla\rho(x)}{\rho(x)^2} : \rho(x) \le \rho_0 \\ 0 : \rho(x) > \rho_0 \end{cases}, \tag{3.4}$$

where $\nabla\rho(x) = \frac{x-b}{||x-b||}$. Here, b represents the coordinate of the closest point of the obstacle to the robot.

3.3 Simulation

In this section, two cases of the artificial potential field will be studied. The first scenario involves placing randomly shaped obstacles at random positions along the robot's path. The robot's initial position and the final target position are also chosen randomly. Then, the potential field is applied to guide the robot to the final position while avoiding the obstacles.

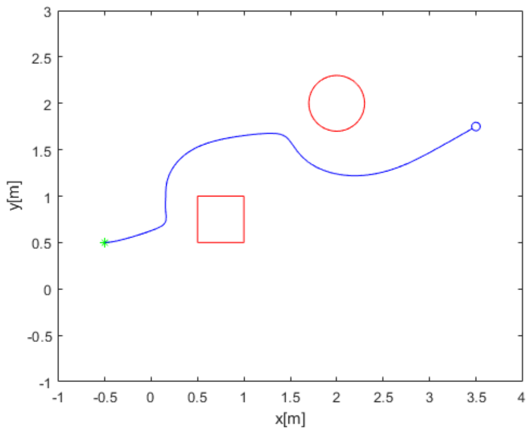

Figure 8: APF with path in blue, initial position is green star, blue circle is the final position and the obstacles in red

In this first scenario, the robot avoids all obstacles and successfully reaches the final position without any issues. In the following second scenario, the robot encounters a specific obstacle, which is non-convex (the usefulness of this obstacle will be explained in the next chapter). This time, the initial and final positions of the robot will be chosen to be offset relative to the obstacle.

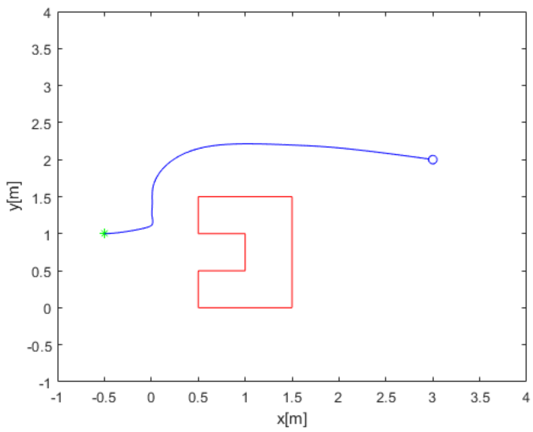

Figure 9: APF with U-shape obstacle in red. Initial position is the green star and final position is blue circle.

It can be noted that the robot once again does its job by avoiding the U-shaped obstacle. Unfortunately, the APF has its limitations in certain situations, which will be examined in the following section.

3.4 Local minima problem

One of the major issues with the APF method is the local minima problem. This phenomenon occurs when the artificial forces cancel each other out, resulting in the robot not being affected by these forces because the sum of its forces equals 0 ([PL03],[MM20],[FA15]). Consequently, the robot becomes trapped in a local minima and can not move. Here are two examples of local minima:

Figure 10: Local minima examples

In the first case, for an U-shaped obstacle, the robot starts to move due to the Artificial potential field, the robot is aiming to the goal position due to the attractive potential, when it gets close to the U-shape obstacle, the repulsive potential starts to influence the pathing of the robot. The U-shape obstacle has two arms that are at equal distance from the center of the U-shape. If the robot is pathing at equal distance from the two arms and is close to the obstacle, the two arms will apply a repulsive force that has the same intensity but different directions and the resulting force will be equal to 0 so the robot will no longer move. The principle is the same for the two circles. Here is an example of a simulation where the robot gets trapped in a local minima:

