People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Chadli Bendjedid University
Faculty of Science and Technology
IT department

الجمهـــوريـــة الجزائريـــة الديمقراطيـــة الشـعبيـــة
وزارة التعليـــم العـــالي والبحث العلمـــي
جامعـــة الشـــاذلي بن جـديـــد
كليـــة العـلـوم و التكـنـولوجيـــا
قســـم الاعـــلام الالـــي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

Master Thesis in Computer Science

Presented by

Aziza Merdaci

Specialty: Intelligent Computer Systems

Theme

A New Approach To Finding The Shortest Path Of a Mobile Robot Based On Artificial Neural Networks

Defence on : 04 /07/2021

Dr Bentrad S	**MCB**	**Univ.of el tarf**	**President**
Dr Benmachiche A	**MCA**	**Univ.of el tarf**	**Supervisor**
Dr Makhlouf A	**MCB**	**Univ.of el tarf**	**Examiner**

University Year : 2020/2021

Thanks

My thanks go first to God Almighty for the will, health and patience he gave me during all these years of study.

In particular, I would like to thank ***Dr. Benmachiche. A****, Senior Lecturer in the Department of Computer at Chadli Ben Djedid University, 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.*

I sincerely thank:

-Dr-Bentrad S, for having honored me to chair the jury

-Dr-Makhlouf A, for agreeing to be an examiner.

To the teachers of our university and especially those of the computer science department.

Thank you very much to all my family members, especially Dr. Diabi Mokhtar and his wife, Merdaci Nadia, and diabi azzedine for their interest in me, encouragement and patient follow-up during my studies. I also thank my colleagues, for the good times we had together.

I would like to thank, Melouk Ahlem has helped me in this work.

Finally, I thank everyone who helped me from afar or from the promoter.

Thank you all from the bottom of my heart

Dedications

First of all, I thank the good God the Almighty for the good health, the will and the patience that he gave me to accomplish this work.

I dedicate this work

In memory of my father that although I never knew him but he is still alive

in my heart and mind.

To the person who plays both roles at the same time and who replaced my father. who was always next to me and who was the cause of my smile and happiness to the most person dear and most loved in my life My mother "warda ".

To my twin "radja Merdaci" who is always there for me and who has always supported me.

To my sisters" Radia "and "Ferhouda" and my brothers "Djamel" and "Yazid"for their permanent encouragement and moral support.

To all my family specially Dr " diabi moukhtar" and my cousin " nedia "and her daughter amira for all the support they gave me during this year

To my dear neighbour " Abir'' who is always there for me and who has always supported me.

To all my friends" ghozlen" "fella" " warda" "yousra" " hana" " wafa" " nada" "rayen" "safia" "rania" "amira" "marwa" "wahida" "kawter"

To all the friends who encouraged me and who helped me in this work"bouchra""ahlem""Saif "walid" "iskander" .

To all my friends from the master2 class "SII"

To all those I love and those who love me.

Merdaci Aziza

Abstract

The design of artificial systems has undergone an epistemic revolution throughout the past fifteen years. It entailed transcending the traditional dichotomy between the organic and inorganic worlds by attempting to imbue machines with the adaptability and autonomy found in living systems.Evolutionary robotics strives to create devices that can learn new skills on their own in an ever-changing, uncontrollable environment. This technology allowed for the construction of genuine robots with complicated reactive behaviors. On the basis of this observation,

Automatic route planning for the robot without collision Environments, have been the subject of a great deal of research in recent years. Over the past twenty years, many planning techniques have been proposed, but None of these techniques is defined as a general method that can be solved Planning problem. In this work, we propose a new technique for robot trajectory planning, which It is based on artificial intelligence methods: intelligent neural networks . The proposed technique consists of environmental modeling Robot with neural networks, simulation results proved ,The proposed neural network path mapping method is accurateand efficient.

