République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid
Faculté des Sciences et de la Technologie
Département d'Informatique

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

Master Thesis in ComputerScience

Presented by

Tridi Samira

Specialty: Intelligent Computer Systems

Title

Trajectory planning of a mobile robot based on Artificial Potential Fields

Defence on: 04/06/2021

Jury composed of:

Quality	**Name and Surname**	**Degree**	**University**
President	Ahmed malek N	MAA	Chadli Bendjedid El-Tarf
Rapporteur	Benmachiche A	MCA	**Chadli Bendjedid El-Tarf**
Examiner	Bouguerne I	MCB	Chadli Bendjedid El-Tarf

University Year: 2020/2021

Thanks

My thanks go first to **G**od **A**lmighty for the will, health and patience he gave me duringall these years of study.

In particular, **I** would like to thank **M**r. Benmachiche.**A**, **S**enior **L**ecturer in the **D**epartment of **C**omputer at **C**hadlien **D**jedid **U**niversity, who has mentored me throughout this thesis, for his valuable advice and support. **I** am grateful for his availability and the trust he has placed in me. **M**y gratitude also to the team of teachersof the **O**ption, **Artificial I**ntelligence.

I sincerely thank:

-Dr-Bouguerne I

-Dr- Ahmed malek N

Thank you so much to my entire family for their care and concern for me, as well

I dedicate

work to:

my father

To my Mother,

To the whole family

And to All my friends.

Contents

Liste des figures

List of tables

List of acronyms

- **AI** : Artificial intelligence
- **APF**:Artificial Potential Fields
- ANN :Artificial Neural Network
- **(CD)** : Cell Decomposition approach
- **GA** : Genetic Algorrithm
- **VBA** : Virtual Bee Algorithm
- **BCO** : Bee Colony Optimization
- **BOD** : Dance Bee Optimization
- **ABC** : Artificial Bee Colony
- **(PSO)** : Particle Swarm Optimization (PSO)

General Introduction

Artificial Intelligence (AI) is an area of computer science in which machines are taught and constructed to emulate human decision-making and reaction functions without the need for human interaction.

The application of a number of approaches aimed at allowing robots to emulate a kind of actual intelligence is known as artificial intelligence (AI). Artificial intelligence is becoming increasingly important in all fields. The mathematician Alan Turing in 1950 was the first to tackle the notion of artificial intelligence, with experiments to determine whether a machine could be considered "conscious". He then describes a test known today as the "Turing Test"

In 1956, American scientist Marvin Lee Minsky defined artificial intelligence as "the construction of computer programs, performed more satisfactorily by human beings."

Intelligence makes it possible to acquire and retain knowledge like a human brain, but artificial. Artificial intelligence is defined as "the set of techniques aimed at enabling machines to imitate a form of real intelligence" [1]. We have several definitions of Artificial Intelligence.

Artificial Intelligence (AI) is the process of human intelligence by machines, more particularly, computer systems. These processes consist of three phases. First of all, learning, that is, the acquisition of information and its rules for using it. Then, reasoning, which is the use of rules to draw rough or definitive conclusions.

Artificial intelligence: the ability, imparted by humans to machines, to memorize and learn from experience, to think and create, to speak, to judge, and to decide The usefulness of artificial intelligence in robotics is to make machines more autonomous and even allows performing robotic tasks.

Artificial intelligence (AI) refers to a machine's or computer's ability to mimic the functions of the human mind. AI uses a variety of technologies to provide robots human-like intelligence in planning, acting, comprehending, learning, and sensing. AI systems are capable of perceiving their surroundings, recognizing objects, making decisions, solving problems, learning from experience, and imitating examples. These skills are combined to perform tasks that would normally need humans, like as driving a car or greeting a visitor.

Artificial intelligence states that intelligent systems' actions are governed by principles. It is based on transferring human abilities and characteristics to a machine. The system employs computer power to perform tasks that are beyond the capabilities of the normal human. The machine will have to learn how to respond to specific actions. A propensity model is built using historical data and algorithms. Machines can learn from their mistakes to do cognitive tasks often reserved for the human brain. The system automatically learns from the data's features and trends.

Engineering and cognitive science are the two pillars on which AI is built. Engineering entails creating instruments that are comparable to human intelligence. Large data sets are paired with a set of instructions (algorithms) and iterative processing. Machine learning, deep learning, neural networks, cognitive computing, computer vision, natural language processing, and knowledge reasoning are some of the topics that cognitive science provides to AI.

Artificial intelligence a broad term that encompasses a variety of systems, it's a broad field. Weather applications, business data analysis apps, taxi hailing apps, and digital assistants are examples of rudimentary, low-level AI systems focused on completing a specific task. This is the form of AI that the average person is most likely to engage with, known as "Narrow AI." Their primary goal is to increase efficiency. Advanced systems, on the other hand, can do difficult tasks by emulating human intellect on a more general level. These include creative, abstract, and strategic thinking. For the time being, this type of completely sentient machine, known as "Artificial General Intelligence" o, exists only on the silver screen, yet the race to realize it is accelerating.

Robotics is a relatively new science. Its first applications date back to the 1960s with household robots. Like many complex disciplines, it benefits from contributions from several other sciences.

The problem addressed in this document is that of "navigation in an environment, the resolution of this problem, by effective methods of movement planning. One of his main contributions was introduced, which allows us to solve problems: called artificial potential field (APF). The idea is defined by the sum of the attractive potential that pulls the robot towards objectives and the repulsive potential pushes the robot away from obstacles.

The first objective of robotics can be deduced from the definition which assimilates it to a set of techniques allowing the manufacture of robots. The usefulness thus attributed to it consists in the provision of assistance capable of taking charge of difficult, dangerous, or repetitive tasks which are incumbent on humans.

Navigation is a quantum difference in perceptual capacity [2], [3]. Man can detect, classify and identify environmental characteristics under different environmental conditions, navigation strategies allowing a mobile robot to move to achieve its objective.

In this context, we call movement planning the problem of the preliminary calculation of the movements necessary for a robot to accomplish a given task.

Motion planning, also called autonomous navigation, is defined as motion planning focused on the specific case of robots.

The document is composed of three (03) chapters:

The first chapter is devoted to the presentation of mobile robots, a general overview of the field of mobile robotics.

In the second, the contribution to the research problem, we present a study on the search for the path for a robot modeling of the environment with the APF grid method. We will discuss the application of a planner for one mobile robot to accomplish a specific task i.e. goal convergence and obstacle avoidance.

In the last chapter, simulation environments are briefly discussed. Several experiments are carried out with a MATLAB development environment to evaluate the performance of APF, examples of simulations are provided to demonstrate the results of the proposed methods for finding the path of a robot.

Finally, we conclude and summarize the entire thesis work, explaining the conclusion obtained through experiments.

Chapter 1: State of the Artt

1.1: Introduction:

The word robot is often used for many devices that have the ability to work automatically or can be controlled remotely. A robot is defined as a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks [4].

A mobile robot is a machine controlled by software that use sensors and other technology to identify its surroundings and move around its environment. Mobile robots function using a combination of artificial intelligence (AI) and physical robotic elements, such as wheels, tracks and legs. Mobile robots are becoming increasingly popular across different business sectors. They are used to assist with work processes and even accomplish tasks that are impossible or dangerous for human workers.

The industry of mobile robotics is responsible for the development of these locomotive robot systems. We define the various navigation strategies and provide a list of existing controller categories. The process of answering the following three questions [5] is referred to as navigation.

i) What am I doing here?

The process of determining where the robot is in relation to its surroundings is known as localization. In order to make effective decisions, the robot must first understand its own position. The answer to this question is found by using robot localization.

ii) How far away from me are the other locations?

While moving, the robot must perceive its surroundings and identify crucial elements that will allow it to record information about its surroundings. The robot must know where it is heading in order to complete the assignment. It must decide on a goal and a course of action.

iii) How do I get to these other locations?

Once the robot understands where it is and where it needs to go in its working space, it must decide how to get there while adhering to the limits imposed by the constraints, such as avoiding obstacles. Path planning is the process of determining how to arrive to the desired location.

The robot navigation in a dynamic environment is to move from the start position to the arrival position. Planning the motion of the robot requires the generation of an executable trajectory. In robotics, we often have to plan trajectories to allow a mobile robot to move from an initial point to an arrival point. Trajectory planning is a subject that is often dealt with in the scientific field. There are therefore quite a few different algorithms for performing such a task.

Figure 1-1 Robot P3AT

Figure 1-2 Legged robot - Lauron III; Institute of Robòtica I Informàtica Industrial

Figure 1-3 R2K5 - Walking Machine; Superior Technology School technologie

Figure 1-4 Ruko Smart Robot

Figure 1.1:Examples of Decent Robots

1.2: Stat of the art:

We use here a classification established, which has the advantage of distinguishing strategies with and without internal models.

1.2.1-Fuzzy Logic:

Binary logic has the virtue of being simple, but it is a long way from human thought. In the case of determining the closeness of an obstacle, fuzzy logic allows for consideration of concepts such as "near enough" or "extremely far," rather than being limited to a binary definition of "obstacle or no obstacle." Zadeh [06] formalized this in 1965.

The fuzzy logic controller's principle is divided into three stages:

- A fuzzification stage, in which the input variables are transformed into fuzzy variables;
- A stage in which logical rules of the kind "if (condition 1) and/or (condition 2) then (action on the outputs)" are applied to a table of behavior rules;
- Finally, the de fuzzification stage turns the behavior dictated by command rules into action. The input variables' space is delimited in a certain number of fuzzy subsets by fuzzy intervals (for example, for proximity, we can have very close (contact), quite close, average distance, far enough, and very far); membership functions are then used to define the degree of truth (probability of belonging) of the fuzzy variable as a function of the input quantifier.

These functions can be a triangle, Gaussian, etc. Thus for given distance measurement, the membership rule will tell us "there is a 95%. Percent chance that the obstacle is close enough, 5%percent chance of being in contact". This notion is based on the fact that there are always uncertainties about the sensor measurements and the information available in general.

The second step is the development of rules of behavior for the robot, following the combination of fuzzy input variables. The behavior rules table is created by hand and is based on the experience of the person who will adjust the controller. A rule for a mobile robot could be "if there is a fairly close obstacle on the right, you must turn left and reduce the robot's speed".

The final step is to command the robot to transform the behavior derived from the rules table. The centers of gravity method, which involves taking a weighted average of a orders to be applied, is one method that can be used to accomplish this. The probability of membership of each input variable is related to the weighting [07].

This controller allows the robot to reach its final position while obeying the robot's kinematic constraints; however it does not take into consideration impediments for the time being. The consideration of the vehicle's final orientation is currently being integrated. The authors of [08] use the right/left symmetry of the robot's logic rules of behavior to simplify them and thus reduce the calculation time.

The problem with these fuzzy logic methods is similar to that of the APF methods, namely the problem of local minima, which allows the robot to remain caught in dead ends. In his paper [09], Xu proposes a method for escaping these traps.

This technique uses virtual targets to remove the robot from the trap into which it has fallen as soon as it realizes it has fallen.

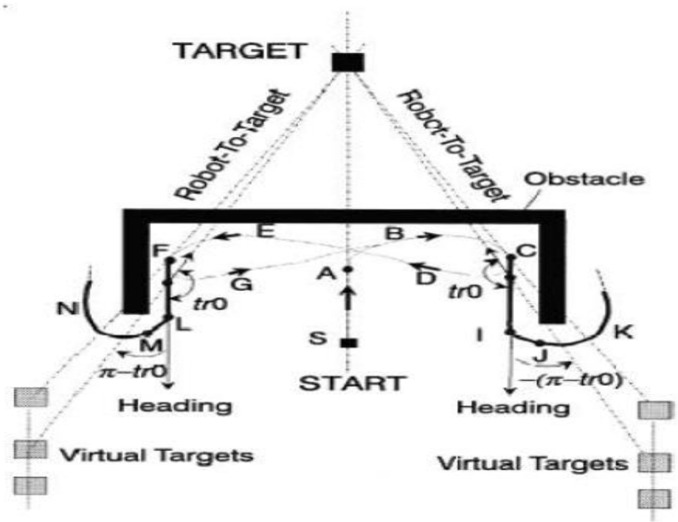

Figure1-5**:** Principle of virtual targets to get out of a local minimum [9].

Another issue with fuzzy logic methods is that they are excessively specialized for a specific sort of environment, and as a result, they have difficulty adapting to different situations. However, as the robot explores a new area, it can use so-called learning methods to adjust its norms of conduct. The issue with these methods is that they take a long time to learn, and it takes much longer for the robot to navigate successfully [10].

1.2.2: Artificial Neural network:

An artificial neural network (ANN) is a computational model to perform tasks like prediction, classification, decision making, etc. It consists of artificial neurons. These artificial neurons are a copy of human brain neurons. Neurons in the brain pass the signals to perform the actions. Similarly, artificial neurons connect in a neural network to perform tasks. The connection between the artificial neurons is called weight.

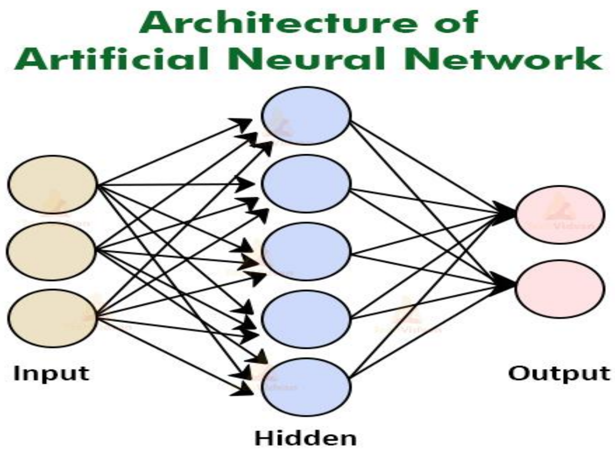

Figure 1-6 Architecture of Artificial Neural Network

The input node takes the information in numerical form. The information represents an activation value where each node has given a number, the higher the number, the greater the activation. Based on weights and activation function, the activation value passes to the next node, each node calculates the weighted sum and updates that sum based on the transfer function (activation function). After that, it applies an activation function. This function applies to this particular neuron. From that, the neuron concludes if it needs to forward the signal or not. ANN decides the signal extension on the adjustments of the weights.

The activation runs through the network until it reaches the output node. The output layer shares the information in an understanding way. The network uses the cost function to compare the output and expected output. Cost function refers to the difference between the actual value and the predicted value. Lower the cost function, closer it is to the desired output.

Path planning for mobile robots ensures a smoot trajectory to the designed target without collision with static or moving obstacles. The aim of this paper is to develop an algorithm using the fuzzy logic controller in order to find a feasible collision-free path with obstacles moving with varying velocities.

The mobile robot needs to find a collision-free path between any two points (from its beginning to its end). To be able to find this path, the mobile robot should run an adequate path planning algorithm. Several research works, for path planning of mobile robots, have been proposed in the literature [11], [12], and [13]. Hence, there is some classical approaches dedicated to static environments are extended to dynamic ones [14].

Hybrid ANN-based approaches in combination with other artificial intelligence-based methodologies such More applicable approaches are fuzzy logic, knowledge-based systems, and evolutionary approaches. In real-world applications, tackling the robot navigation problem [15], [16].

One of the major tasks of path planning of mobile robots has been presented. Based on the developed algorithm, we have found a collision-free path in a cluttered environment containing static and moving obstacles. On the other side, we have used a fuzzy logic controller to reduce the chattering phenomenon in the control law of the original method which uses the sign function. Simulation results prove that the developed algorithm shows high effectiveness in obstacle avoidance with the shortest path to the destination and the lowest elapsed time. Indeed, the high frequency in the control input has been successfully reduced.