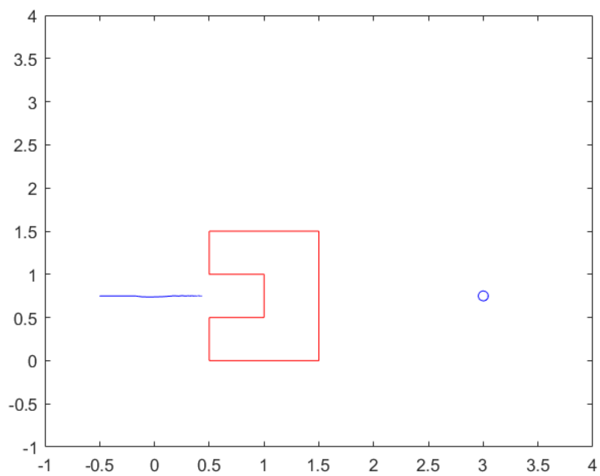

Figure 11: Robot get stuck in local minima for U-shape obstacle

4 Global Navigation fields

In this chapter, various methods will be explored to overcome the local minima problem encountered with the artificial potential field. To begin, an explanation of a numerical method of navigation field will be provided: the wavefront planner, which is a grid-based method. Next, the general idea of the global navigation field will be discussied and modelled. Once this is done, a first example of a global navigation field function will be discussed: the sphere-space navigation functions, along with their advantages and disadvantages. Then, another example more suited to the reality of a construction site environment will be addressed, along with the accompanying mathematical details.

4.1 Wave-Front planner

The Wave-Front planner is a method ([Bud],[Sap],[CL05]) that counters the local minima problem. It is based on the notion that the robot's working environment (construction site in our case) is seen as a grid with squares. The squares in this grid are either 0 or 1 to define the cells where there is an obstacle, making them inaccessible to the robot, or free for the robot to move onto. The final objective for the robot is marked by the number 2. Initially, all the 0 squares around the objective take the value 3, all the 0 squares around the squares with a value of 3 take the value 4, and so on until reaching the robot's starting square on the grid. Once this procedure is complete, the planner will begin to analyze the grid square by square as the robot progresses to construct the path to the objective square. For example, if the robot starts at square 20 and finds 3 squares around it with a value of 19, the planner simply chooses to continue onto one of these squares when there are multiple choices available. The planner continues to move the robot forward until it reaches the final square marked with a value of 2. The discretization of the work environment into a grid and the continuity of the distance function ensure that there will always be a square with a lower value than the one the robot is currently on. Therefore, the robot will never be stuck and can advance square by square, decreasing the value of the square each time until reaching the final square marked with a value of 2.

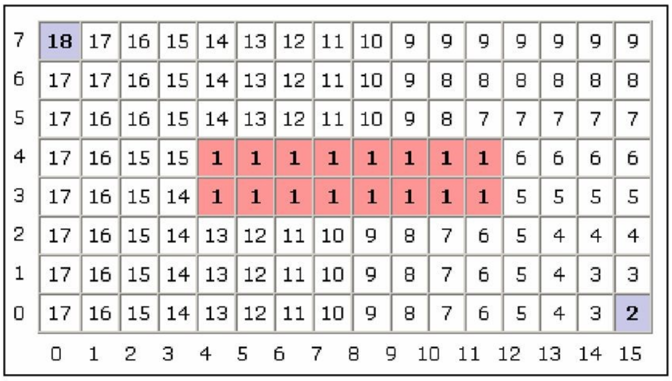

Figure 12: Wave front planner example [Bud]

The major advantages of the Wave-Front planner method include avoiding the local minima problem encountered in the APF. The major disadvantages of the method include being based on a grid with 0 or 1 values for obstacles or free space, and the fact that the planner chooses the fastest path, which may require the robot to get relatively close to obstacles. This poses a danger to the robot that could be avoided and will be addressed in the following sections. Another major disadvantage is that the planner needs to analyze the entire space to discretize it into a grid and find the shortest path. Consequently, the planner cannot account for dynamic obstacles. The planner would need to constantly update the grid with moving obstacles at very small time intervals, which is not computationally efficient for practical settings.

4.2 Navigation Potential Function

In the previous section, the Wave-Front planner was discussed, but it suffers from an exponential calculation problem and requires a significant amount of time, making it inefficient. The method investigated in this chapter addresses this issue while retaining the advantage of the Wave-Front planner, which is avoiding the local minima problem. The general idea of the global navigation field is to find a global function. The potential is a function of the distance from the robot to obstacles. This function is applied to a specific class of space configurations (the robot's working environment), two examples of which will be studied in this chapter.

Mathematically, this function is defined as $\varphi : Q_{free} \rightarrow [0,1]$. This function must be continuous and differentiable to order 2 or higher, have a minimum at the final position, be maximal at the boundaries of the space (the robot's working environment), and finally, be a Morse function. A

Morse function is a function whose critical points are non-degenerate [CL05]. Now that the navigation potential function is defined, two examples of space configurations will be studied and discussed along with their advantages and disadvantages.