Abstract

Résumé

La conception des systèmes artificiels a connu une révolution épistémique au cours des quinze dernières années. Cela impliquait de transcender la dichotomie traditionnelle entre les mondes organique et inorganique en essayant d'imprégner les machines de l'adaptabilité et de l'autonomie trouvées dans les systèmes vivants. La robotique évolutive s'efforce de créer des appareils capables d'acquérir de nouvelles compétences par eux-mêmes dans un environnement en constante évolution et incontrôlable. Cette technologie a permis de construire de véritables robots aux comportements réactifs compliqués. Sur la base de ce constat,

La planification automatique d'itinéraires pour le robot sans collision Les environnements, ont fait l'objet de nombreuses recherches ces dernières années. Au cours des vingt dernières années, de nombreuses techniques de planification ont été proposées, mais aucune de ces techniques n'est définie comme une méthode générale pouvant être résolue. Problème de planification. Dans ce travail, nous proposons une nouvelle technique de planification de trajectoire de robot, basée sur des méthodes d'intelligence artificielle : les réseaux de neurones intelligents. La technique proposée consiste en un robot de modélisation environnementale avec des réseaux de neurones, les résultats de la simulation ont été prouvés.

Abstract

ملخص

شهد تصميم الأنظمة الاصطناعية ثورة معرفية على مدار الخمسة عشر عامًا الماضية. لقد استلزم تجاوز الانقسام التقليدي بين العالمين العضوي وغير العضوي من خلال محاولة إضفاء القدرة على التكيف والاستقلالية الموجودة في الأنظمة الحية على الآلات. تسعى الروبوتات التطورية جاهدة لإنشاء أجهزة يمكنها تعلم مهارات جديدة بمفردها في بيئة دائمة التغير ولا يمكن التحكم فيها. سمحت هذه التكنولوجيا ببناء روبوتات أصلية بسلوكيات تفاعلية معقدة. على أساس هذه الملاحظة ،

لقد كان تخطيط المسار التلقائي للروبوت بدون تصادم البيئات موضوع قدر كبير من البحث في السنوات الأخيرة. على مدار العشرين عامًا الماضية ، تم اقتراح العديد من تقنيات التخطيط ، ولكن لم يتم تعريف أي من هذه الأساليب على أنها طريقة عامة يمكن حلها مشكلة التخطيط. في هذا العمل ، نقترح تقنية جديدة لتخطيط مسار الروبوت ، والتي تعتمد على أساليب الذكاء الاصطناعي: الشبكات العصبية الذكية. تتكون التقنية المقترحة من روبوت النمذجة البيئية مع الشبكات العصبية ، أثبتت نتائج المحاكاة ، وطريقة رسم خرائط مسار الشبكة العصبية المقترحة دقيقة وفعالة.

Contents

List of Figures

List of tables

Abreviations list

- **AI** : Artificial intelligence
- **APF**:Artificial Potential Fields
- **GA** : Genetic Algorithm
- **VBA** : Virtual Bee Algorithm
- **BCO** : Bee Colony Optimization
- **BOD** : Dance Bee Optimization
- **ABC** : Artificial Bee Colony
- **RNN:** Recurrent Neural Network
- **MLP :**Multilayer Perceptron
- **BPTT :** Back-Propagation Through Time

General Introduction

Artificial Intelligence (AI) is a branch of computer science where machines are trained and engineered to mimic the human decisive and reactive functions without human intervention.

AI is the implementation of a number of techniques aimed at allowing machines to imitate a form of real intelligence. The need to use artificial intelligence is increasing in all areas.

In 1950; Mathematician Alan Turing developed a test to measure artificial intelligence. The Turing test, thanks to a series of questions, determines whether the answer can be determined by the machine in this way.If the computer's responses are indistinguishable from those of humans, the computer is considered to be artificially intelligent.

The purpose of artificial intelligence is to design systems capable of reproducing the behavior of humans in their reasoning activities. AI sets as its goal the modeling of intelligence taken as a phenomenon (as well as physics, chemistry or biology which aim to model other phenomena).

The scientific field that specializes in studying, designing and implementing "smart machines" is called artificial intelligence. It should first be remembered that the word "machine" does not designate a physical object but rather an automatic system capable of processing information.

With the advancement of computer science and technology, a wider area of research has opened in the area of robotics, which is devoted to machines that perform movements in space. So when we talk about "robot" in artificial intelligence we are referring to a computer program showing some form of intelligence.

A robot is a machine equipped with the capacities of perception, decision and action that allow it to act autonomously in its environment.

Robots are widely used to help or possibly replace humans in several areas: Industrial, military, surgical, as well as in everyday life where the robot performs cleaning and maintenance tasks or provides assistance to a person disabled.

A robot endowed with the ability to perceive and provide information on its environment, must be able to move independently, while avoiding obstacles.

The automatic planning of the trajectory in robotics is used to find a series of valid movements that allow a robot to go from an initial state to a desired end state.