The problem of these fuzzy logic methods is generally, the problem of local minima, another problem that comes out of the fuzzy logic methods is that they are too specialized for a given type of environment. There are so-called learning methods that allow the robot to modify its rules of behavior as it explores a new environment.

1.2.3: Ant Colony:

The classical ant colony algorithm has the disadvantages of initial search blindness, slow convergence speed, and ease to fall into local optimum when applied to mobile robot path planning. This paper presents an improved ant colony algorithm in order to solve these disadvantages. First, the A* search algorithm for initial search to generate uneven initial pheromone distribution to solve the initial search blindness problem. At the same time, the algorithm also limits the pheromone concentration to avoid local optimum. Then, the algorithm optimizes the transfer probability and adopts the pheromone update rule of "incentive and suppression strategy" to accelerate the convergence speed. Finally, the algorithm builds an adaptive model of the pheromone coefficient to make the pheromone coefficient adjustment self-adaptive to avoid falling into a local minimum. The results proved that the proposed algorithm is practical and effective.

Ant colony algorithm is widely used as an intelligent optimization algorithm to solve the problem of path planning of mobile robots. Ant colony algorithm has the advantages of high efficiency, strong robustness, and ease to combine with other algorithms, but it still has the defect [17] which is easy to fall into local optimal solution and slow convergence speed.

Therefore, many domestic and foreign scholars began to study the improvement of the ant colony algorithm. Hsu successfully solved the problem of the ant colony algorithm falling into local optimum by adjusting parameters and updating pheromone continuously [18]. But it does not solve the initial search blindness problem. Qu set the pheromone boundary in the algorithm and made some improvements to solve the deadlock problem [19]. But the convergence speed of the algorithm may be slowly. This algorithm, which can accelerate the convergence speed by generating uneven initial pheromone distribution and optimizing probability, and avoid local optimum by building an adaptive model of pheromone coefficient.

Ant colony will easily fall into the local optimum and tend to poor convergence. To this end, many scholars delivered a variety of improved methods to solve problems regarding pheromone update and path search strategy.

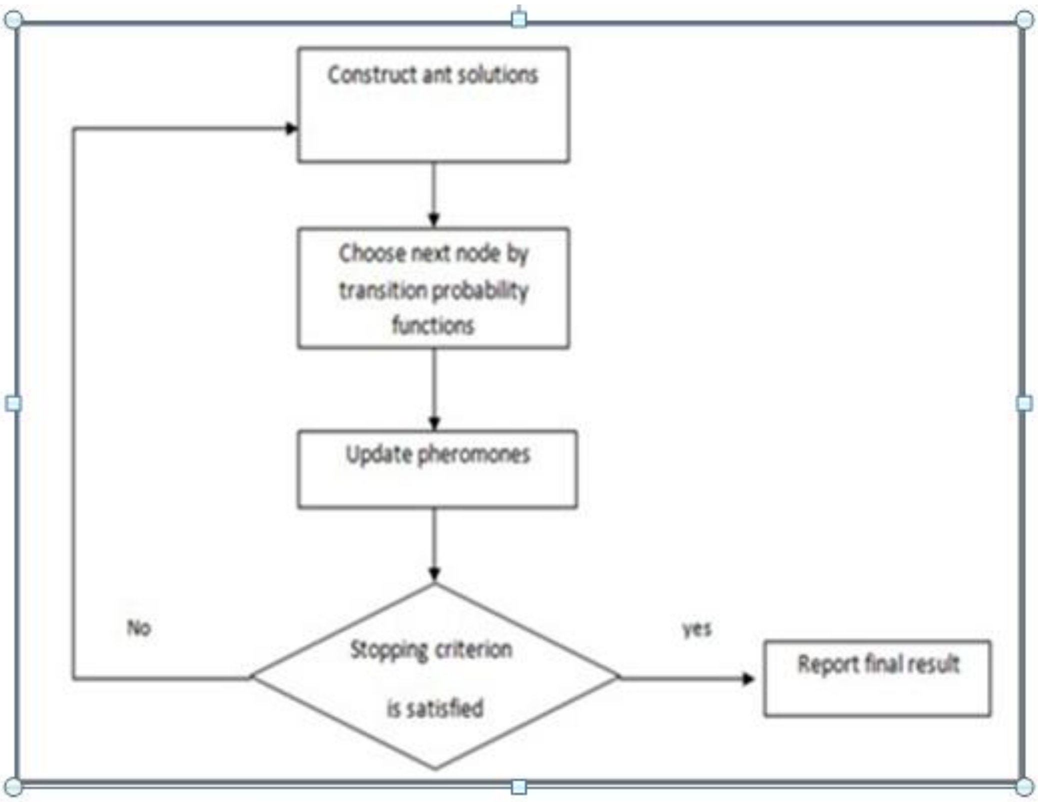

Figure 1.7: Ant Colony

Ant Colony algorithm was proposed to speed up the convergence, on the path of the optimal ant of each generation. Changing the volatilization rate and adjusting the pheromone updating formula, the search ability of the ant colony and the convergence rate of the algorithm was improved, some intelligent algorithm was proposed to generate an initial path, which can be transformed into the initial pheromone distribution to avoid the blind search of the ant colony.

The path information (such as the crowded path and the steep path weight) was added into the initial pheromone matrix, which could affect~~s~~ the efficiency of the algorithm.

Ant Colony could find a shorter path and its convergence was better. In addition, many scholars have combined the ant colony algorithm with other (intelligent) algorithms to improve the convergence rate and the smoothness of the path. The geometric method was used to optimize path, the force factor in the artificial potential field method is transformed into a local diffusion pheromone to improve the ability of the ant colony algorithm to find the obstacle. The fuzzy ant colony optimization method was proposed to minimize the iterative learning error.

1.2.4: Bee Colony:

We can develop a class of new algorithms by only using parts of the nature or behavior of bees and adding some new features. The following sections show some (the most well-known) algorithms based on bee behavior during foraging, however they are not exhaustive.

1.2.5: Virtual Bee Algorithm:

Although only functions with two parameters have been given as examples, this approach was invented by XinShe Yang in 2002 [20], for solving numerical optimization issues. It can optimize functions and discrete issues. The VBA algorithm starts with a virtual bee troop, where each bee moves randomly into the search space, which in most applications is just a 1-D or 2-D grid.

Establishment of a multi-agent or virtual bee population, each bee has a solution vector with a number of parameters to optimize. Optimization functions (objective functions) coding and virtual food conversion (Virtual Food), establishment of a criterion for communicating direction and distance in a manner analogous to bee's physical abilities (the dance of the bees). Update a population of bees in new positions for virtual food study by performing a virtual dance to specify distance and direction; "the virtual dance of waggle."

The maximum mode, in terms of the number of virtual bees or the intensity / frequency of the bees that make the visit, correlates to the best evaluation after a set period of evolution.

1.2.5.1: Bee Colony Optimization

We have found algorithms for difficult combinatorial optimization problems made by yuce et al in 2013 [21] is called Bee Colony Optimization BCO.

There are two alternating steps (step forward and backward), with each step forward, each artificial bee visiting N solutions creates a partial solution and then returns to the hive, we find several solutions, then the best one is determined, and it is used to update the best overall solution and like that, an iteration of BCO is accomplished. At this point, all solutions are removed and a new iteration is born.

BCO can be described as follows:

- Initialization: an empty solution is assigned to each bee;

- For each bee: to. k = 1

a. Count constructive progress)

b. Evaluate all possible steps;

vs. Choose a step;

re. k = k + 1;

If k ≤ NC, go to b.

Return all of the bees to the hive;

- Determine the value of the goal function for each bee.

- Each bee chooses at random whether or not to continue its own investigation.

- Become a recruiter, or become the harvesting bee. Choose a fresh solution from the recruiters for each follower.

- If the solutions aren't complete, move on to the next phase (b). If the stop criterion is not checked, proceed

- To step (b); otherwise, proceed to step (c).

- Display the best solution discovered.

1.2.5.2: Dance Bee Optimization

The BOD (Dance Bee Optimization) algorithm for mobile robot to solve the problem of T coloring graphs we introduce the BOD algorithm (Dance Bee Optimization), for each bee recruited (solution), we associate a neighborhood meta-heuristic, and we have to build a set the best bee dancers. The dance is vertical, the pollen is in the direction of the sun.

Laga and Nouioua created the BOD (Dance Bee Optimization) method in 2009 [22] to handle the problem of T-coloring graphs.

The remaining bees are generated at random to complete the population. The colony will comprise, on the one hand, of m bees representative of each neighborhood (to increase the search) and, on the other hand, of bees assigned randomly at the conclusion for each iteration (to diversify the search).

This process is repeated until a predetermined stopping criterion is met (a number of iterations or a stagnation number). The pollen is towards the sun, therefore the dance is vertical. The pollen is in the opposite direction of the sun, therefore the dance is vertical and directed downwards.

The angle formed by the dance plane with the vertical is the same as the angle formed by the food with the Sun in a horizontal plane. The vertical, seen from below upwards, represents the sun's direction, and the angle of the spoils' direction with that of the sun is duplicated in respect to the azimuth.

1.2.5.3: Artificial Bee Colony

The ABC (Artificial Bee Colony) algorithm was developed by Karaboga and Basturk in 2007 [23], inspecting the behavior of real bees to discover the food source, termed nectar, and share knowledge from food sources to others bees in the nest.

The ABC (Artificial Bee Colony) algorithm is one of the most-recent population based (swarm intelligence based) meta-heuristic algorithms, which simulate the foraging behavior of honey bee colonies

The minimal model of swarm-intelligent forage selection in a honey bee colony which the ABC algorithm simulates consists of three kinds of bees. These bees get flower's nectar and carry them to the hive. The communication is performed by a special dance. The ABC (Artificial Bee Colony) algorithm is introduced to find the food source with real bees. There is only one bee that job with artificial bees is defined and categorized into three groups: bees foraging for food, observation bees, and scouts.

Employing bees share information with beekeepers in a beehive so that beekeepers can choose a food source to explore.

1.2.6: Particle Swarm Optimization (PSO)

Particle swarm optimization is a population-based method inspired by animal behaviors that uses particles to discover the global minimum while taking into account the swarm's social and cognitive activities. Basic particles in the PSO are characterized by their position and velocity in the search space.

Particles are drawn to certain locations in the search. space that represents their best personal discovery as well as the swarm's best discovery (local-best and swarm-best).

Positions in the world's top ten) [24]. The PSO, on the other hand, has its own flaws in terms of Iii) early convergence, and iii) a lack of dynamic adjustment As a result, hill-climbing solutions are not possible [25], [26], [27].

In order to overcome these challenges, Because of the PSO's disadvantages, some modified versions of the PSO have been developed.

for path planning and navigation of mobile robots PSO, on the other hand, has been demonstrated to be effective in a number of investigations [28], [29], [30], [31], [32].

1.2.7: Genetic Algorithm:

Genetic Algorithms are part of the Evolutionary Algorithms family of algorithms [33]. These algorithms are random optimization techniques based on Darwin's theory of evolution, with the goal of arriving at an approximation solution at the right time. Genetic Algorithms use techniques derived from genetics and natural evolution: crosses, mutations, selections, and so on. They are a stochastic optimization method of order "0," which means that neither continuity nor differentiability are required for the method to run smoothly; only knowledge of the value where the function to be optimized is close enough is required. As a result, the effectiveness of a genetic algorithm is dependent on a thorough understanding of the problem to be solved. Genetic algorithms are the outcome of research conducted by John Holland and his colleagues and students at the University of Michigan beginning in 1960 [34].

The novelty brought by considering the crossover operator in addition to mutations in this operator, which allows us to get closer to the optimum of a function by merging the genes included in separate introns in the individuals of the population.

Natural selection is the basis of genetic algorithms, which apply it to a population of solutions to a problem.

The stages of a genetic algorithmes execution are as follows:

Initial population: We must choose a random population of n chromosomes, where each chromosome indicates the robot's next position, and we can also adjust it so that each gene represents the robot's next orientation.

Measure the fitness of each chromosome in the population using the fitness function.

Create a new population and repeat the processes above until the population are complete.

Crossover and mutation: Each pair produces two progeny, with two chromosomes exchanging one or more portions for data from the new chromosomes during these operations. If no mixing occurs, the out come is a carbon duplicate of the parents.

Mutation is the process through which one gene in a chromosome might randomly replace for another.

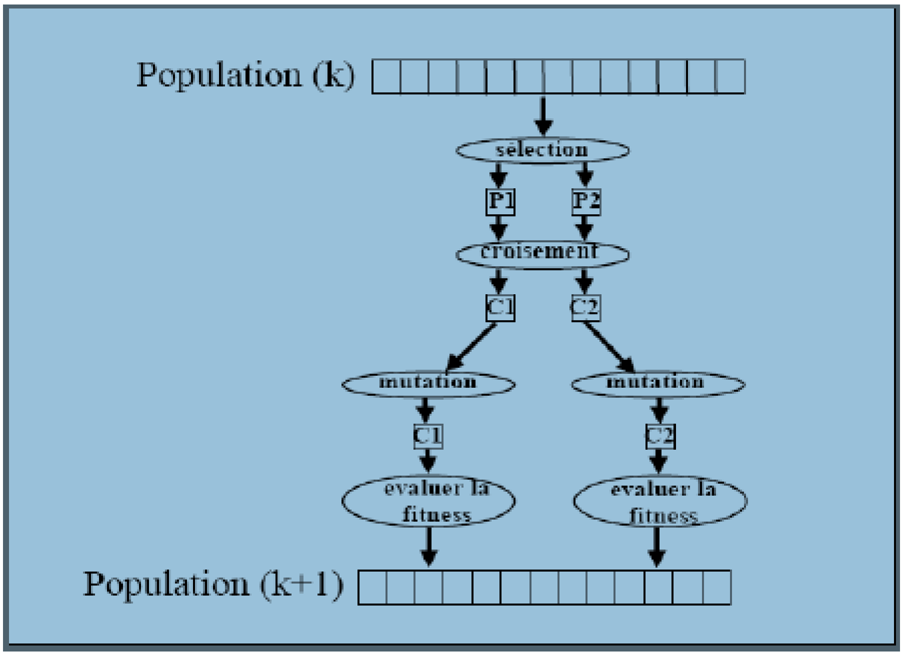

Figure 1-8The general principle of how a genetic algorithm works [34].

In terms of calculation volume, these algorithms are costly, especially when it comes to the evaluation function and memory size used.

- The path found is not optimal: Even after a large number of generations, it is impossible to be certain that the solution identified is the best. We can only be certain that we are approaching the ideal solution (the parameters and the evaluation function).

- There is no guarantee that the algorithm will converge or that the non-existence of a solution will be discovered. Algorithms are more difficult to construct when some parameters, such as population size or mutation rate, are unknown. However, the algorithm's efficiency is further limited by the fact that it is dependent on a number of tests. Furthermore, selecting an appropriate evolution function is essential critical. Itmust, therefore, choose carefully.

- The problem of local optimums: if an individual occupies a significant position in a population at a given moment, the population will converge towards that individual and cannot evolve.

There are several efforts and some methods to correct this problème, which is tiède to the principle of the algorithm itself and has no relevance to the environment or the robot utilized. The results of using genetic algorithms in the realm of autonomous robot trajectory search are not encouraging. In addition to the duration and volume of crucial computations, the method's convergence is insufficient, which leads to the hybridization of this type of algorithm with other algorithms. The genetic algorithm outperforms traditional optimization strategies.