4.3 Sphere-space

The first space configuration([FK11],[CL05],[RK92]) is one where the space is bounded by a large sphere (circle in 2D) centered at a point q_0 in space. The robot can only move within this large sphere. The obstacles are defined as spheres (circles in 2D) inside the large sphere, each centered at a point $q_1, q_2, ..., q_n$. Since the navigation potential is based on a distance function of the robot to obstacles, these distances need to be defined. There are two theoretical distances to define: the distance from the robot to the boundaries of the large sphere that delimits the space and the distance between the robot and the obstacles.

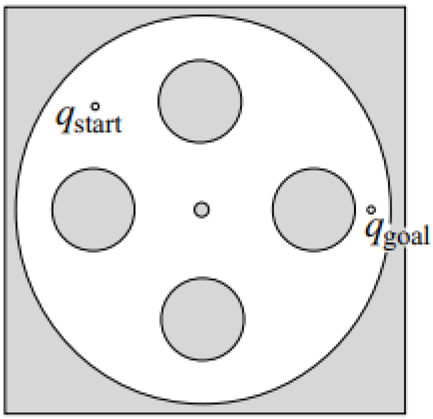

Figure 13: Sphere-space configuration [CL05]

The distance between the robot's position and the large sphere is defined as follows:

$$D_0(q) = -d^2(q, q_0) + r_0^2, \tag{4.1}$$

where $d(q, q_0)$ is the distance between the position of the robot's center and the position of the center of the large sphere. In addition, we consider also the radius of the large sphere r_0. D_0 decreases as the robot moves away from the boundaries of the space.

For the distance between the robot's position and the obstacles, it is the same formula, but now it is the positions $q_1, q_2, ..., q_n$ and their respective radii that are taken into account:

$$D_i(q) = d^2(q, q_i) - r_i^2, \tag{4.2}$$

where i is the index of obstacle i. D_i increases as the robot's position q moves away from obstacle i.

Now that these distances are obtained, to create this navigation potential function, it is necessary to create functions that satisfy the necessary conditions so that the final function obtained after assembling all these functions is a navigation potential function φ.

4.3.1 Modelization

Firstly, to calculate this navigation potential function, we need a distance function. The two theoretical distances calculated above will be used for this. Instead of taking individual distances from obstacles and the large sphere, a global function needs to be found as mentioned in section 4.2, which takes into account all obstacles and the large sphere at the same time. For this, the distance function will be defined as a product of all individual distances:

$$D(q) = \prod_{i=0}^{n} D_i(q). \tag{4.3}$$

The equation 4.3 is zero as soon as the robot reaches the boundary of the large sphere or an obstacle because the distance D_0 or D_i will be zero and therefore cancel out the contributions of other distances. This function thus acts as a repulsive function.

The second function to define will determine whether we have reached the final goal or not. It is defined as follows:

$$\gamma_\kappa(q) = d(q, q_{goal})^{2\kappa}. \tag{4.4}$$

It can be seen that the function increases with an exponent of 2κ as the distance between the robot and the goal increases. κ is a parameter to be tuned which will be discussed later. The function $\frac{\gamma_\kappa(q)}{\beta(q)}$, which is a division between the two functions obtained above, equals zero only when the robot has reached the goal position, and it goes to infinity as the robot's position q approaches the boundary of any obstacle.

It is necessary to create a function that will limit this function $\frac{\gamma_\kappa(q)}{\beta(q)}$ to avoid calculation problems with limits that tend to infinity. For this purpose, a function is defined as follows to address the issue:

$$s(q, \lambda) = \left(\frac{\gamma_\kappa}{\gamma_\kappa + \lambda\beta}\right)(q). \tag{4.5}$$

Using $s(q, \lambda)$, we limit the function $\frac{\gamma_\kappa(q)}{\beta(q)}$. It equals 0 when the robot reaches the goal position and has a value of 1 when it is at the boundary of the large sphere of the space or at the boundary of an obstacle.