In general, a valid movement is a movement that does not produce collision and which respects the kinematic constraints of the robot.[1]

For twenty years, many automatic planning techniques trajectories have been proposed, nevertheless none has really established itself as a general method that can satisfactorily solve the planning problem (finding a trajectory, minimize the cost function), especially at the industry level. First approximation, we can distinguish two types of methods: those which seek to construct are presentation of free space (global methods) and that based on information local to incrementally build a trajectory (local methods). [1]

Our route planning method offers a safe and efficient solution to fundamental problem of planning and even the problem of optimization. In our work, the path is defined from the modeling of the robot's environment by the artificial neural networks.

The objective of this study is to study the techniques of artificial intelligence as a solution for this problem and to discover the benefits of these methods compared to other methods already developed for the resolution of this problem. 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 is taken up to examine the typology of mobile robots. We present a state of the art of navigation in general and cooperative and autonomous navigation in dynamic environments with obstacle avoidance in particular.

In the second chapter, the contribution to the research problem, we present a conceptual study on finding the shortest path for an autonomous robot using neural networks artificial. We present a conceptual study, the proposed solutions which are detailed based on an analysis of existing systems.

In the last chapter contains the interfaces we have created for the planning of the trajectory by neural networks, simulation environments are briefly discussed. Several experiments are carried out with a MATLAB development environment to evaluate the performance of our artificial neural networks. Examples of simulations are provided to highlight the results of the proposed methods to find the shortest path for an autonomous robot.

Finally, we conclude and summarize the entire thesis work, explaining the conclusion obtained through simulated experiments. The possible scope of the future work of this thesis is mentioned.

Chapter 1: Navigation Systems

1 – 1 Introduction:

A mobile robot may be a mechanical, electronic, and computing system that concretely acts on its environment to deal with an objective that has been particularized thereto. This machine is flexible and may adapt to certain distinctions of its operating conditions. it's the role of perception, decision, and action. So, the robot should be ready to perform various tasks, in some ways, and complete them properly, albeit it encounters new situations abruptly. In this study, we have an interest in autonomous mobile robots.

A robot is claimed to be autonomous: it's ready to choose its actions to succeed in its goal and if it's ready to perform the task correctly and encounters new unexpected situations without human intervention.

In this thesis, we'll present the matter of navigation in mobile robots. We determine the various navigation strategies and offer the categories of existing controllers. Navigation is defined because the process of answering the subsequent three questions [2].

i) Where am I?

ii) Where are the opposite places in reference to me?

iii) How am I able to reach these other places from where I am?

So navigation is [3] that the set of techniques and methods for knowing the situation (coordinates) of mobile with reference to a coordinate system, or with reference to a hard and fast point. Calculate or measure the trajectory to follow to succeed in another well-known coordinate point by submitting some constraints and criteria that arise from several factors, which usually depend upon the characteristics of the robot, the environment, and therefore the sort of task to run. Calculate any information like the movement of this mobile (distance and duration, speed time period, estimated time of arrival, etc..).

During this section, we discuss autonomous navigation techniques for mobile robots in a dynamic environment.