Using only the function: objective evaluation without regard for its nature. Indeed, we didn't need any specific properties on the function to optimize it (continuity, differentiability, convexity, etc.), giving it additional flexibility and applications.

The coding of parameters, rather than the settings themselves, is used by GAs. Chromosomes are represented in binary. As a result of this decision, they are intuitively applicable to all problems whose solutions can be converted to binary. A chain of bits is then used to represent the chromosomes. This representation is independent of the problem and adds to the robustness of the genetic algorithm.

1.2.8: Roadmap approach

The Retraction, Skeleton, Highway, or Freeway approach is another name for the roadmap approach. One of the early path planning that has been widely used is the technique to solve the problem of finding the shortest way. The concept of configuration space (C_{space}) and the continuous path is used in this technique. The C_{space} is used in this strategy, and the fundamental aspect of this technique is the creation of a roadmap or freeway.

The visibility graph, voronoi diagram [35], and other sorts of roadmap techniques are available. It is important to note that the use of because this strategy is so reliant on sensory range and precision, VD is strongly reliant on it.

The distance between the robot and the obstacles is maximized. It's capable of doing a lot of things.

The VG's flaws are being addressed. Figure 1-9 shows a straightforward VD example (b).

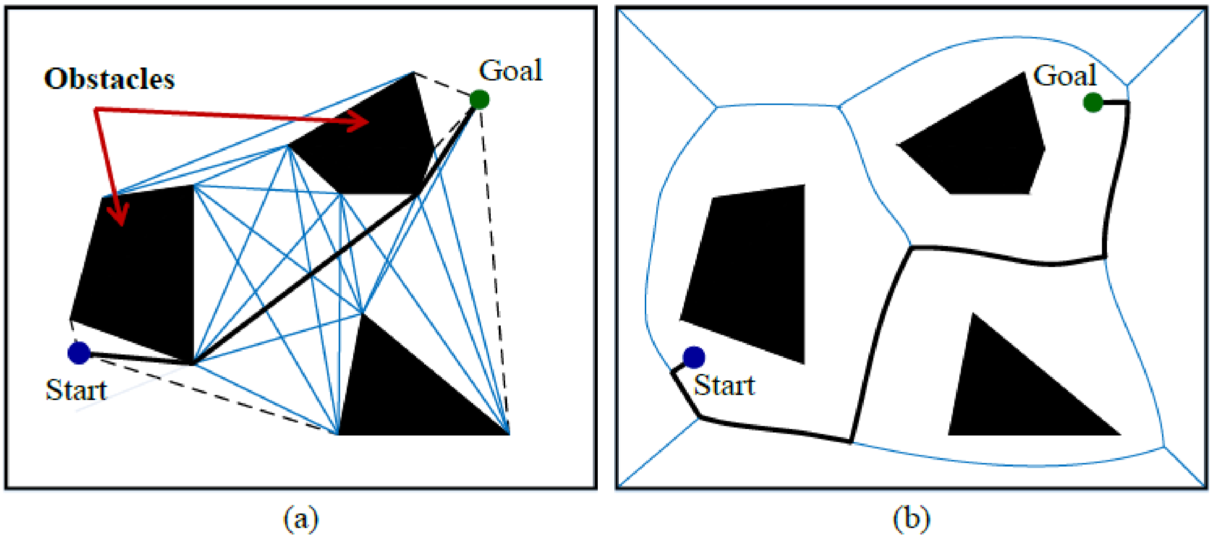

Figure 1- 9: Roadmap based path planning: (a) Visibility graph, (b) Voronoi diagram

In robot path planning, the highway net and silhouette have become more popular [36], [37], in more familiar situations than the others.

The roadmap approach is based on recording the connectedness of a network of 1-D curves in the robot's free space (straight lines). The roadmap is a set of straight lines that connect two nodes of distinct polygonal obstacles in the free space Cfree. All segments connecting a vertex of one obstacle to a vertex of another without entering the interior of any polygonal obstruction are shown. If a continuous path can be located in the roadmaps free space, the beginning and objective points are connected to it to reach the ultimate solution, which is a free path. Dijkstra's shortest path algorithm is frequently used to discover the optimal path when more than one continuous path is found and the number of nodes in the graph is modest. The visibility graph, voronoi diagram, and other sorts of roadmap techniques are available. In robot path planning, the highway net and silhouette have become more popular, in more familiar situations than the others. The roadmap is categorized as a comprehensive strategy (i.e. it finds a free path, if one exists).

Other non-complete (probabilistic) versions for building and searching the roadmap do exist [38]. In general, probabilistic roadmaps boost the algorithm's speed. The following are the major drawbacks of the roadmap approaches:

(i) The roadmap's goal is to find a way that is not congested (not an optimal path or near-optimal)

(ii) Due to the necessity for flexibility, it is complex and unsuitable for dynamic contexts. Whenever a change occurs, the roadmap is reconstructed.

1.2.9: Cell Decomposition (CD) approach:

In path planning problems, the cell decomposition method is widely utilized in the literature. The primary idea behind this method is to discover a path between the beginning point and the objective point by splitting the open space of the robot's configuration into smaller portions known as cells. It lowers the search space in this representation of the environment, or in other words, words, the C_{free} is decomposed into cells. After this decomposition, a search operator is used.

From the starting point to the destination point, is used to find a succession of collision-free cells. The adjacency relationships between the cells are used to construct the connectivity graph, where the nodes represent the cells in free space and the links between the nodes reflect the connections between the nodes that the related cells are in close proximity to one another. When a cell becomes corrupted due to the presence of a portion of an obstruction, the corresponding cell is divided into two new cells, and the obstacle-free cell is then added to the collision-free path. The steps to take in a CD-based path planning strategy for a mobile robot are outlined below.

Different CD techniques have been introduced [39].

- Exact Cell Decomposition

-Approximate Cell Decomposition

- Probabilistic Cell Decomposition

1.2.10: Artificial Potential Fields (APF):

The authors were the first to imagine the idea of fictitious forces acting on the robot in [40] and [41]. The purpose of the path planning algorithm of mobile robot navigation is to go to the target while avoiding collisions with obstacles in known or unknown settings. In this regard, in addition to the previously mentioned capabilities of simultaneous path planning and navigation, obstacle avoidance, appealing mathematical representation, and conceptual simplicity have all contributed to the success of this project. Path p has popularized potential fields.

The potential field-based path planning algorithm, on the other hand, has an inherent local minima problem, where the robot can become trapped in another field.

It is in a position that is away from its purpose. In order to solve this challenge, few techniques are presented. These will be considered in conjunction with the possibility of field-based path planning.

Reaching a goal position in mobile robotics without colliding with obstacles is a difficult task.

Creating a positive workplace atmosphere is a difficult task. One simple way is to create an artificial potential field. It's commonly utilized in such situations, and it's simple to adopt for avoiding roadblocks. The idea behind potential fields is that some virtual potential fields interact with the robot to govern its movements. Because it reacts to external factors, a potential field has a reactive architecture. It can be used as an off-line global planner as needed.

In static environments, entire information about a map or area is known. Alternatively, utilizing real-time sensor data, it can be utilized as an online local planer. In the course of our implementation, we generate the potential field for local path planning using real-time laser range data.

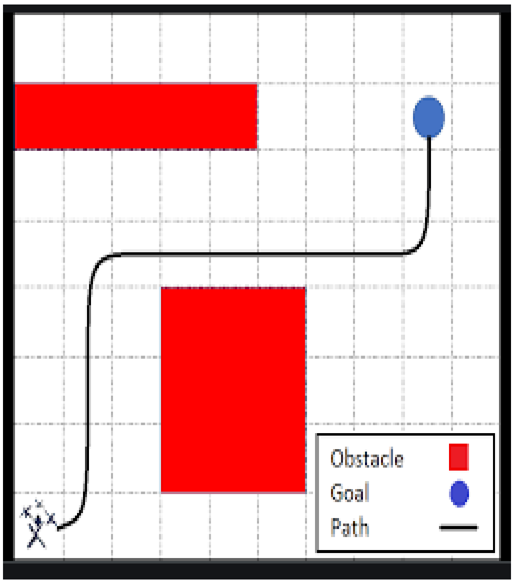

Figure 1-10 the path and obstacle avoidance for artificial potential field

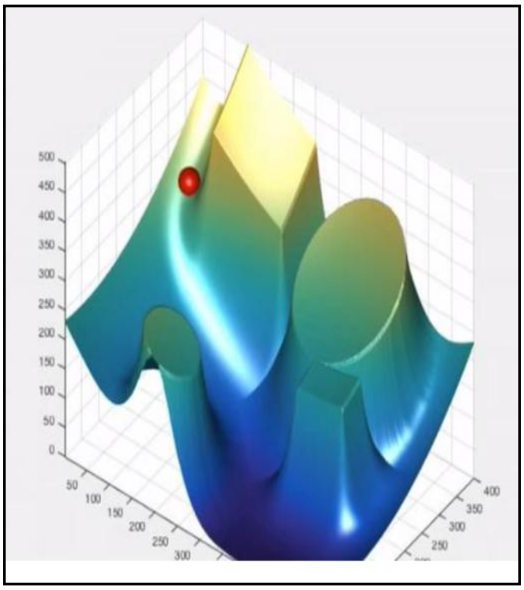

Figure 1-11 the robot move in the artificial potential field

One of the most often utilized path planning and obstacle avoidance strategies for autonomous mobile robotics is the artificial potential field. It can be viewed as a landscape with multiple mountains formed by barriers and valleys, with the lowest valley point being the 31st target point. In the field of robot route planning, a robot is viewed as a particle that goes from a high potential point to a low potential point.

Khatib [42] first proposed the artificial potential field approach for obstacle avoidance in robotics applications in 1985. There are two fields in this: attractive potential field and appealing potential field with a repulsive effect. The aim generates appealing potential, which attracts the robot.

The robot is pushed away from the detected obstacle by the repulsive potential produced by the detected obstacles inside the effective region, avoiding a collision, when the robot is immersed in a potential field, a combination of both potential fields acts on it to lead it to the objective point. This pair of forces is dedicated to controlling the robot's motion in a collision.

The movements of the latter (robot) are iteratively calculated by an algorithm of "Descent the gradient" of the obtained potential. It is sufficient to evolve the robot according to the decreasing gradient of the potential function in order for it to be drawn to the goal and pushed back by obstacles. This approach is first utilized for obstacle avoidance and control, then for planning [43].

The key advantage of this method is its calculation speed, which allows it to be used in real time, but it is always vulnerable to the occurrence of local minima, which can result in blocking or wobbling configurations [44].

Configurations that block or wobble [44], these minimums are usually linked to the geometry and distribution of obstacles in the robot's workspace, and specifically to penalty coefficients produced during the potential field building.

In order to remedy this problem, "Barraquand and Latombe" proposed a solution avoidance or rather the exit of a local minimum by the application of movements random, called "Brownians".

1.3 Comparison between Methods

Method	Advantages	Disadvantages
Fuzzy logic	- The closest to human logic. - Translate human experience into a set of principles. - It is not cumbersome in terms of calculation, time savings, or memory space.	- Because of the approximation reasoning process, the navigation path is not optimal. - Requires the presence of an expert. - These rules restrict robot operation.

Neural networks	- Create a straightforward solution -The capacity to precisely compute a navigation path	-The method of knowledge representation.
Ant colony	- Extremely adaptable. - Ideal for graph-based issues.	A blocking state is possible. - Execution time can be lengthy at times. -It isn't applicable to all types of issues.
Bees Colony	-Efficient in identifying the best options and simple to implement - Eliminate the difficulty of the local optimal - Responsive to difficult issues	-It has an evolution and diversification mechanism - Large values can cage the algorithm for numerous cycles and destroy a solution before it can be exploited.
Genetic algorithm	- AGs work with parameter encoding - AGs work with a population of points - The application of probabilistic transition rules to prevent local optimums. - Aerodynamic form, structure, and composite material synthesis	- The difficulty of industrial network scheduling - Form recognition and learning by decreasing it bioinformatics pattern identification – electronic circuit synthesis
CD	Complete, sound, 2D and 3D, point or rigid robot.	Non-optimal, Heavy computation time
APF	Real-time, 2D or 3D, point/ rigid Robot	Not-complete, non-optimal, local Minima

1.4. Conclusion

In this chapter, we reviewed the state of the art in mobile robot navigation and the many algorithms that may be used to manage it, such as finding the shortest paths and distances in a given area. We've also seen the many stages; we've also highlighted the benefits and drawbacks of each strategy, and we've discovered that there is no navigation strategy that provides a solution that eliminates all of them.

Chapter 2: System Design

2.1. Introduction

Robotic applications have shifted away from industrial environments and toward more demanding scenarios such as residential applications and space/deep-sea research in recent years. It is not possible to identify all of the barriers in the environment a priori in these situations in order to use a global path planner.

As a result, global path planning approaches aren't appropriate for this type of application. Instead, path planning algorithms must be developed that can handle both the real-time discovery of new obstacle information and the processing of that information quickly enough to compute the path online. Robot path planning utilizing potential is unlike any other local path planning method.

The appealing mathematical depiction, as well as the mental simplicity, has Potential fields have become prominent in path planning. However, there is a possible field-based path.
The planning method has an inherit local minima problem, which means the robot can get stuck in another area.

Reaching a goal position in mobile robotics without colliding with obstacles is a difficult task.
Creating a positive workplace atmosphere is a difficult task. One simple way is to create an artificial potential field. It's commonly utilized in such situations, and it's simple to adopt for avoiding roadblocks. One of the most often utilized path planning and obstacle avoidance techniques is artificial potential fields approaches for self-driving mobile robotics It can be viewed as a landscape with various different elements.
Mountains formed by obstacles and valleys, with the lowest valley point representing the source a point for you. In the field of robot route planning, a robot is seen as a moving particle from a position of enormous potential.

There are two types of artificial potential fields: Potential field with a repulsive effect, the aim generates appealing potential, which attracts the robot near the finish line the repulsive potential that the observed obstructions produce to avoid a collision, the robot is pushed away from the obstruction inside the effective region. When the robot is immersed in a potential field, a combination of both potential fields acts on it to lead it to the objective point. The goal of this combination of two forces is to keep the robot on a collision-free, safer course.

2.2.1: Our Problem:

The goal of our project is to determine the shortest path for a mobile robot. The goal is to find the best strategies to get to the desired end state for the least amount of money. We examine how APF can be developed and applied to large-scale planning issues with intractably enormous and highly complicated search spaces in order to identify optimal solutions.

Our effort especially attempts to respond to the following research question: *How can a robot get from a starting point to an arrival point in the most efficient and secure manner possible?*

2.2.2:The proposed solution:

The research focuses on adapting existing and inventing new efficient APF to tackle scale optimization challenges of tasks assigned to a mobile robot in order to answer the research issue.

Furthermore, the emphasis is on adapting and using an existing APF to assign problems with search spaces in order to locate specific assignments that are mapped onto a specific part of the solution space. Furthermore, existing APF are modified to overcome path planning issues.