One of the conditions for the created function to be a navigation potential function is that it must also be a Morse function; however, this is not the case as there may still exist critical points that are degenerate with the function. To address this issue, another function must be added, which is of the following form:

$$\xi_\kappa(q) = x^{\frac{1}{\kappa}}. \tag{4.6}$$

Finally, by combining this function $\xi_\kappa(q)$ with the function $s(q,\lambda)$, the navigation potential function φ is obtained with $\lambda = 1$:

$$\varphi(q) = (\xi_\kappa \circ s)(q) = \frac{d^2(q, q_{goal})}{[d(q, q_{goal})^{2\kappa} + \beta(q)]^{\frac{1}{\kappa}}}. \tag{4.7}$$

This navigation potential function $\varphi(q)$ ensures having only one minimum when the robot reaches q_{goal} for a sufficiently large κ. In this graph, the sphere space can be observed for different values of κ.

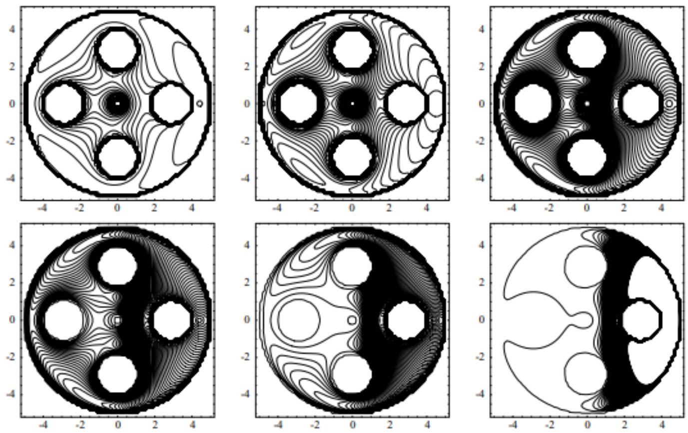

Figure 14: Sphere-space configuration for different k = 3,4,6,7,8,10 respectively [CL05]

It can be seen on Figure 14 how changing the potential function affects the way critical points move towards the goal and how local minima transform into saddles. However, this change has a downside. This navigation function appears smooth near the goal and far from it, but it has abrupt changes in between.

Now that the potential function is obtained, we need to calculate the gradient of this potential([SR16]), just like with the APF, to determine the force that will be exerted on the robot. Here is the calculation:

$$\nabla\varphi(q) = \frac{2d(q,q_{goal})\nabla d(q,q_{goal})[d(q,q_{goal})^{2\kappa}+\beta(q)]^{\frac{1}{\kappa}} - d^2(q,q_{goal})\nabla[d(q,q_{goal})^{2\kappa}+\beta(q)]^{\frac{1}{\kappa}}}{[d(q,q_{goal})^{2\kappa}+\beta(q)]^{\frac{2}{\kappa}}} \tag{4.8}$$

with :

- $\nabla d(q,q_{goal}) = \frac{q-q_{goal}}{d(q,q_{goal})}$;
- $\nabla[d(q,q_{goal})^{2\kappa}+\beta(q)]^{\frac{1}{\kappa}} = \frac{1}{\kappa}[d(q,q_{goal})^{2\kappa}+\beta(q)]^{\kappa}[2\kappa d(q,q_{goal})^{2\kappa-1}\nabla d(q,q_{goal})+\beta(q)]$.

Finally, by substituting the two expressions above into equation 4.8, the final formula for the gradient of the navigation potential function is obtained:

$$\nabla\varphi(q) = \frac{2d(q,q_{goal})(q-q_{goal})[d(q,q_{goal})^{2\kappa}+\beta(q)]^{\frac{1}{\kappa}} - d^2(q,q_{goal})\frac{1}{\kappa}[d(q,q_{goal})^{2\kappa}+\beta(q)]^{\kappa}[2\kappa d(q,q_{goal})^{2\kappa-2}(q-q_{goal})+\nabla\beta(q)]}{[d(q,q_{goal})^{2\kappa}+\beta(q)]^{\frac{2}{\kappa}}} \tag{4.9}$$

Expression 4.9 represents the force that will be applied to the robot to move it within the sphere space.