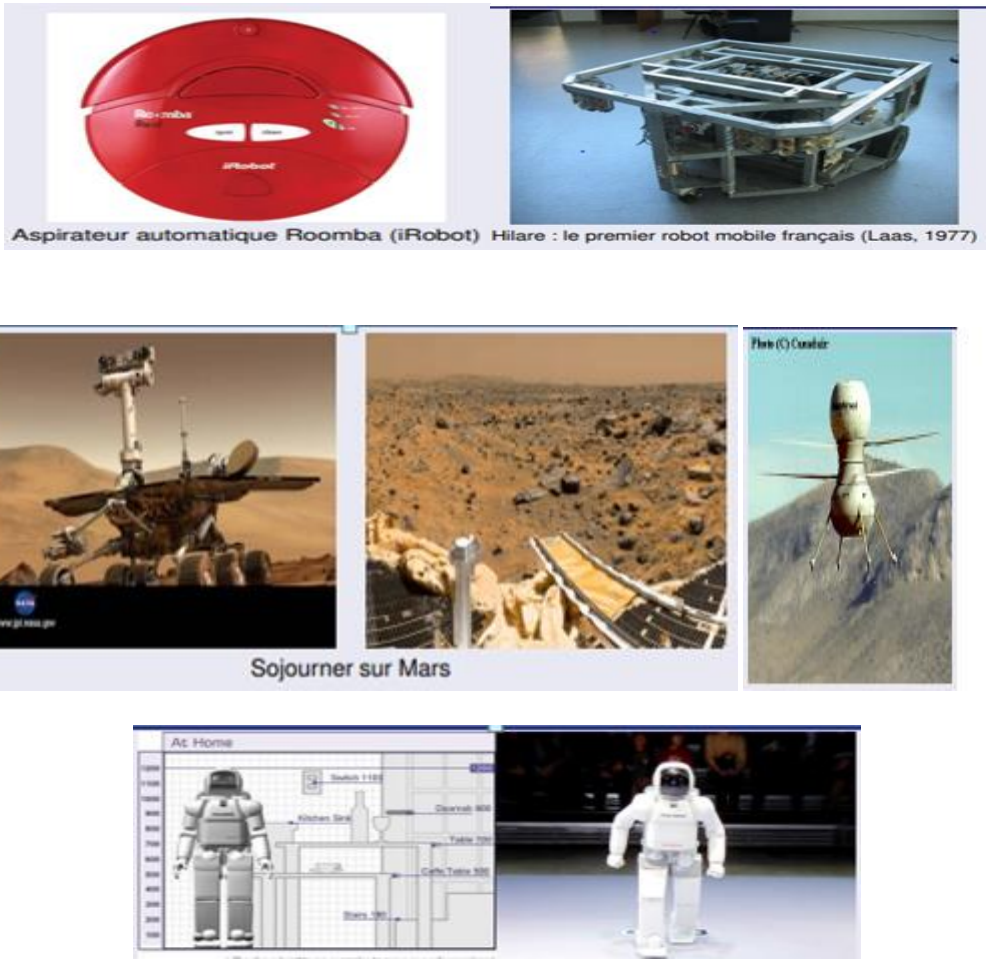


Figure 1.1: Examples of Decent Robots. [43]

1.2. Navigation Strategies:

The navigation strategies allowing a mobile robot to maneuver and to succeed in a goal are excessively diverse, as are the classifications which will be made from them.

1.2.1 Artificial Potential Fields (APF):

APF is conceived so as to create a field of potentials on the robot's navigation environment. The worth of this field is minimal on the purpose that the robot must reach and continuously grows together move faraway from that time. The obstacles generate a repulsive field of potential for the robot, useful superior to the other point, not like an obstacle of the sector potential, and therefore the repulsion field extends around obstacles with intensity inversely proportional to space from the obstacle.

The thought is then to ask the robot to maneuver within the direction of the strongest negative potential gradient on the general potential field obtained.

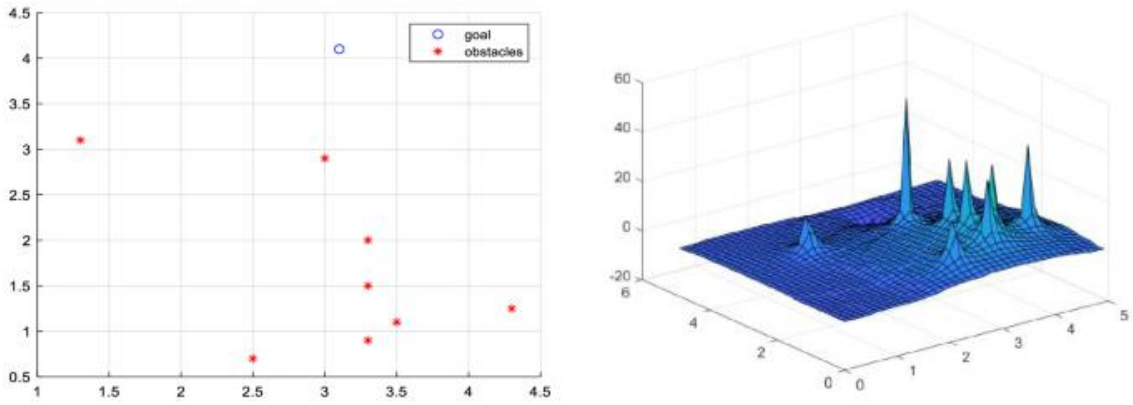


Figure 1.2: Map of potential fields around an obstacle. [44]

In [4], and in [5], the authors were the primary ones to imagine the thought of imaginary forces working on the robot. These methods have the advantage of being simple to implement, and that they were the primary to be physically implanted on real robots in 1985 by Brooks [6] and in 1989 by Arkin [7]. Despite the pc equipment still limited at this point, trajectory /control calculations being in no time with this sort of approach, it allows the primary experiments on relatively slow robots. In [8], Agirrebeitia proposes an extension of the principle of APF methods for robot navigation during a 3D space. The experiment revealed some recurring problems associated with the very principle of those methods:

- Local minima causing situations where the robot is trapped (typically the U-shaped trap).
- No passage detected between obstacles close enough.
- Oscillations caused by obstacles and narrow lanes.