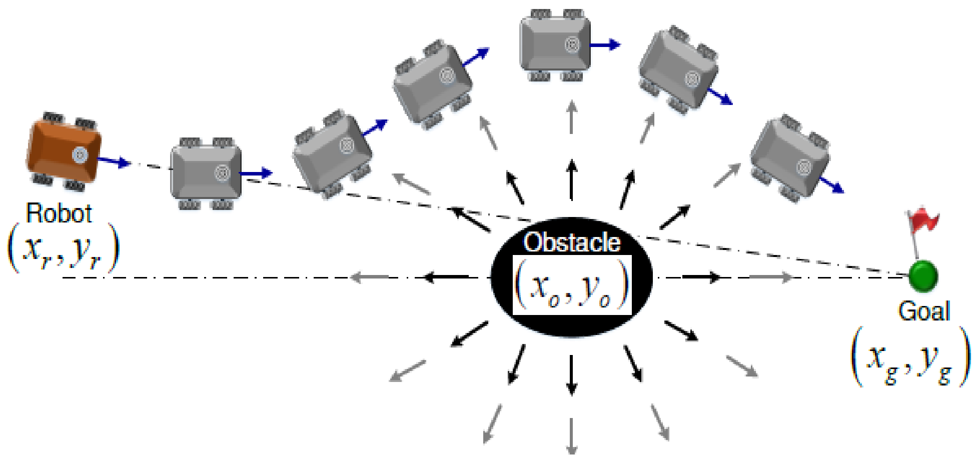

Figure 2-1: Obstacle avoidance concept using potential field

2.2.3: Problem definition:

We'll look at the following scenario: In the beginning, there's a state that depicts the environment, with obstacles strewn about at random.

A robot must go from a start point to an arrival point; the robot has a start location (Xr, Yr) and an arrival location (Xg, Yg), but it is unaware of its surroundings. The following difficulties piqued our interest: determining the best trajectory (shortest path) and avoiding obstacles.

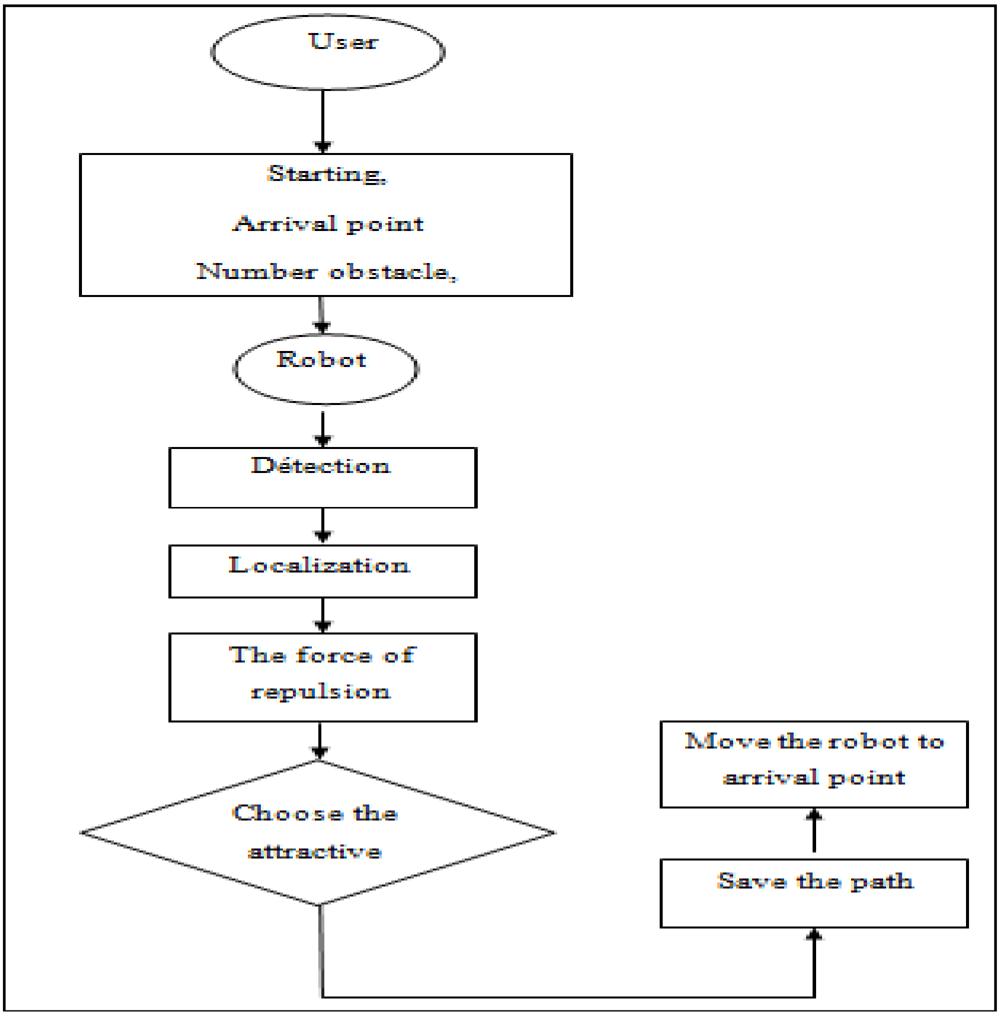

Figure 2.2: General Architecture of our system

2.2.4:Use case diagram:

A use case is a logical unit that represents functionality that can be seen from the outside. It delivers an end-to-end service for the actor who begins it, having an initiation, unfolding, and conclusion. As a result, a use case simulates a system-provided service without imposing the service's implementation.

The robot has four main functions: search, movement, obstacle avoidance, and detection:

• Research is the process of determining objectives.

• Movement refers to actions that involve changing positions in response to controller commands

• Obstacle avoidance is also a method of achieving objectives.

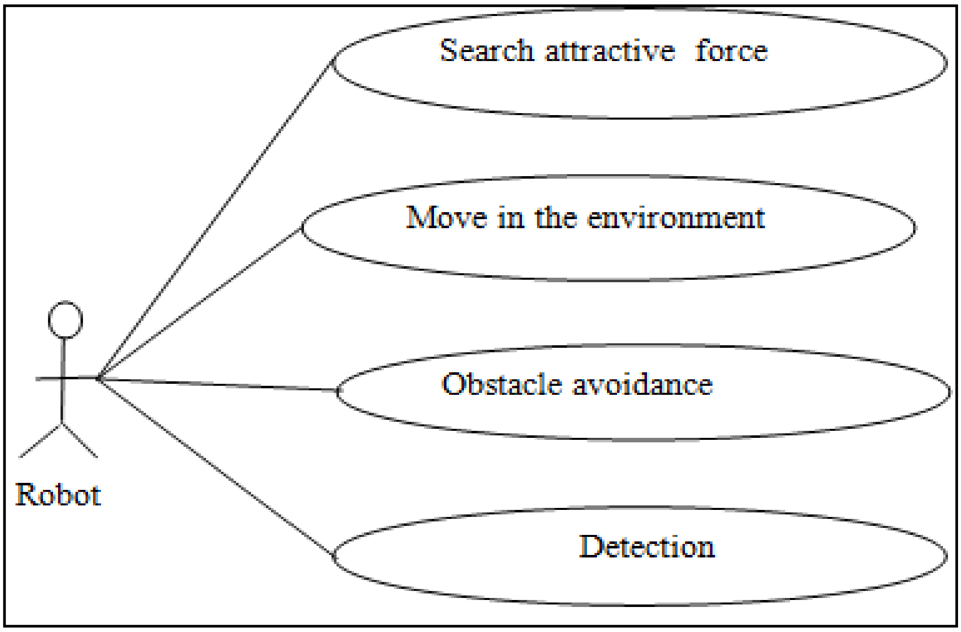

Figure 2.3: Robot use case diagram

We have a single entity in our diagram that represents our robot as the main player. The latter has four central procedures to carry out:

• The search for attractive force: our robot is in the starting position in an environment with an end point to reach, and it is up to it to determine the attractive force.

• Move in the environment; when the ending point in the recording is reached, the robot begins to move through the area.

• Obstacle avoidance: Because obstacles are put randomly in the environment based on the start and end points entered, the robot must avoid them when they cross its route.

The user's major tasks are to determine the starting and ending points, as well as the amount and location of obstacles.

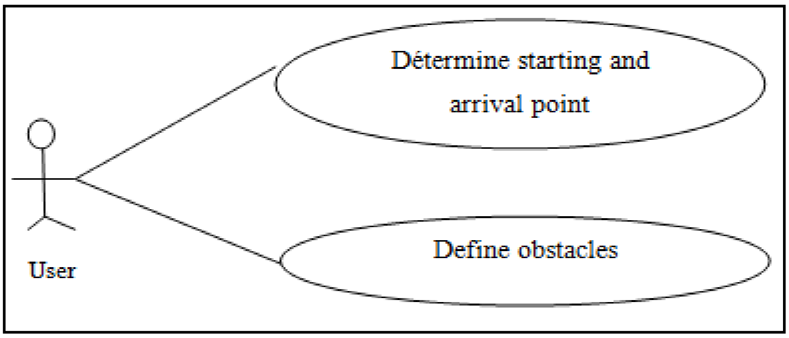

Figure 2.4: User use case diagram

The user is the only actor with a single operation in this use case diagram. When you first start, set the starting and ending points to decide the path that the robot must follow, the user also defines obstacles.

2.2.5: Sequence diagram:

A sequence diagram is a diagram created with the Unified Modeling Language (UML) that depicts the flow of messages between objects during an interaction. The communications that these objects exchange during the interaction are represented by lifelines in a sequence diagram.

Sequence diagrams show the order in which messages are sent between objects. They can also depict object-to-object control systems. The sequence diagrams in Figure 1.8 model communication between these objects.

The robot receives the instructions to find the path by sending the solution and then sends it at each position using the determination method.

Our process starts when the bots receive the path, which is then sent to the domain system for analysis. The system consults with the environment in the hand descended before creating a potential path for the robots.

The robot avoids the force of repulsion and chooses the attractive force to move the robot to the end.

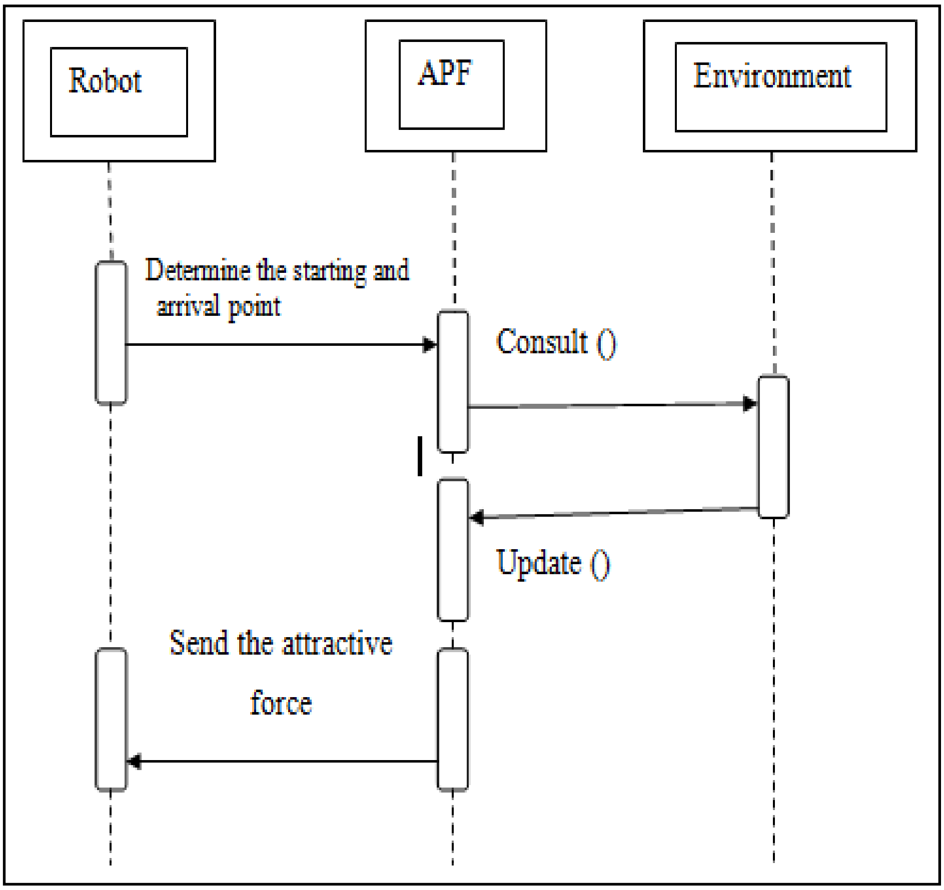

Figure 2.5: Sequence diagram illustrating the robot action

2.2.6: The Activity Diagram:

An activity diagram in UML depicts the behavior of a system by describing the process's sequence of actions because they represent the flows between actions in an activity, activity diagrams are related to information processing flowcharts. Because they represent the flows between actions in an activity, activity diagrams are related to information processing flowcharts. Simultaneous parallel and replacement flows can also be depicted in activity diagrams.

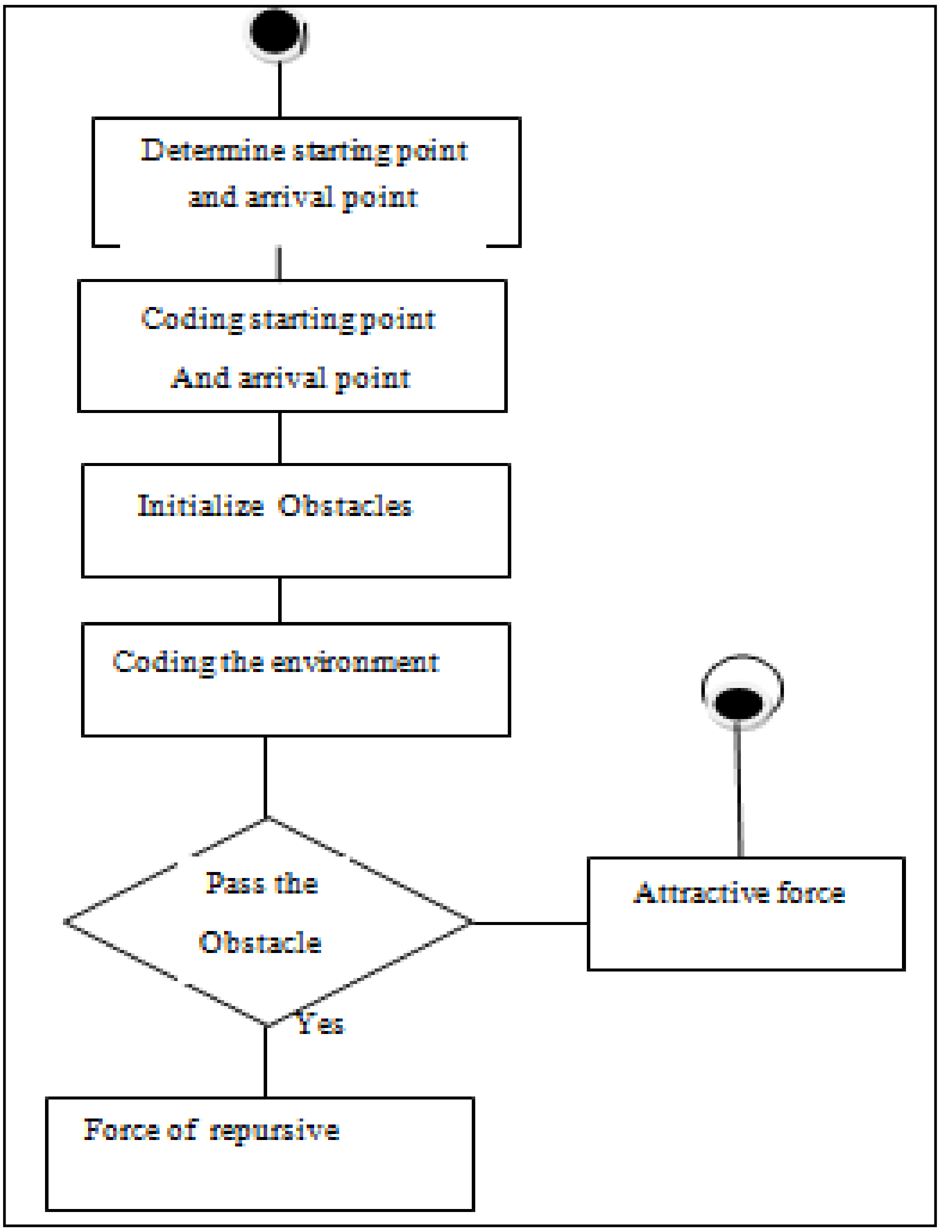

Figure 2.6: Activity Diagram

2.2.7: The Class Diagram:

The class diagram depicts the information model's static structure, particularly the existing objects and their internal structure, as well as their interactions with other items. A class diagram should not include any time-related information.