4.3.2 Simulation

The simulation involves creating a spherical environment with a large sphere that the robot cannot exit and spherical obstacles. Three circular obstacles of different radius and positions were created. The robot starts from an initial position and must reach the goal position. To reach this position, it is subjected to the navigation potential function 4.9 (calculated for a sphere space) that will guide it to the target. Here is a simulation on MATLAB of the sphere space with 3 obstacles :

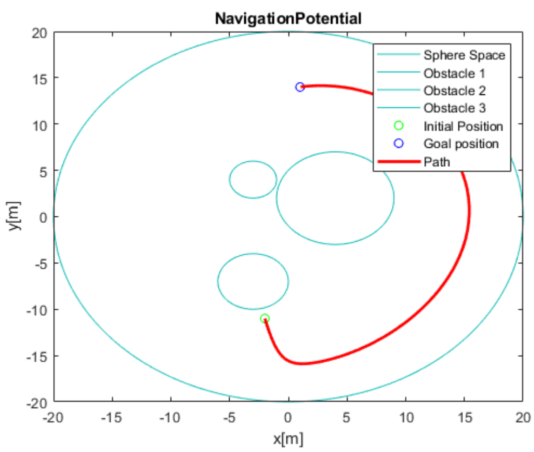

Figure 15: Navigation potential function applied on robot for path planning with 3 obstacles. Goal position is after the 3 obstacles.

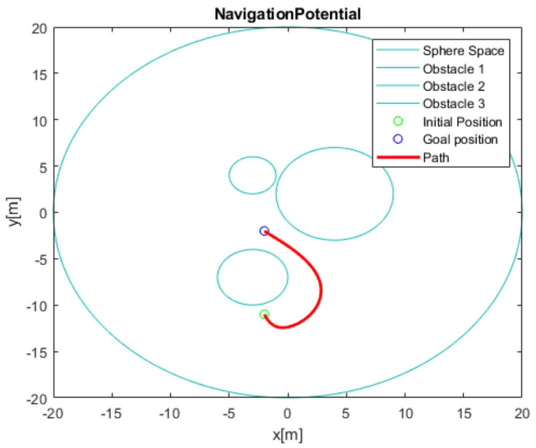

Figure 16: Navigation potential function applied on robot for path planning with 3 obstacles. Goal position is between the 3 obstacles.

As we can see in Figure 16, the navigation potential function allow the robot to pass between two circles at equal distance from the path planning of the robot and thus avoiding local minima problem. In comparison, as it can be seen on Figure 17, the robot get stuck in local minima problem (see section 3.4) and then can not move compared to the robot for the navigation potential function.

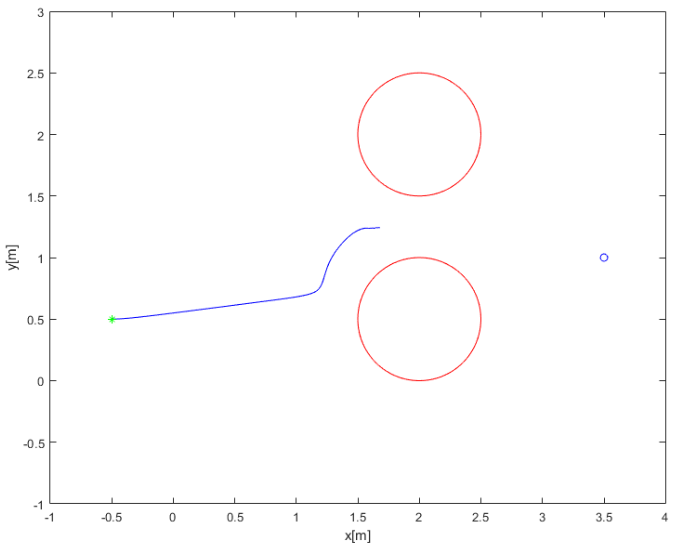

Figure 17: Artificial Potential Field applied on robot for path planning with 2 circles obstacles. The robot get stuck between the 2 circles and thus face the local minima problem. green star is the initial position and the blue circle is the final position.

4.3.3 Comparison between APF and Navigation Potential Function

In this subsection, advantages and disadvantages will be listed for APF and Navigation potential function with a comparison and conclusion at the end.

For the Artificial Potential Field (APF) Method, the advantages are: easy to understand and implement using basic concepts of attraction and repulsion. Quick computations that make it suitable for real-time applications. The disadvantages are: the robot can get stuck in local minima, preventing it from reaching the goal. It requires proper tuning of force coefficients to work effectively. The robot may oscillate near closely spaced obstacles due to conflicting forces.