In [9], authors in the field of mathematics have demonstrated the instability (causing oscillations) of these methods, and these problems are particularly prominent when implementing these methods on "fast" systems.

1.2.2 Fuzzy Logic:

Binary logic has the advantage of simplicity but is way faraway from the human way of reasoning. If one takes the instance of the qualification of the proximity of an obstacle, the symbolic logic makes it possible to involve notions like "near enough" or "very far", rather than being limited to a binary definition « obstacle or no obstacle ». This was formalized by Zadeh in 1965 [12].

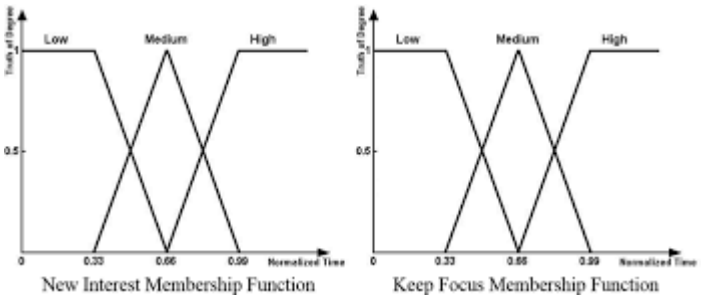
The principle of a controller supported symbolic logic comes in 3 phases: A fuzzification step, which can transform the input variables into fuzzy variables;

A step employing a table of rules of behavior, logical rules of the sort "if (condition 1) And/ Or (condition 2) then (action on the outputs)";

A last, the defuzzification step translates the action determined by the principles of behavior in command to send to the actuators.

The fuzzification stage uses fuzzy intervals, which can delimit the space of the input variables during a certain number of fuzzy subsets (for example for proximity, we will have very close (contact), quite close, average distance, far enough and really far); membership functions are then wont to define the degree of truth (probability of belonging) of the fuzzy variable as a function of the input quantity.

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



New Interest Membership Function Keep Focus Membership Function

Figure 1.3: Membership functions [45].

The second step is that the development of rules of behavior for the robot, following the mixture of fuzzy input variables. The principles of the behavior table are built manually and are hooked into the experience of the one that will adjust the controller. For a mobile robot, a rule can be: "if there's asonably close obstacle on the proper, you've got to show left and reduce the speed of the robot".

The last step is to rework the behavior obtained by the principles table, in command of the robot.

The centers of gravity approach, which entails taking a weighted average of the instructions to be implemented, is one way that can be used to do this. The weighting is based on the probabilities of each input variable's membership. In [13] the authors investigated a fuzzy

controller for navigation of a robot-car-type mobile robot with dual steering. The robot can reach I with the help of this controller work is underway to integrate the consideration of the final orientation of the vehicle. In [14], the authors use the right/left symmetry of the logic rules of behavior of the robot to simplify these, and thus, reduce the calculation time.

The problem with these fuzzy logic methods is similar to that of the APF approaches, namely the problem of local minima, which allows the robot to remain caught in dead ends.

In his paper [15], Xu proposes a method for escaping these traps. This method employs virtual targets to pull the robot out of the trap it has fallen into as soon as it realizes it has fallen into one.

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 [16].

1.2.3 Ant Colony:

Ant colony algorithms are based on the behavior of ants when seeking a path between food sources and their colonies. When an ant moves down a road, it leaves a trail by creating a substance known as "pheromones." Other ants find this subject appealing, so they follow in the footsteps of pheromones.

The presence of this matter is proportional to distance, and the longer the distance, the lesser the concentration of matter. If the journey is short, we see significant traces of this material, and the ants will follow the strongest trace, indicating a shorter path.

The goal of ant colony algorithms is to find the best path for mobile robots. These algorithms require a group to be realized, and they are employed for a task that requires numerous mobile robots to work together.

The goal of this research methodology was to mimic and develop the way ants [17] navigate using complex combinations of information, as a vector of integration and visual signals, on a simulator, then a real robot. The purpose was to solve the navigation problem specifically.