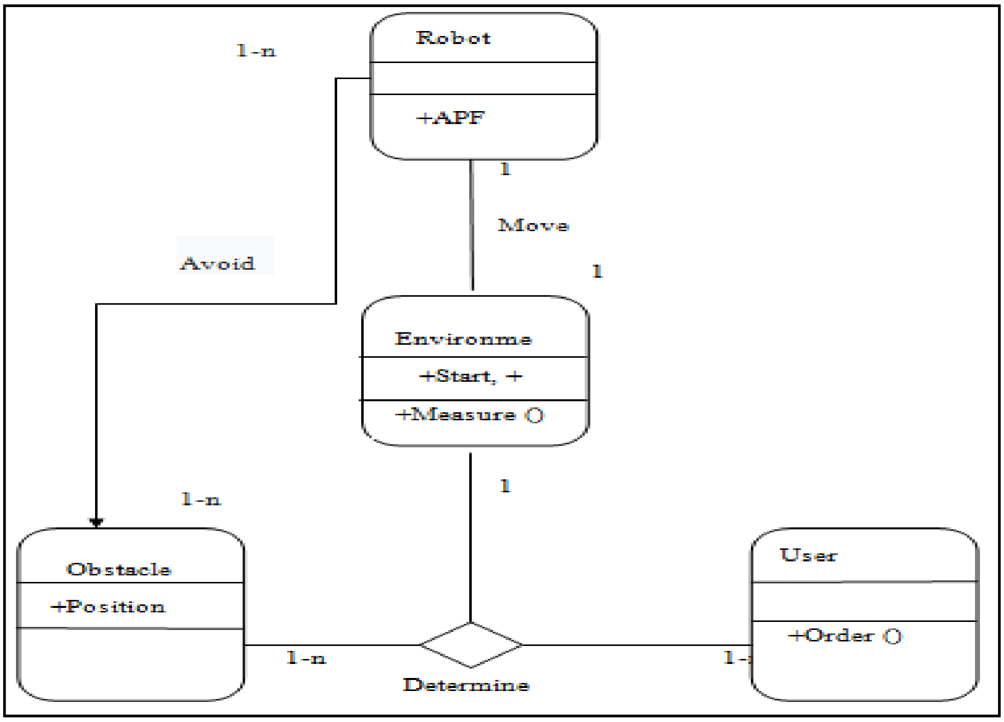

Figure 2.7: Class Diagram

Classes, subclasses, properties and values, methods, linkages (multiplicity, generalization, and composition), categories, and dependencies are all included in the content. So that the robot does not halt between two trajectories, the robot class must control the sequence of trajectory generation and commands. UML captures this behavior through the usage of Statecharts, which effectively depict parallelism.

- It manages all Artificial Potential Field operations, which is essentially managing the sequence of trajectory creation and commands to avoid errors and confusion, such as malfunctioning and abrupt halting.

- It manages the entry of the starting and ending places, as well as the measurements of the chosen path.

- Obstacles Class it oversees all obstacle operations, including positions. It is the user class that manages all of the user's operations.

2.3: Types of potential fields:

A variety of other forms of potential fields have been proposed that can be coupled to create more complicated fields , in addition to the numerous basic attracting and repulsive potential fields. Robot navigation techniques can be directed to specified locations by combining them, a variety of scenarios to fulfill various demands. As demonstrated in Figure 2- 8(a) and (b), the attracting and repulsive fields have properties that drive the robot towards the objective and away from the obstacle sites (b). The center point in the attractive field exerts an attraction force on the robot, causing it to try to move towards it.

Alternatively, the repulsive field is the polar opposite of the attracting field, and it is associated with obstacles. It's useful for keeping the robot away from potential hazards.
When a robot navigation task requires moving the robot in a defined preferred direction, such as moving the robot along a hallway, there may be no goal point specified. In such instances, the uniform potential field (Figure 2- 8(c)) can be employed to avoid the disadvantages of other types of fields.

A repulsive force can be generated by a surface, such as a wall, as well as by a point. The perpendicular field depicted in Figure 2- 1(d) can be used for this type of implementation. This field can be used to avoid geometrical obstacles like walls and boundaries.

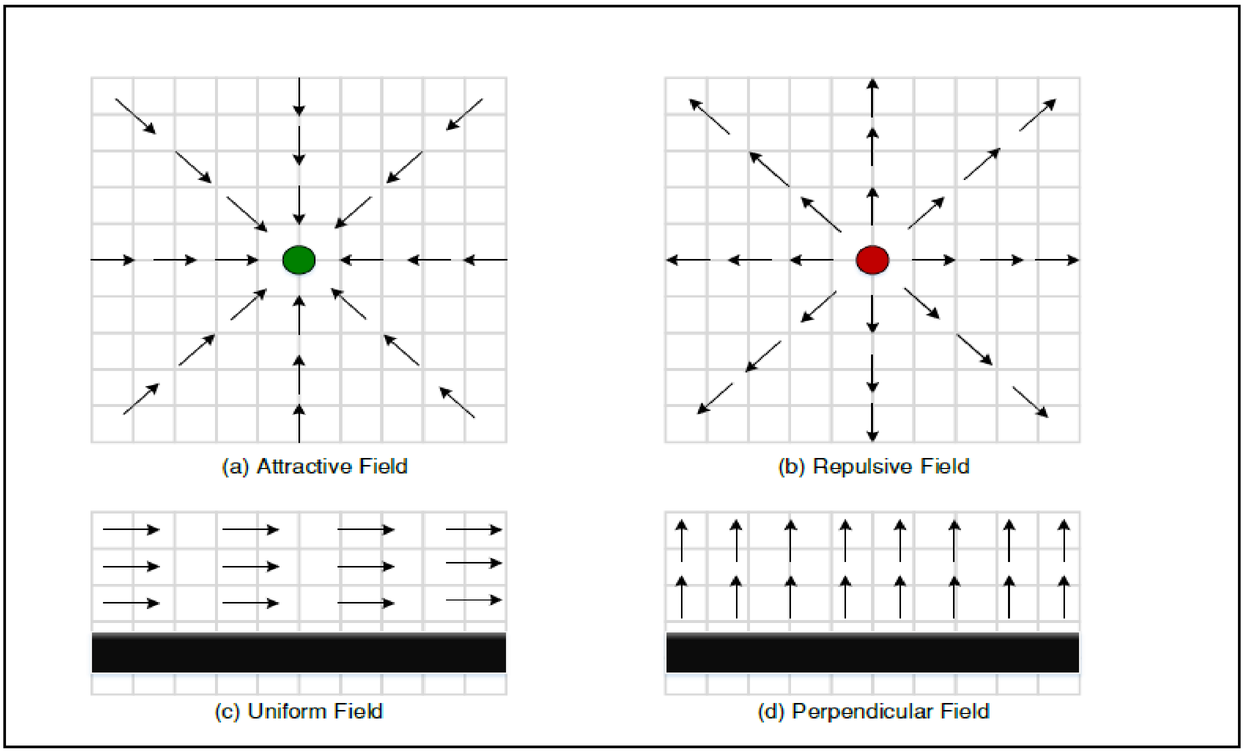

Figure 2- 8: Types of potential Fields.

Another sort of potential field called a tangential potential field, which can be used to lead the robot around an obstruction while also allowing the direction to be selected. The magnitude and direction of this field are determined in the same way as the magnitude and direction of the previous field.

2.4: The mathematical background of artificial potential field method:

The artificial potential field approach has become a popular path planning algorithm inrecent years. Due to their efficient mathematical underpinning and simplicity, mobile robots are a viable option.

This method takes advantage of a function for finding the robot's obstacle-free path to the target in the working space a rudimentary overview of the relevant mathematics and creation of data is provided in this part. The topic of a possible field is discussed.

The obstacles and goal configuration are taken into account sources to create a potential field that represents the configuration's features space. The attractive potential field is always one of two components of the potential field as well as the repellent potential field A vector force is generated in an attractive potential field by The goal setup directs the robot toward the goal. A vector force that exists in a repulsive potential field is generated by the barriers from their placements in relation to the robot's position.

The resultant of these two vector forces can therefore be defined as the total potential. The total potential force on the robot determines the robot's motion direction, or in other words, the robot's path.

Figure 2- 9 depicts the attractive and repulsive artificial force distribution on a robot caused by a particle impediment and a goal in an environment. The robot's position and the goal can be written as vectors of:

- pr = [xr, yr] T and pg = [xg, yg] T, respectively. Consider the position of the k^{th} particle obstacle as po = [xo k, yo k] T, as shown in Figure 2- 2 a.

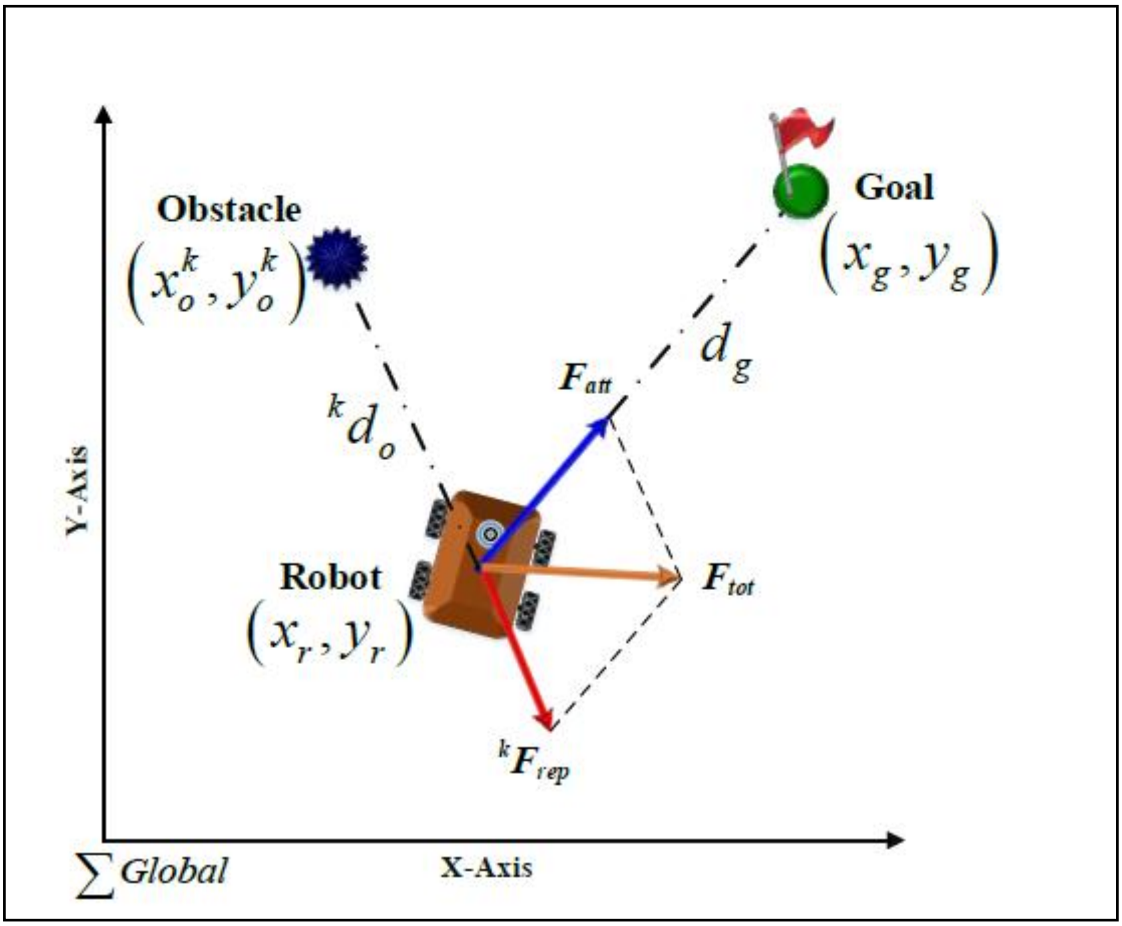

Figure 2- 9: Force distribution of the artificial potential field

To find the potential forces (attractive and repulsive) the positions of the robot, goals, and obstacles should be known.

2.4.1: Repulsive potential force:

Similarly to find the repulsive potential force, let $[ok,\ yok\ ]\ T$ be the position of the k^{th} obstacle in an environment. The repulsive force created by the k^{th} can be described by the Eq. (1)

$$^kF_{rep} = \begin{cases} a_o \cdot exp(-b_o \cdot {}^kd_o^2) \cdot {}^k\boldsymbol{e}_o & if\ {}^kd_o \leq d_d \\ 0 & else \end{cases} \qquad (1)$$

The greatest value of the repulsive force is specified as a_o. The k^{th} obstcale generates F_{rep} k, and b_o is a constant that reflects the width of the distribution. The variable The Euclidean distance between the robot and the k^{th} detected barrier is denoted by d_o k. The influence distance around the robot is defined by the parameter dd. A unit vector e_o k is defined as the vector towards the robot from the detected k[th] barrier, as given in Eq. 2.

$$^{k}\boldsymbol{e}_{o} = \frac{1}{^{k}d_{o}}\left(\Delta\, ^{k}x_{o}\boldsymbol{i} + \Delta\, ^{k}y_{o}\boldsymbol{j}\right)$$

$$\Delta\, ^{k}x_{o} = x_{r} - \, ^{k}x_{o} \text{ and } \Delta\, ^{k}y_{o} = y_{r} - \, ^{k}y_{o} \,. \qquad (2)$$

In fact, the overall repulsive force is equal to the sum of all individual repulsive forces. Forces are the same as in Eq. 3.

$$\boldsymbol{F}_{rep} = \sum_{k} {^{k}\boldsymbol{F}_{rep}} \qquad (3)$$

In general, repulsive potential fields are designed to generate a high potential around the obstacle; as a result, the amplitude of the repulsive potential force near the barrier is enormous and gradually reduces as one move away from the barrier. As seen in Figure 2-3, the force vector is pointing away from the obstruction.