For the Navigation Potential Function Method, the advantages are: designed to avoid local minima by using a global navigation function. It handles complex and realistic environments better, especially with diffeomorphism. It is adaptable to different spaces and obstacles, offering more flexibility. The disadvantages are: more challenging to understand and implement compared to APF, with specific definitions and conditions. The method can face numerical problems, especially with sharp transitions in the potential function. It requires complex calculations and transformations, making it demanding on resources. Finally, it needs careful tuning of parameters to perform effectively.

APF focuses on local information and can get stuck in local minima, while the navigation potential function provides a global solution to avoid this issue. APF is simpler and faster to implement but less effective in complex environments. The navigation potential function is more robust in complex scenarios but harder to implement. APF is better for real-time applications due to lower computational demands, while the navigation potential function might face challenges in real-time due to its computational complexity. The navigation potential function method is more adaptable to different environments and obstacle configurations than APF. These two methods can also be used with robot arm or mobile manipulators as well (they are not limited to just mobile robots). By approximating robot links as several spheres one can use the propreties of this method.

4.4 Star shape space

The sphere-space is a highly general theoretical space because it consists of a simple shape, the sphere (i.e., circle in 2D). However, in reality, the space where the robot operates does not solely contain spherical obstacles. Therefore, it is necessary to define other spaces that better resemble reality. The sphere-space can be used as a model space to create other space configurations. This is possible through the mathematical concept of diffeomorphism([Enc]). Essentially, the sphere-space can serve as a model space for another space configuration as long as that configuration is diffeomorphic to the sphere-space. Thus, it is essential to find the diffeomorphism between the two configurations, which acts as a bridge between the spaces.

Consider two space configurations, A and B, where A represents the sphere-space configuration and B represents the star-shape space ([RK92]) configuration. There exists a navigation function $\varphi : A \to [0,1]$ for the sphere-space configuration. Then, there is a mapping $h : B \to A$ that serves as a diffeomorphism linking the two space configurations([CL05]). With this, it is possible to obtain the navigation function for the star-shape space configuration. Specifically, $\varsigma = \varphi \circ h$, where ς is the navigation function for the star-shape space configuration B. This diffeomorphism facilitates a correspondence between critical points in the two space configurations, enabling the definition of the navigation function ς for the star-shape space based on that of the sphere-space φ.

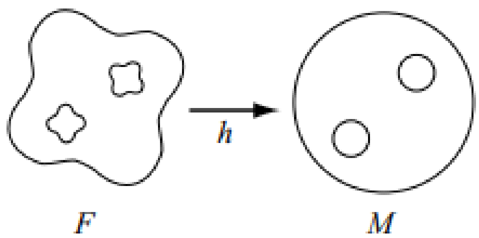

Figure 18: Diffeomorphism between star-shape configuration and sphere-space configuration [CL05]

Here is an example of a mapping between star-space and sphere-space [CL05] :

$$h_\lambda(q) = s_{q_{goal}}(q,\lambda)q + \sum_{i=0}^{M} s_i(q,\lambda)T_i(q), \tag{4.10}$$

where :

- $T_i(q)$ is a translated scaling map. It is necessary to create the h mapping.
- $s_{q_{goal}}(q,\lambda)$ is a switch for the goal which is one at the goal and zero on the boundary of the

free space.

- $s_i(q, \lambda)$ is a switch for the obstacles and the boundary. It is equal to one on the boundary and zero on the goal and the obstacles.

Although further research and studies are needed, this method could potentially address some of the issues associated with artificial potential fields.

5 Conclusion

In conclusion, to enable the movement of the robot in a workspace such as a construction site with various obstacles along its path, several steps were taken. Firstly, the Kuka Youbot was chosen for the task due to its specialized wheels called omniwheels, which possess many favorable characteristics for efficient movement in a construction site. Next, its kinematics and dynamics were studied to facilitate its movement. Once this was accomplished, the obstacles had to be accounted for so that the robot could avoid them. The Artificial Potential Field (APF) was modeled, imagining the obstacles as repulsive and the objective as attractive to the robot. This allowed the robot to experience a force that guided it to avoid obstacles while being drawn towards the objective. However, the APF suffers from a major issue known as local minima. A solution was devised to address this problem: the global navigation field. This approach seeks to devise a global navigation function to overcome the local problems that can arise with the APF method. Within the global navigation field framework, the wave-front planner method was theoretically discussed. It involves discretizing the construction site into a grid with 0 and 1 representing obstacles and free spaces. However, this method requires extensive calculations and time, making it inefficient. Instead, a global navigation function was conceived. Initially, a sphere-space workspace was considered, containing only spheres for a simplified study. However, in practical scenarios, encountered obstacles may not necessarily be spherical, prompting an investigation into other types of spaces. The star-shape space consists only of star-shaped obstacles. To transition between space configurations, the mathematical concept of diffeomorphism was employed. Star-shape spaces represent special cases within normal sets.