As a first step, a simulator was created to analyze directly inspired navigation strategies in ant behavior. The simulation's main goal is to enable the rapid examination of biological hypotheses before moving on with a physical implementation. Then, because the simulation does

not allow for all parts of the study to be taken into consideration, some components of the study are simplified, such as the environment or physical behavior of the robot; the researchers have developed a robot able to really test the strategies of displacement.

In an obstacle-filled environment, the robot must first detect and then avoid the obstructions. If the robot has already crossed a certain point, it will be able to predict the next step without getting distracted by the vast amount of data generated by the presence of several barriers. If the robot-ant comes upon a particularly long obstacle it should not be considered as impassable even if no issue appears on the rangefinder that we use to prevent the robot to hit the obstacles present in its environment. It must go along and around it by taking the direction provided by its integration vector being updated.

They created the initial modeling based on the ants' spontaneous realization of the vector of integration (direction of the nest, distance) during their first forays out of the nest:

- If the robot-ant encounters no obstacles, it will continue in a straight line from its starting position, following its integration vector.

- If the robot-ant comes across a close obstacle in a weak angular sector (Figure 2), it will take the free direction "closest" to the integration vector, allowing it to avoid the obstacle.

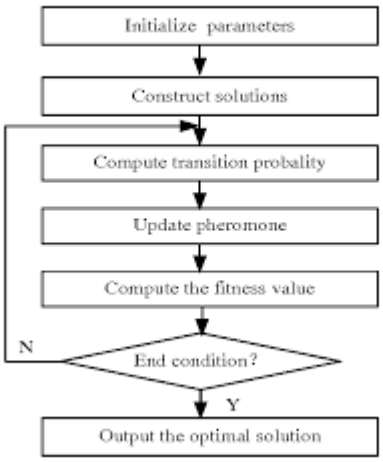


Figure 1.4: Ant Colony [46]

The robot-ant can memorize couples (Perception, Action) and thus construct a basic representation of space.

The action may correspond to the movement to be done, and the perception of the value of the integration vector and/or the perceived image (direction, distance to be traveled).

In a way, these pairings are the insect's analogs of the local vectors stated before. We define the notion of point of choice for each place where the robot-ant must make a decision in order to minimize the stored information and hence be as realistic as possible.

This is the case, for example, for barriers to avoid. Then, the robot-ant can use these pairings (perception, action) by comparing them to the current circumstances. The move from regular training to automatic utilization of the routine is then made gradually.

TAs a result, the robot-ant is making increasingly rapid movements, because once the routine has been learned and "automated," it saves time in the perceptual phase throughout this type of voyage. The robot then only uses its sensors to readjust itself at each memorized point of choosing, rather than throughout each basic movement. The trajectories are more and more smooth and efficient, which corresponds to the behaviors we observed in ants.

Ant Colony algorithms are very robust (they are always efficient, even if some people fail), versatile (a colony may adapt to a new environment), and quick (they support parallel processing and the use of heuristic information, among other things). However, there are numerous drawbacks. However, there are some drawbacks: we can enter a blocking condition, runtime can be lengthy, and it does not apply to all types of situations.

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.

1.2.4.1 Virtual Bee Algorithm:

Although only functions with two parameters have been given as examples, this approach was invented by XinShe Yang in 2005 [08] for solving numerical optimization issues. It can optimize functions and discrete issues. The VBA method begins with a virtual bee troop; each bee wanders randomly into the search space, which in most situations is a 1-D or 2-space -D. The virtual functions for optimizing functions are the primary steps of the bee algorithm:

- The creation 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).
- Definition of a criterion for communicating direction and distance in a manner comparable to bees' physical abilities (the dance of the bees).

- Update a population of people 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.
- Decoding the results in order to arrive at a solution to the problem.
- Decoding of the results to obtain the solution of the problem.

1.2.4.2 Bee Colony Optimization

This algorithm was introduced by yuce et al in 2013 [18] in order to find the optimal solution for a given difficult combinatorial optimization problem, such as the problem of a commercial traveler, the problem of p-Median, problem routing in optical networks.

Every bee comes up with a solution to the problem. To construct one step in the BCO algorithm,there are two stages that alternate (step forward and step back). Each artificial bee visits Nsolutions, generates a partial solution, and then returns to the hive with each step ahead.