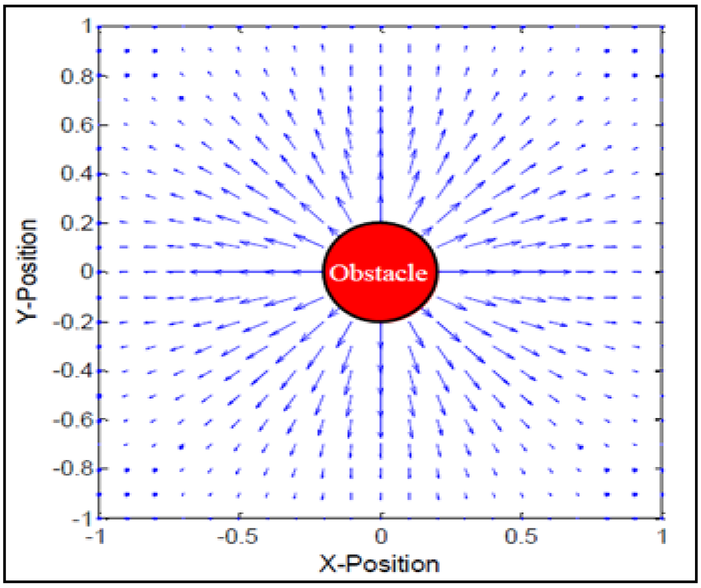

Figure 2- 10: Repulsive force distribution around the goal

2.4.2: Attractive potential force:

While many other attractive potential functions have been proposed in the literature, the attractive force created by the goal can be described using the Gaussian function in Eq (4).

$$\boldsymbol{F}_{att} = a_g\left[1 - exp\left(-b_g \cdot d_g^2\right)\right] \cdot \boldsymbol{e}_g$$

(4)

Where ag is the maximum value of the attracting force Fatt and is a pre-defined parameter. The width of the distribution is represented by the constant bg. The dg parameter is the distance between two points. The robot's Euclidean distance from the goal position, which can be calculated as follows.

$$d_g = \sqrt{(x_r - x_g)^2 + (y_r - y_g)^2}$$

(5)

can be used to express the unit vector towards the goal position eg (6)

$$\boldsymbol{e}_g = \frac{1}{d_g}(\Delta x_g \boldsymbol{i} + \Delta y_g \boldsymbol{j})$$

$$\text{Where } \Delta x_g = x_g - x_r \text{ and } \Delta y_g = y_g - y_r.$$

(6)

The attractive potential field, in general, takes the shape of Figure 2- 11, where each point of The force components are pointing towards the goal point in the working space.

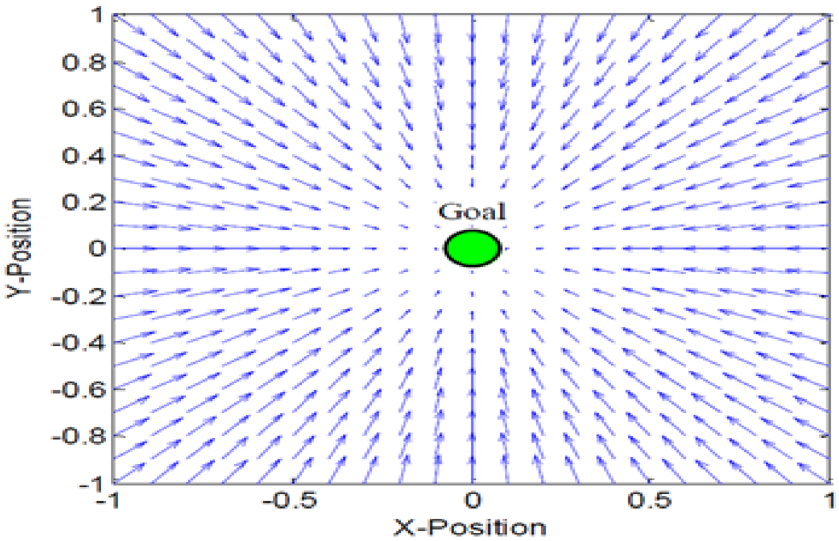

Figure 2- 11: Attractive force distribution around the goal

2.4.3: Total potential force:

There will be one goal or multiple goals, as well as impediments in the surroundings, in robot navigation challenges. As a result, the forces generated by the two fields by obstacles and goals must be combined. By adding the repulsive potential to the total potential field, the total potential field is obtained. All of the obstacles' forces, as well as the goal's appealing repulsive force Eq. can be used to express this (Eq7).

$$\boldsymbol{F}_{tot} = \boldsymbol{F}_{att} + \boldsymbol{F}_{rep} \tag{7}$$

Figure 2- 5 shows how to generate the total potential force distribution. A circle-like obstacle is positioned at (1, 1) with a radius of 0.5, and the goal point is positioned at (-1, -1). Except for the surrounding effective area of the obstacle, the total potential force field vectors are heading away from the obstruction but towards the goal.

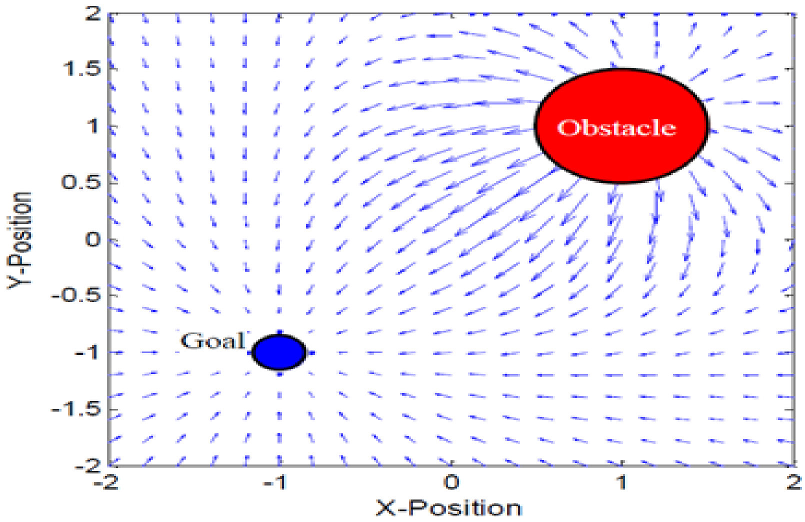

Figure 2- 12: Total potential force distribution

2.3. Conclusion

In this chapter we describe the scope of our artificial capabilities for mobility in a mobile robot, by developing this criterion of finding a solution to the good performance that is sensitive to uncertainty, the goal being to find a path to reach the goal point.

We have provided various diagrams: start states diagram, sequence diagrams, activity diagram, class diagram, these diagrams make us get a solution so that the robot can move in the obstructed spaces.

The simulation results make the robot able to achieve its goal by avoiding obstacles, for this reason, we propose the field of artificial capabilities to solve this problem, and in the next chapter will be the tools and languages used in our application.

Chapter 3: Realization

3.1. Introduction

To demonstrate autonomous control of the mobile robot, we used a MATLAB application that is an intelligent system with numerous functions for simulating the robot's behavior in obstacle avoidance to achieve the destination and evaluating the performance of the strategies utilized.

Optimization problems are becoming more complicated, and technological advancements have made the employment of an evolutionary strategy more vital. Furthermore, the cost-to-performance ratio in parallel IT systems is decreasing. The evolutionary technique is used in the design and implementation of meta-heuristics to speed up research, improve the quality of solutions obtained, increase robustness, and address large-scale challenges.

Our study in this thesis is focused on the topic of determining the shortest path for an autonomous mobile robot, based on the Artificial Potential Field; we developed a new evolutionary strategy for solving the challenge of identifying the path for mobile robots

3.2Tools and working environments:

3.2.1:Software environment:

Matlab or Matrix Laboratory is a high-level programming language consisting of an interactive environment mainly used for numeric computation, programming, and visualization. It has been developed by MathWorks. Matlab's core functions include charting functions and data, creating user interfaces, and manipulating matrices. It also supports C, C++, FORTRAN, and Java for interfacing with other programming languages. In addition, it is used to analyze data, develop models and applications, and develop algorithms .In addition to all of this, introduction to Matlab includes a number of built-in functions for executing mathematical operations such as multiple calculations, numerical methods, plot generation, and a variety of other tasks.

Matlab also has a lot of applications in the automotive and robotics fields, and it's widely used in the medical, automotive, and aerospace industries. It contains a large library of built-in functions that support a variety of mathematical functions such as filtering, optimization, Fourier analysis, linear algebra, statistics, numerical integration, and differential equation solution [45].

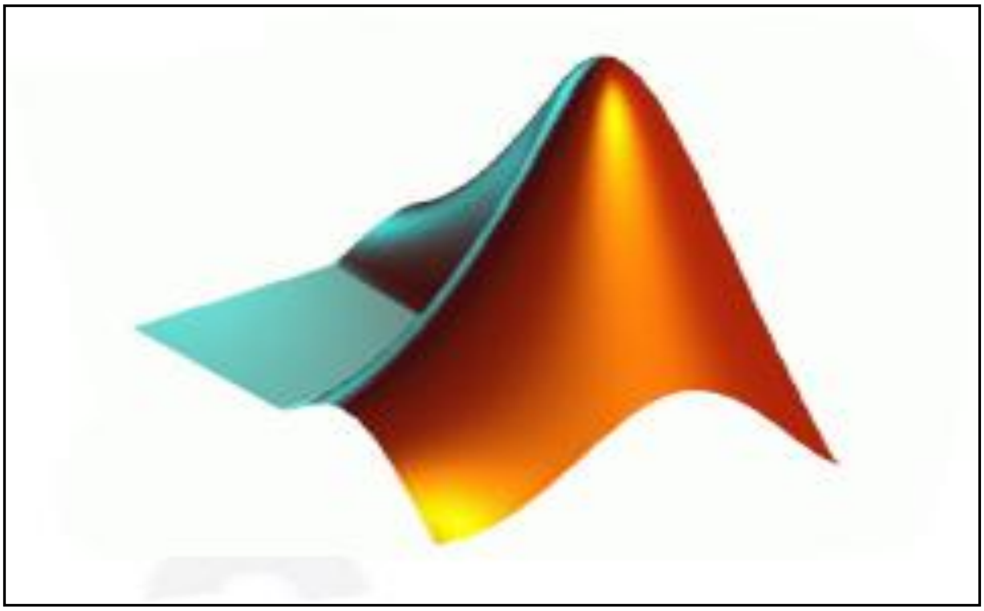

Figure 3.1: Matlab [45]

3.3. Description of the different modules

3.3.1:The environment:

An Artificial Potential Field (APF) method treats a robot's shape as a point in a potential field that combines attraction to the target and repulsion from obstacles. Paths are created from the obtained trajectories. The advantages of this strategy are a small computation amount and ease of comprehension.

They may, however, fall into the local minimum of the potential field and be unable to discover the path, or they may be unable to find the best path. Artificial potential fields can be thought of as a continuous equation, as the electrostatic potential field (thinking of a robot as a point charge), or the movement through the field can be discretized using a set of language rules".

Below, borrowed by 2D simulation, obstacles to doing simulation; so we started by defining the beginning and the goal.

The artificial potential field method involves creating an artificial potential field to mimic this mechanism under known starting, end, and obstacle positions. The artificial potential field method has the advantage of being a feedback control mechanism.

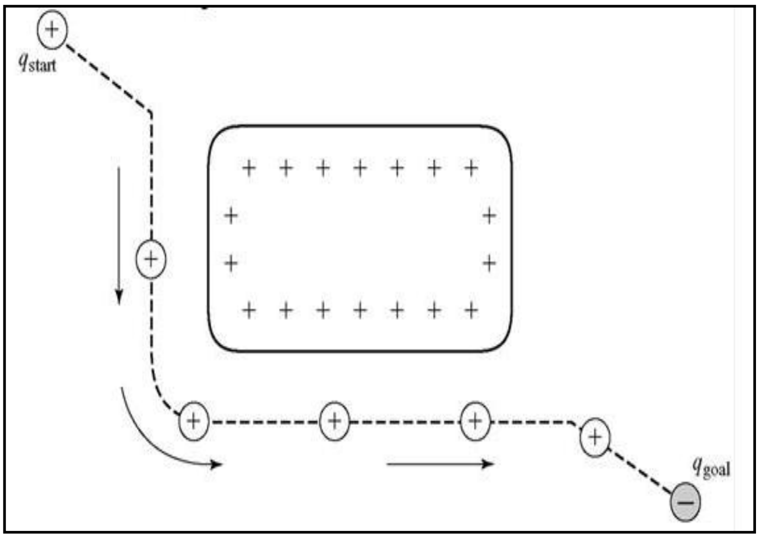

Figure 3.2: determine the starting point and the ending point.

Let's use two analogies to explain how the artificial potential field method works. First, the configuration space is compared to an electric potential field plane, and the robot (its current configuration) is compared to a point in space.

Whereas if the robot's beginning location and obstacles are positively charged, the robot's arrival points are negatively charged, and it is positively charged. The robot will proceed to the end along a given path under the operation of electric field force and avoid positively charged barriers, as indicated in Figure 2.8, due to the principle of repulsive charges of the same electric charge and attraction of charges of the opposite electric charge.

```
clc
close all
clear all
step = 10;
goal = [600, 70];
start = [200, 600];
```

Figure 3.3: Matlab code of determine the starting point and the ending point.

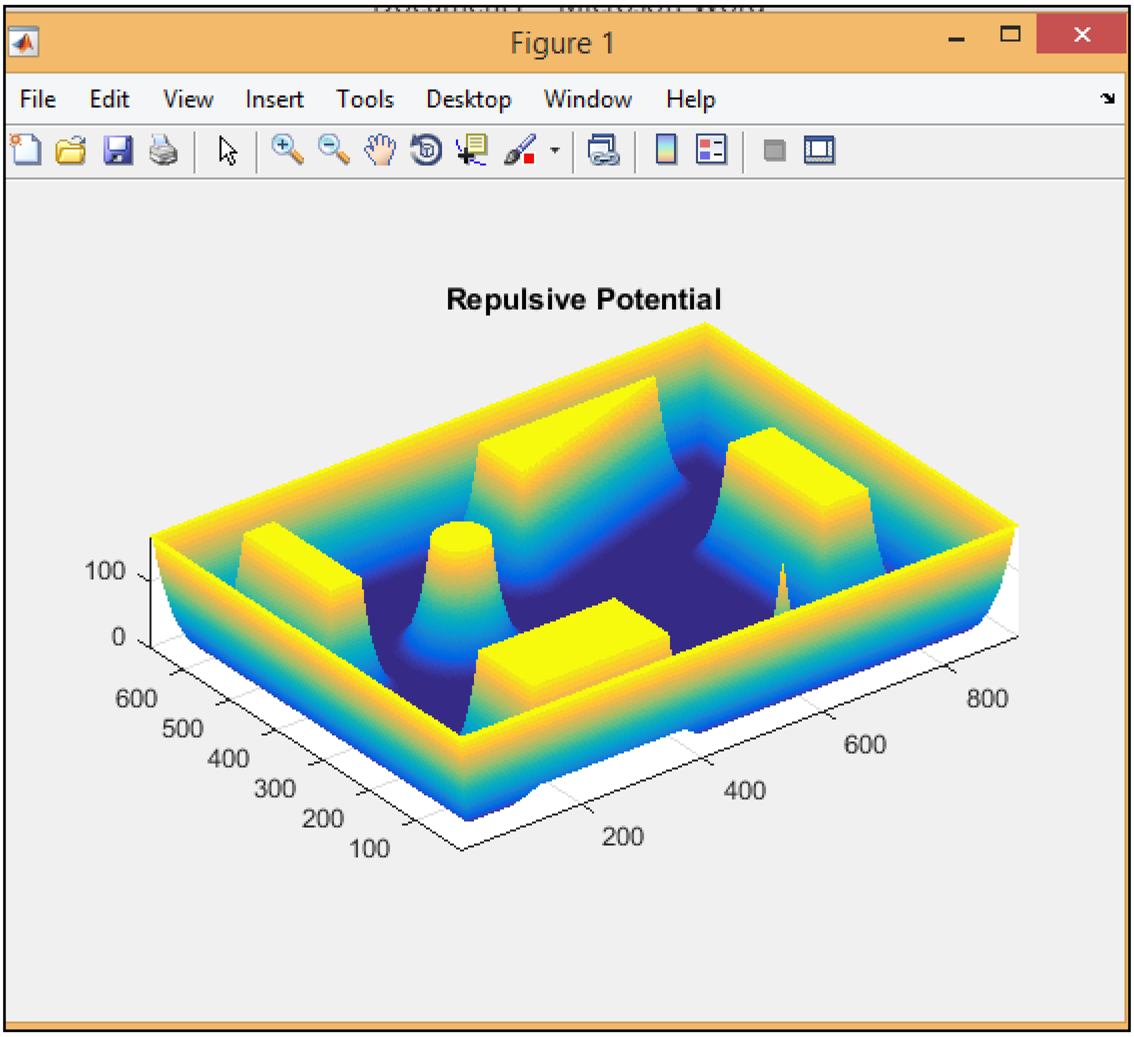

Figure 3.4: repulsive potential

3.3.2: Code the Repulsive Potential:

The Repulsive Potential: The behavior of a properly designed repulsive potential function should resemble that of a tightened spring or magnets approaching each other, in order to achieve obstacle avoidance. The minimum distance required for a barrier to be created the repulsion gain is a repulsive effect on the robot.