Further work on this project is to investigate in details of star shaped world to model various configurations and solve other problems of path planning. Also using Explicit Reference Governor (ERG), as it is similar to the APF. We can explore if these aspects could be used to solve issues with local minimas also with ERG constrained control scheme.

Bibliography

[A N22] A. Nasuha A. S. Priambodo, G. N. P. Pratama (2022). "Vortex Artificial Potential Field for Mobile Robot Path Planning". In: *Journal of Physics: Conference Series*.

[ak17] ak, Stanislaw H. Z (2017). *An Introduction to Model-based Predictive Control (MPC)*. Tech. rep. ResearchGate.

[Aa] Alshorman, Ahmad and al. (n.d.). *Fuzzy-Based Fault Tolerant Control for Omni-directional Mobile Robot*. URL: `https://www.researchgate.net/publication/344151914_Fuzzy-Based_Fault_Tolerant_Control_for_Omni-directional_Mobile_Robot`. (accessed: 02.12.23).

[Ba03] Bekris, Kostas E. and al. (2003). *Multiple Query Probabilistic Roadmap Planning using Single Query Planning Primitives*. Tech. rep. Rice University.

[Ben14] Bennett, Kelly (2014). "MATLAB Applications for the Practical Engineer". In: InTech. Chap. 3.

[BP17] Brian Ichter, James Harrison and Marco Pavone (2017). *Learning Sampling Distributions for Robot Motion Planning*. Tech. rep. Stanford University.

[Bud] Budhathoki, Ram Kaji (n.d.). *Project on Implementation of Wave Front Planner Algorithm*. URL: `https://slideplayer.com/slide/4710705/`. (accessed: 17.05.24).

[Che11] Chen, Siyang (2011). *The A* Search Algorithm*. Tech. rep. Duke university.

[CL05] Choset, Howie and Kevin Lynchn (2005). "Principles of Robot Motion". In: MIT Press.

[dev] developers, Kuka youBot (n.d.). *KUKA YouBot Kinematics, Dynamics And 3D Model*. URL: `http://www.youbot-store.com/developers/kuka-youbot-kinematics-dynamics-and-3d-model`. (accessed: 21.11.23).

[Enc] Encyclopedia of Mathematics (n.d.). *Diffeomorphism*. URL: `https://encyclopediaofmath.org/index.php?title=Diffeomorphism&oldid=53428`. (accessed: 20.04.24).

[FA15] Fethi Matoui, B. Boussaid and M. Abdelkrim (2015). “Local minimum solution for the potential field method in multiple robot motion planning task”. In: *IEEE*.

[FK11] Filippidis, Ioannis and Kostas J. Kyriakopoulos (2011). “Adjustable navigation functions for unknown sphere worlds”. In: *ResearchGate*.

[Gev19] Gevargiz, Patrick (2019). *Control and Path Planning of Multi-Agent Systems: Artificial Potential Field Algorithm*.

[HR17] Hendzel, Z. and Ł. Rykala (2017). “Modelling of Dynamics of a Wheeled Mobile Robot With Mecanum Wheels With the Use of Lagrange Equations of the Second Kind”. In: *Int. J. of Applied Mechanics and Engineering*.

[Ka11] Kalakrishnan, Mrinal and al. (2011). *STOMP: Stochastic Trajectory Optimization for Motion Planning*. Tech. rep. University of Southern California and Willow Garage.

[Ka96] Kavraki, Lydia E. and al. (1996). “Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces”. In: *IEEE Transactions on Robotics and Automation*.

[Kha86] Khatib, Oussama (1986). “Real-time Obstacle Avoidance for Manipulators and Mobile Robots”. In: *The International Journal of Robotics Research*.

[LaVnd] LaValle, Steven M. (n.d.). *Rapidly-Exploring Random Trees : A New Tool for Path Planning*. Tech. rep. Iowa State University.