The bees congregate in the hive and take the first move backwards. When all of the solutions have been completed, the best of them is chosen and used to update the best overall solution, completing an iteration of BCO. All of the solutions are deleted at this phase, and a new iteration is produced. Let 'B' represent the number of bees in the hive, and "NC" represent the number of positive steps forward. All bees are in the hive as the search begins. The BCO algorithm's pseudo-code can be summarized as follows:

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

- For each bee: at. k = 1

a. Count the constructive moves forward)

b. Evaluate all possible steps;

c. Choose a step;

d. k = k + 1;

If k ≤ NC, Go to b. Return of all the bees to the hive;

- For each bee evaluate the value of the objective function.

- Each bee decides randomly either to continue its own exploration.

- Recruiter, or become the bee who does the harvest. For each follower, choose a new solution from the recruiters.

- If the solutions are not complete, go to step (b).

- Evaluate all the solutions and find the best one among them.

- If the stop criterion is not checked, go to step (b), otherwise, go to the next step.

- Show the best solution found.

1.2.4.3 Dance Bee Optimization

Laga and Nouioua created the BOD (Dance Bee Optimization) method in 2009 [18] to handle the problem of T-coloring graphs. This algorithm was inspired by the foraging activity of bees. The method begins by placing the n bees in the search space at random.

After evaluating the fitness features of these bees, the bees with the best fitness (elite bees) are chosen for neighborhood building. In the next step, the algorithm guides the search in the vicinity of the best sites m found by elite bees. Indeed, these are the bees recruited to search around the best sites e, ie. Follow the best dancers, are also recruited bees pursue the other dancers. The primary operation of the BOD algorithm is recruiting. We assign a neighborhood meta-heuristic to each recruited bee (solution) to search around it.

In the end, only the best of the m bees (solutions) in each neighborhood are retained to build the next population. This limit is established in the algorithm to decrease the number of solutions to investigate; there is no similar limitation in nature.

The remaining bees are generated at random to complete the population. In 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 of 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.

1.2.4.4 Artificial Bee Colony

Karaboga and Basturk invented the ABC (Artificial Bee Colony) algorithm in 2007 [19], which examined the behavior of real bees to discover a food source, termed nectar, and share information from food sources with other bees in the nest. Artificial bees are defined and categorizedinto three classes in this algorithm: Bees (food-seeking bees), spectators (observation bees), and scouts are in charge of discovering new foods (the new nectar source). Only one type of bee is used for each food source. The number of worker bees equals the number of food sources, in other words. If a worker bee at a site is unable to locate the food source, she must act as a Scout, searching for new food sources at random. Employing bees share information with hive visitors so that visitors can select a food source to investigate.

Among the advantages of the bee colony method, mention may:

-Very effective in finding optimal solutions.

-overcomes the problem of the local optimum.

-Easy to implement.

-The use of several adjustable parameters.

-Sensitive to extremely difficult problems.

However, there are a number of drawbacks, such as most optimization algorithms having an evolution and diversification mechanism, incrementing a counter for solutions that do not improve and do not reach a threshold limit, and adjusting this parameter is a challenge in and of itself. , and small values can eliminate a solution before exploring its neighborhood incomplete, while large values may trap the algorithm in minima premises for several cycles.

1.2.5. Genetic Algorithm:

Genetic Algorithms belong to a family algorithm called Evolutionary Algorithms [20]. These algorithms are random optimization techniques inspired by Darwin's theory of evolution, which aim to find an approximate solution at the correct time. Genetic Algorithms use techniques derived from genetics and natural evolution: crosses, mutations, selections, etc ..., they represent a stochastic optimization method of order "0", which means neither continuity nor differentiability is necessary for the smooth running of the method, only the knowledge of the value where the proximity of the function to be optimized is sufficient. So, the effectiveness of a genetic algorithm depends on the good knowledge of the problem to be treated.

Genetic algorithms are the result of research by John Holland and his colleagues and students at the University of Michigan who, as early as 1960 [21], worked on this topic. The novelty introduced by taking into account the crossover operator in addition to the mutations in this operator allows us to get closer to the optimum of a function by combining the genes contained in the different individuals of the population.

Genetic algorithms are based on the notion of natural selection and apply it to a population of solutions to a given problem.

The stages of execution of a genetic algorithm:

- 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.
- Fitness function: Measure the fitness of each chromosome in the population.
- Selection: Create a new population with repetition next steps until the population is complete.
- Crossover and mutation: Each pair generates two children, in these operations two chromosomes exchanging one or more parts for data of the new chromosomes. If there is no mixing, the result is an exact copy of the parents.
- Mutation means that a gene in a chromosome can substitute for another in a random way.
- Stop test: If this criterion is not checked then go to step (2).