The total repulsive potential field can be calculated using the following formula: Taking into account all of the potentials created by the hurdles, by pursuing the negative, the movement is realized. The sum of attractive/repulsive potentials gradient, the translation velocity assignment is simple in this investigation.

Potential Navigational functions many additional local-minima free potential field solutions, such as wave-front, have been offered as a solution to the additive attractive/ repulsive potential field's local minima problem. As a result, for high-dimensional and vast configuration spaces, they become computationally intractable. To overcome the problem of local minima, a custom function must be built that has the only minimum at the objective position.

- When the object is far away from the target point, gravity exerts a particularly strong repulsive force; in a tiny or even negligible situation, the object may run into an obstruction along the route.

- when there is an obstacle near the target site, there will be a large repulsive force, gravity is relatively small, it is difficult to reach the object's target point, at a certain point, attraction and repulsion are exactly the same size, like inverse direction, the object will fall into a local optimal solution or shocks, a variety of improved versions of the potential field.

The initial field's repulsion, together with the impact of distant targets and objects intuitively, as the item approaches the target, the repulsive force increases, but the distance decreases, implying that the drag is repulsive to some extent.

3.3.3:Code Attractive Potential:

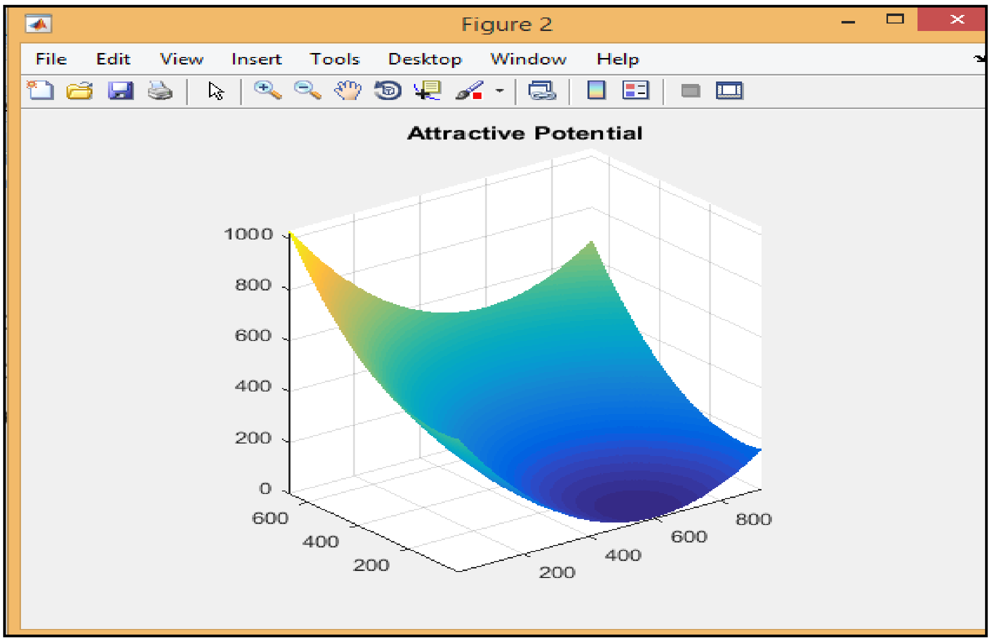

Figure3.5: Attractive Potential.

The attractive field is made possible by the repulsive field is the target and direct to the target point, whereas the target and direct to the target point are the target and direct to the target point, respectively.

Synthesis of a repulsive field of many obstacles and the synthesis repulsive field is pointing away from obstacles.

As a result, the APF of a robot, which is defined as the resultant of attracting and repulsive fields, is the potential function. Along the APF direction, the robot regulates its movement toward the target location. By searching the route along the decline direction of the potential function, the robot might identify a collision-free path using the APF method. The attractive field between robot and target is constructed to pull the robot to the goal area. The attractive field created by the goal is given.

3.3.4. Configuration Space:

In This figure, we show the environment in which the robot can move.

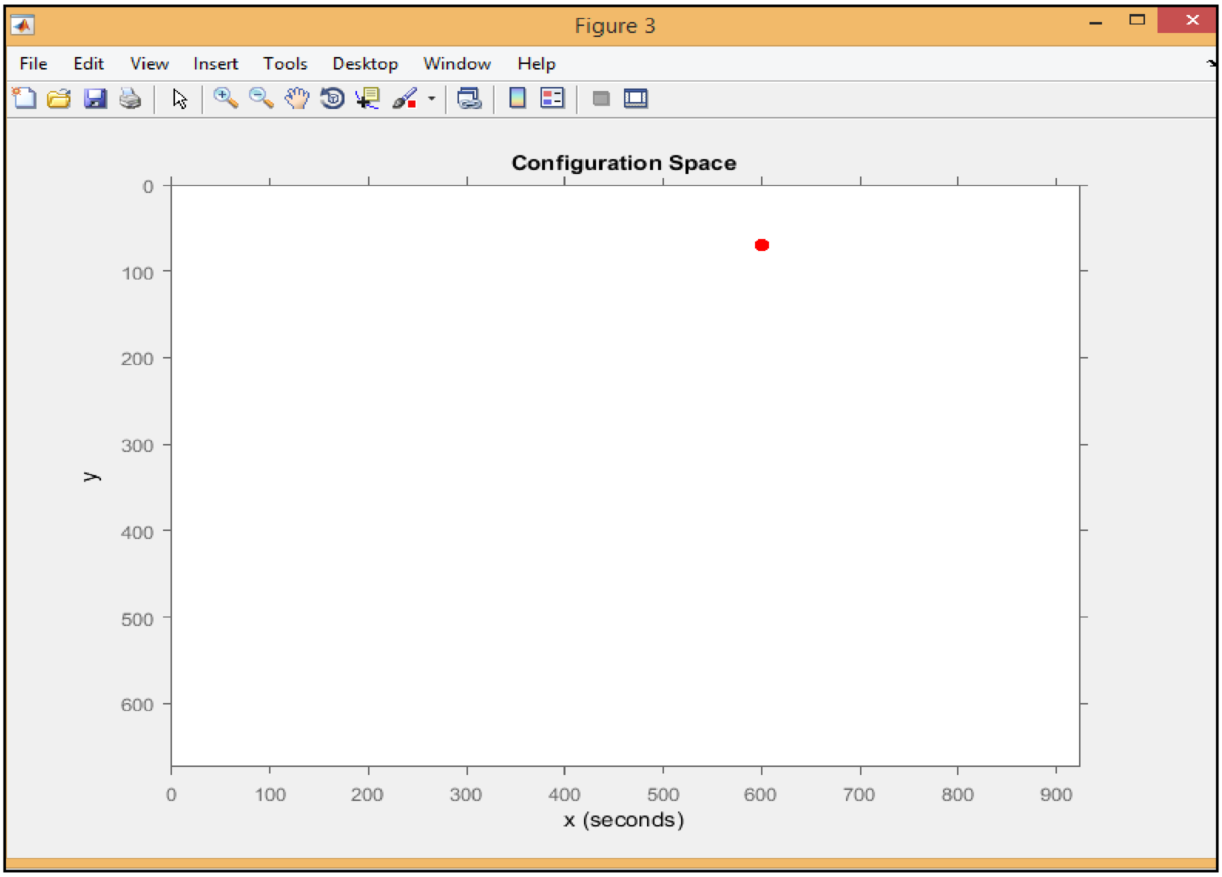

Figure3.6: Configuration Space.

3.3.5. Code Total Potential:

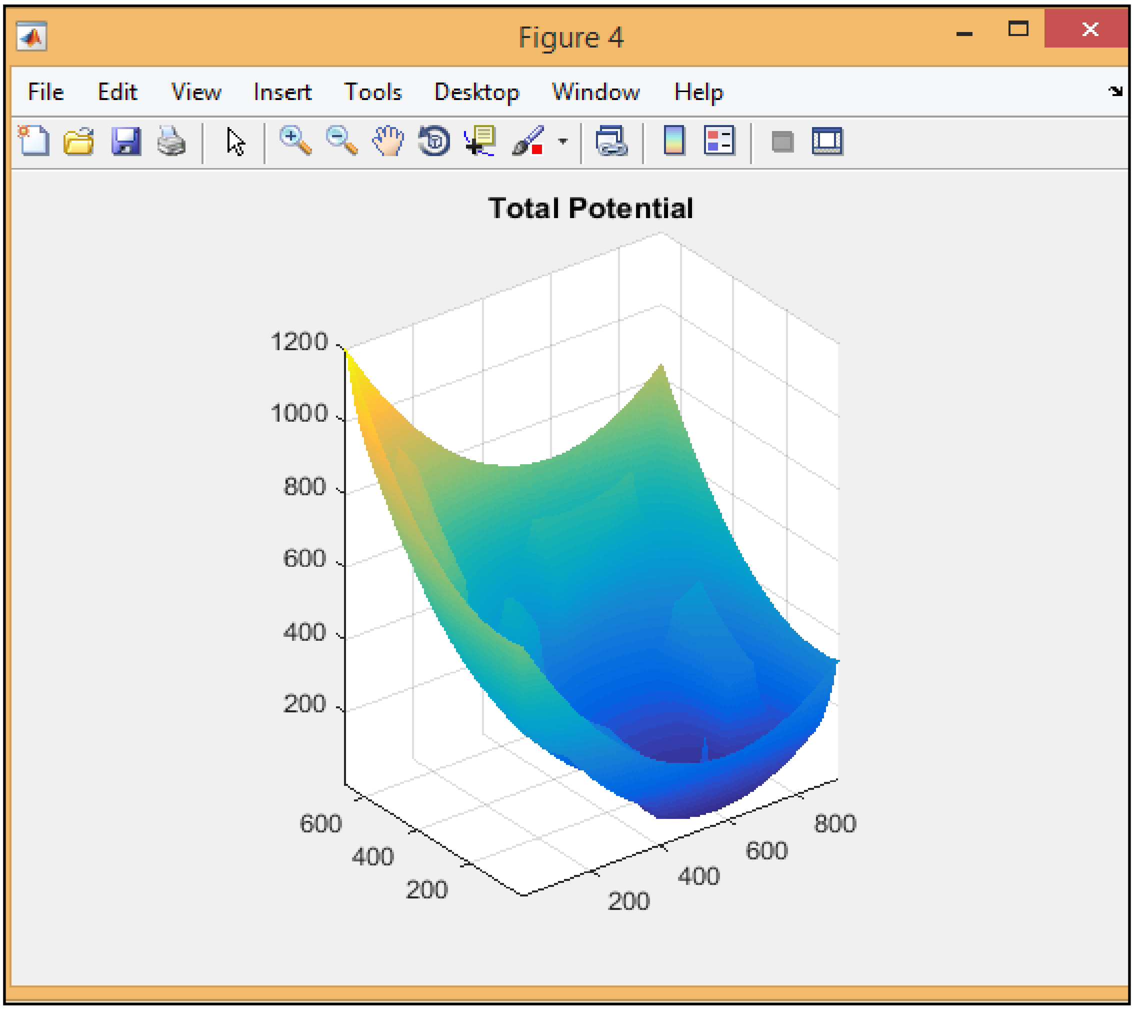

Figure3.7: total potential.

Similarly, the configuration space can be compared to an area. The beginning place and obstacles are in a higher area, while the finish point is in a lower area, among them. The robot is thought to be a sphere. The robot will then slide down a specific course from a higher starting point to a lower arrival point, avoiding taller obstacles, thanks to gravity. As illustrated in Figure total potential.

The additive potential functions' intuition is that they attract the robot to the objective position while repelling it from obstacles by superimposing these two effects into a single resultant force given to the robot. U(q) = Uatt(q) + Urep(q), which is the sum of the attractive and repulsive potentials, can be used to create this potential function.

```
%% Combine terms

f = attractive + repulsive;

figure;
m = mesh (f);
m.FaceLighting = 'phong';
axis equal;

title ('Total Potential');
```

Figure3.8: code Matlab total Potential

3.3.6. Robot move:

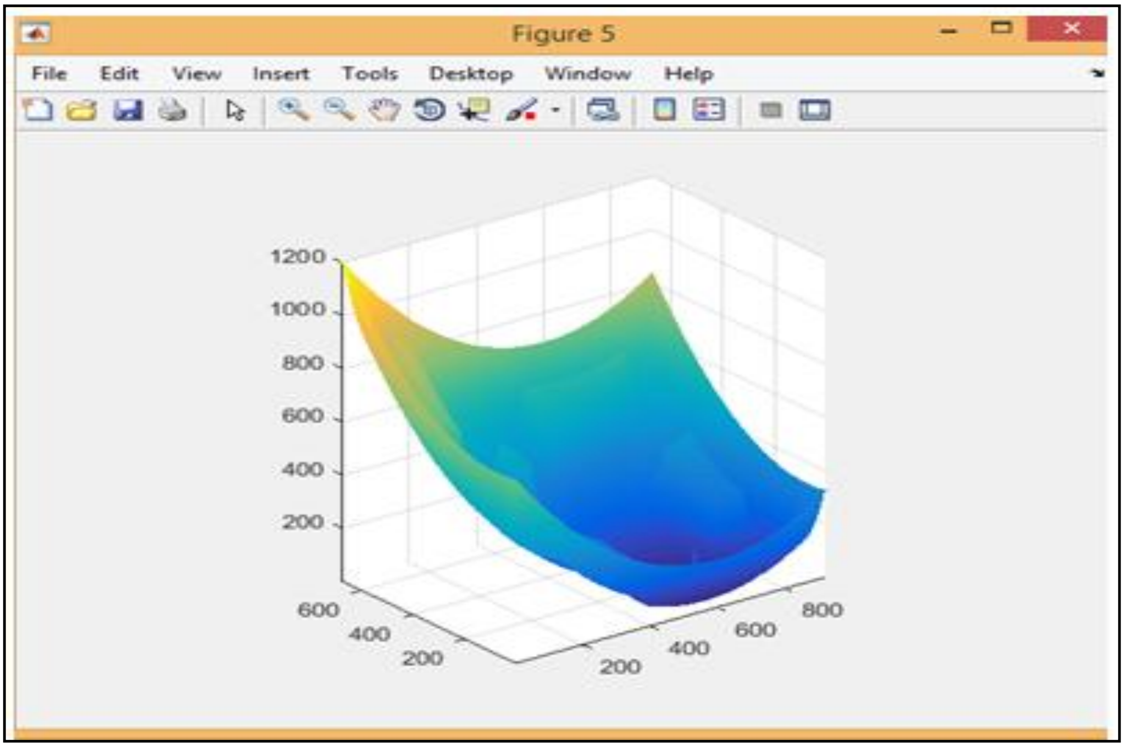

Figure 3.9: robot move

The beginning place and obstacles are in a higher area, while the finish point is in a lower area, among them. The robot is thought to be a sphere. The robot will then slide down a specific course from a higher starting point to a lower arrival point, avoiding taller obstacles, thanks to gravity. As illustrated in Figure 5. The following illustration depicts a robot moving in a two-dimensional environment with obstacles. The animation shows how the artificial potential field method works in detail.