[Ma] Maddahi, Yaser and al. (n.d.). *Calibration of omnidirectional wheeled mobile robots: method andexperiments*. URL: `https://www.researchgate.net/publication/259434997_Calibration_of_omnidirectional_wheeled_mobile_robots_Method_and_experiments`. (accessed: 25.11.23).

[MM20] Mohammed Isam Ismael Abdi Muhammad Umer Khan, Ahmet Günes and Deepti Mishra (2020). “Escaping Local Minima in Path Planning Using a Robust Bacterial Foraging Algorithm”. In: *MDPI*.

[PL03] Park, Min Gyu and Min Cheol Lee (2003). “A new technique to escape local minimum in artificial potential field based path planning”. In: *KSME International Journal*.

[PP23] Peter Corke, Witold Jachimczyk and Remo Pillat (2023). “Robotics, Vision and Control (Third Edition)”. In: Springer.

[RK92] Rimon, Elon and Daniel E. Koditschek (1992). “Exact Robot Navigation Using Artificial Potential Functions”. In: *IEEE Transactions on Robotics and Automation*.

[Roba] Robotics, Clearpath (n.d.[a]). *HUSKY Unmanned Ground Vehicle*. URL: `https://clearpathroboti` `com/husky-unmanned-ground-vehicle-robot/`. (accessed: 20.05.24).

[Robb] Robotics, Coppelia (n.d.[b]). *Coppelia Robotics*. URL: `https://coppeliarobotics.com/`. (accessed: 10.05.24).

[Rym] Rymsha Siddiqui (n.d.). *Path Planning Using Potential Field Algorithm.* URL: `https://medium.com/@rymshasiddiqui/path-planning-using-potential-field-algorithm-a30ad12bdb08`. (accessed: 11.12.23).

[Sa21] Sánchez-Ibáñez, José Ricardo and al. (2021). "Path Planning for Autonomous Mobile Robots: A Review". In: *MDPI.*

[SR16] Santiago Paternain, Daniel E. Koditschek and Alejandro Ribeiro (2016). "Navigation Functions for Convex Potentials in a Space with Convex Obstacles". In: *University of Pennsylvania.*

[Sap] Saptarshi Dey (n.d.). *Shortest Path Planning using wavefront Propagation.* URL: `https://dev.to/darkmortal/shortest-path-planning-using-wavefront-propagation-4pom`. (accessed: 10.04.24).

[Sa17] Schulman, John and al. (2017). *Proximal Policy Optimization Algorithms.* Tech. rep. OpenAI.

[Sew19] Sewak, Mohit (2019). "Deep Q Network (DQN), Double DQN, and Dueling DQN: A Step Towards General Artificial Intelligence". In: *ResearchGate.*

[Sa] Sharma, Shashank and al. (n.d.). *Unified Closed Form Inverse Kinematics for the KUKA youBot.* URL: `https://www.researchgate.net/publication/261074766_Unified_Closed_Form_Inverse_Kinematics_for_the_KUKA_youBot`. (accessed: 25.11.23).

[Sic08] Siciliano, Bruno (2008). "Robotics: Modelling, Planning and Control". In: Springer.

[Ste94] Stentz, Anthony (1994). *The D* Algorithm for Real-Time Planning of Optimal Traverses.* Tech. rep. Carnegie Mellon University.

[TN15] Tharindu Weerakoon, Kazuo Ishii and Amir Ali Forough Nassiraei (2015). "An Artificial Potential Field based mobile robot navigation method to prevent from deadlock". In: *JAISCR.*

[Ven14] Venkat, Ravikiran Jaliparthi (2014). *PATH FINDING- Dijkstra's Algorithm.* Tech. rep. Indiana State University.

[XL20] Xiaojing Fan, Yinjing Guo and Hui Liu (2020). "Improved Artificial Potential Field Method Applied for AUV Path Planning". In: *Hindawi.*

[YD19] Yunwang Li, Shirong Ge and Sumei Dai (2019). "Kinematic Modeling of a Combined System of Multiple Mecanum-Wheeled Robots with Velocity Compensation". In: *MDPI.*

[Za21] Zeng, Wenyuan and al. (2021). "End-to-end Interpretable Neural Motion Planner". In: *ResearchGate.*

[Za13] Zucker, Matt and al. (2013). *CHOMP: Covariant Hamiltonian Optimization for Motion Planning.* Tech. rep. The Robotics Institute Carnegie Mellon University.