The artificial potential field method generates a force field based on the motion environment, and the robot then follows the force field's guidance, beginning at the "top," avoiding the "small mountain peak" formed by obstacles along the way, and rushing all the way to the "foot of the mountain," where the target location is located.

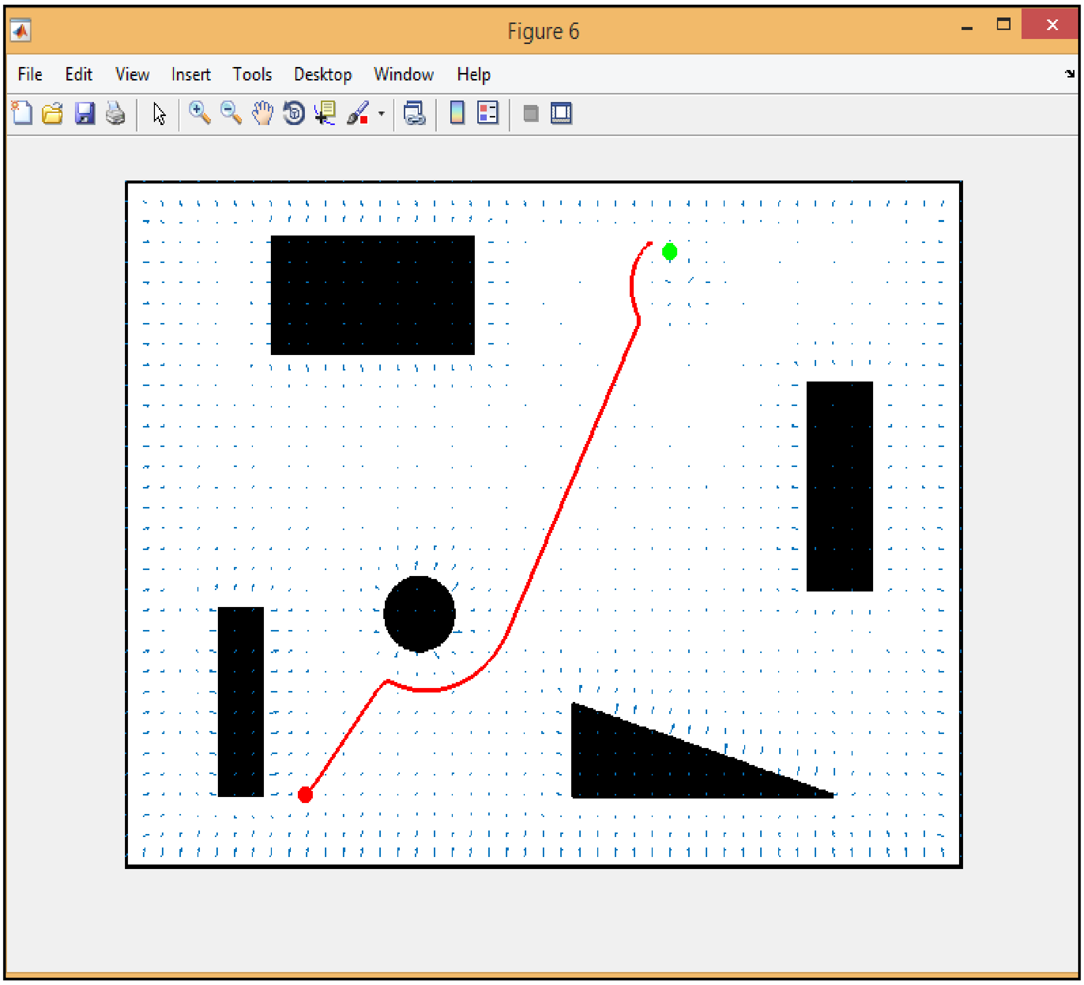

Figure3.10: Robot move from the beginning to the goal

This is the last figure that shows us how the robot moved from the beginning to the goal by avoiding obstacles.

3.4. Discussion

The diversity of work and perspectives has enriched the field of navigation. This makes it possible to define a set of criteria to ensure the quality of such a solution, and to show the performance of the Artificial Potential Fields for solving the shortest path problem, we made a small comparison between two other methods: fuzzy logic and artificial neural network, and noted that the convergence rate of the artificial potential field algorithm is more interesting.

But:

Artificial Potential Field does not always provide the best answers for issues with a high number of factors. APF may converge to local optimums rather than finding a global optimal. The emergence of local minimums in oscillation and blocking topologies is sensitive to this AI technique, and these minimums are often related to the geometry and distribution of obstacles.

3.5. Conclusion

In this chapter, we have tried to present the different stages to enable the robot to move. This method directs the robot to move from the starting point to the target by avoiding obstacles that may encounter it. We also found that the robot can be trapped in the local minimum in addition to the oscillations that result from, but this does not prevent that we presented a rather easy method and Simple to enable the robot to move.

Conclusion and Perspective

In recent years, the subject of navigation of a mobile robot in natural surroundings has piqued interest. Roboticists are attempting to gradually enhance the degree of autonomy of their robots until they reach complete autonomy and are capable of long missions.

The movement of a mobile robot in a given area is still a challenging challenge to tackle.

It necessitates the integration of functionalities that allow the perception, decision, and action cycle to be completed. To be able to deal with the vast range of scenarios that the robot may encounter while navigating.

In general, we used MATLAB in this work to illustrate the importance of intelligent systems not just for autonomous control of mobile robots. The MATLAB tools enable the modeling of events that are similar to human reasoning in some ways.

The Artificial Potential Field's implementation demonstrates that the goal was met by avoiding impediments. We've discovered that the Artificial Potential Field isn't always the best option; they can be tweaked to solve similar difficulties.

Following our work, a variety of perspectives are possible. As a result, it would be fascinating to incorporate "dynamics" into our navigation study, taking into account the obstructions that may pose particularly difficult issues.

This will allow us to detect a greater number and variety of obstacles, avoiding any navigational disruption caused by an unanticipated change in the surroundings.

References

[1] Schwartz, J, T., Sharir M., On the Piano Movers' Problem: I. The Case of a Two-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers. Communications on Pure and Applied Mathematics, vol. 36, pages 345–398, 1983a.

[2] Kruse, T., T. Kruse, A. Pandey, R. Alami and A. Kirsch, "Human-aware robot navigation: Asurvey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726-1743, 2013.

[3] Schwartz 1983c J. T. Schwartz & M. Sharir. On the Piano Movers' Problem: III. Coordinating the Motion of Several Independent Bodies. International Journal of Robotics Research, vol. 2, no 3, pages 97–140, 1983.

[4] Latombe 1991 J.-C. Latombe. Robot motion planning. Kluwer, Boston, MA, 1991.

[5] Hachour, O., 2008. Path planning of Autonomous Mobile Robot, International Journal of Systems Applications, Engineering & Development, vol. 2, no.4, pp.178-190.

[6] Zadeh, L., 1965.Fuzzy sets", *Information and Control*, 8(3):338- 353.

[7] Ouadah, N., Azouaoui, O., Hamerlain, M., 2005. Implémentation d'un contrôleur flou pour la navigation d'un robot mobile de type voiture, *Troisième Congrès francophone*, Majecstic, Rennes, France.

[8] Chatterjee, R., Matsuno, F., 2001. Use of single side reflex for autonomous navigation of mobile robots in unknown environments", Robotics and Autonomous Systems, Volume 35, Issue 2, pp 77-96.

[9] Xu, W, L., 1999. A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behaviourbased mobile robot, Institute of Technology and Engineering, College of Sciences, Massey University, Palmerston North, New Zealand.

[10] Mellouk, A., and Benmachiche, A., 2019. "A Survey on Navigation Systems in Dynamic Environments", In Proceedings of ACM ICIST conference, Hammamet, Tunisia - 27-28 December (ICIST '2020).

[11] Jun-Hao Liang, Ching-Hung Lee,“Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm”, Advances in Engineering Software, 79:47–56, 2015.

[12] Ippei Nishitani, Tetsuya Matsumura,“Human-centered X-Y-T space path planning for mobile robot in dynamic environments”, Robotics and Autonomous Systems, 66:18–26, 2015

[13] Xunyu Zhong, Xungao Zhong, Xiafu Peng, “Velocity-Change-Spacebased dynamic motion planning for mobile robots navigation”, Neurocomputing,143:153–163, 2014.

[14] S. Ge, Y. Cui, “Dynamic motion planning for mobile robots.

[15] A. Zou, Z. Hou, S. Fu and M. Tan, "Neural networks for mobile robot navigation: A survey," *Springer-Verlag Berlin Heidelberg*, pp. 1218-1226, 2006.

[16] Z. Laouici, M. Amine and M. Khelfi, "Hybrid Method for the Navigation of Mobile Robot Using Fuzzy Logic and Spiking Neural Networks," *International Journal of*

Intelligent Systems and Applications (IJISA) , vol. 6, no. 12, pp. 1-9, 2014.

[17] T Tan, S D Liu, Y Zhang. Simulation Research on mobile robot path planning [J]. Journal of Computer Simulation, vol.33, no.8, pp.354-358, 2016.

[18]. C C Hsu, R Y Hou, W Y Wang. Path Planning for Mobile Robots Based on improved Ant Colony Optimization. IEEE International Conference on Systems, vol.8215, no.2, pp.2777-2782, 2013.

[19] H Qu, L W Huang, X Ke. Research on robot path planning based on Improved Ant Colony Algorithm in dynamic environment. Journal of University of Electronic

Science and technology of China, vol.44, no.2, pp.260-265, 2015.

[20] Goldberg D.E., 1987. Genetic algorithms in search, optimization, and machine learning, Addison Wesley

[21] Yuce, B., Packianather, M., Mastrocinque, E.,TruongPham, D., and Lambiase, A., 2013. Honey Bees Inspired Optimization Method: The Bees Algorithm, *Insects 2013*, 4, 646-662; doi:10.3390/insects4040646.

[22] Yuce, B., Packianather, M., Mastrocinque, E.,TruongPham, D., and

Lambiase, A., 2013. Honey Bees Inspired Optimization Method: The Bees Algorithm, *Insects 2013*, 4, 646-662; doi: 10.3390/insects4040646.

[23]Karaboga, D., Basturk, B., 2007. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, *International Fuzzy Systems Association World Congress, IFSA 2007: Foundations of Fuzzy Logic and Soft Computing pp 789-798*

[24] A. Stacey, M. Jancic and I. Grundy, "Particle swarm optimization with mutation," in *The 2003 Congress on Evolutionary Computation CEC'03*, 2003.

[25] S. Pasupuleti and R. Battiti, "The gregarious particle swarm optimizer (G-PSO)," in *In Proceedings of the 8th annual conference on Genetic and evolutionary computation*, Washington, 2006.

[26] A. Ratnaweera, S. Halgamuge and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *In IEEE Transactions on evolutionary computation*, vol. 8, no. 3, pp. 240-255, 2004.

[27] J. Vesterstrom and R. Jacques, "Particle swarms extensions for improved local, multimodal, and dynamic search in numerical optimization," In Master’s thesis, Dept Computer Science, Univ Aarhus, Aarhus C, Denmark, 2002.

[28] P. Fourie and A. Groenwold, "The particle swarm optimization algorithm in size and shape optimization," *Structural and Multidisciplinary Optimization*, vol. 23, no. 4, pp. 259-267, 2002.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE Press Proceedings of the 1995 IEEE International Conference on Neural Networks*, 1995.

[30] J. Kennedy and W. Spears, "Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator", in *Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence*, 1998. 157

[31] J. Pugh and A. Martinoli, "Multi-robot learning with particle swarm optimization," in *International Conference on Autonomous Agents and Multi Agent Systems AAMAS06*, Hakodate, Japan, 2006.

[32] J. Pugh, Y. Zhang and A. Martinoli, "Particle swarm optimization for

unsupervised robotic learning," in *Proceedings of 2005 IEEE in Swarm Intelligence Symposium, SIS05*, 2005.

[33] Benmachiche, A., Bouhadada, T., Laskri M, T., Zendi, A. 2016. A dynamic navigation for autonomous mobiles robots, *Intelligent Decision Technologies*, ISSN 1875-8843 (E), 10 (1).

[34]XinShe, Y., 2005. Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK xy227@eng.cam.ac.uk.

[35] J. Latombe, Robot Motion Planning, Boston, MA: Kluwer Academic Publisher, 1991.

[36] Y. Hwang and N. Ahuja, "Gross Motion Planning - A Survey," *ACM Computing Surveys,* vol. 24, no. 3, pp. 219-291, 1992.

[37] M. d. Berg, O. Cheong, M. Kreveld and M. Overmars, "Computational geometry-algorithms and applications," *Springer-verlag,* 1997.

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

[39] C. Goerzen, Z. Kong and B. Mettler, "A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance," *Journal of Intelligent and Robotic Systems,* vol. 57, no. 1-4, pp. 65-100, 2010.

[40] Khatib, O., 1990. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, IEEE International Conference on Robotics and Automation, St. Louis, Missouri, pp.500-505, March 25-28.

[41]Andrews, J, R., Hogan, N., 1983. Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator, *Control of Manufacturing Processes and Robotic Systems*, ASME, Boston, pp. 243-251.

[42] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research* , vol. 5, no. 1, pp. 90-98, 1986.

[43]Ouadah, N., Azouaoui, O., Hamerlain, M., 2005. Implémentation d'un contrôleur flou pour la navigation d'un robot mobile de type voiture, *Troisième*

Congrès francophone, Majecstic, Rennes, France.

[44]Arkin, R, C., 1989. Motor Schema-Based Mobile Robot Navigation, the International Journal of Robotics Research, pp. 92-112.

[45] Adrian Biran et Moshe Breiner, *MATLAB pour l'ingénieur : Versions 6 et 7*, Pearson Education, 2004

Résumé

L'homme Depuis fort longtemps rêve de créer des machines intelligentes capable D'effectuer tache à sa place.

Les récents développements de la microélectronique mener le domaine de la robotique mobile vers dehorizons fort prometteurs.

Le développement de la robotique mobile est le développement des techniques de Commande; on cherche à intégrer des techniques de plus en plus performantes sur des structures de plus en plus compactes.Les méthodes et les techniques de navigation sont variées, citons les premiers méthodes sont les Champs de Potentiels Artificiels (APF). Cette méthode introduite dans les années soixante-dix sons principe est de construire un champ de potentiel sur l'environnement de navigation du robot on a pour les obstacles génèrent un champ potentiel répulsif et un champ attractive pour le but.

Abstract

Man has long dreamed of creating intelligent machines capable of Stain for him.

Recent developments in microelectronics are leading the field of mobile robotics towards very promising horizons.

The development of mobile robotics is the development of techniques for Ordered; we are trying to integrate more and more efficient techniques on more and more compact structures.

The methods and techniques of navigation are varied, let us quote the first methods are the fields of artificial potentials (APF). This method, introduced in the 1970s, is based on the principle of building a potential field on the robot's navigation environment; for obstacles generate a repulsive potential field and an attractive field for the goal.