Figure 1.5: Genetic Algorithm**.** [47]

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

- The path found is not optimal: The impossibility of being sure that the solution found is the best one even after a significant number of generations. We can only be sure that we are getting closer to the optimal 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.

- 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 problem, which is tied to the principle of the algorithm itself and has no link 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 has various advantages over traditional optimization techniques, including:

Using only the objective function: 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.

GAs uses the coding of parameters, not the settings themselves. Chromosomes have a binary representation. This choice makes them intuitively applicable to all the problems whose solutions can be transposed into binary. The chromosomes are then represented by a chain of bits. This representation is independent of the problem and makes the genetic algorithm all the more robust.

Generating a form of parallelism by working on several points at the same time (population size N) at the place of a single iterated in classical algorithms, GAs work on a population of points instead of one.

The use of probabilistic transition rules (probabilities of crossing and mutation), unlike deterministic algorithms where the transition between two successive iterations is imposed by the structure and nature of the algorithm. This use makes it possible in some situations for genetic algorithms to avoid local optimums and to move towards a global optimum.

1.2.6. Artificial Neural network:

The formal abstraction of the behavior of biological neurons leads to artificial neurons [10]. It's a perfect solution to some problems that are highly thought-provoking, or highly complex.This is thanks to their brain.A neuron may be a nonlinear function, parameterized, with bounded value. A neuron contains 2 main elements:

- The weights related to neuron connections.
- An activation function, the input values are multiplied by their corresponding weight and summed to get a sum U_i.

$$Ui = E(x1, \dots., xj, \dots. xn) = \Sigma Wij \, x \, j \qquad (1)$$

Learning is often understood as a change within the capacity or behavior of an organism caused by experience [11]. The training algorithm will formulate the specific rules that allow it to generalize.

The formulation of the principles is completed by the change of the synaptic weights which results in the change of the behavior of the network; the change is administered by a group of iteration which makes these networks ready to react to new situations supported by the experience passed.

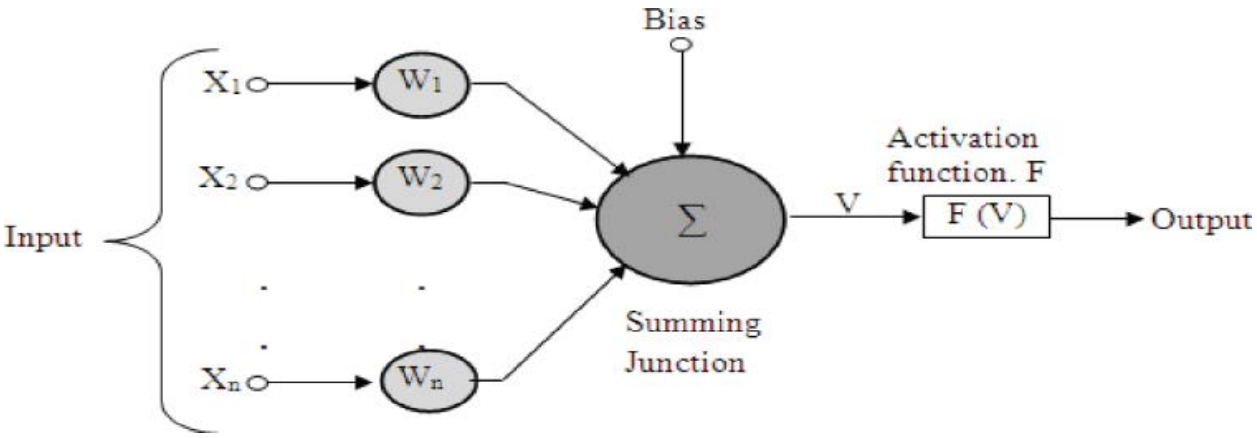


Figure 1.6: Artificial Neural network. [48]

The neural understanding of data is different; such knowledge isoften within the following form:

The optimal path is found by the synaptic weight vector W, hence a special coding of data. This representation of data is closer to the machine language, which makes the interpretation difficult for the person as compared with the opposite methods of representation of data.

For this reason, the networks of neurons are called a recorder. During alone amongst one in every of one among the goals of the research is to know how knowledge is distributed within a neural network and the way to extract that knowledge in a comprehensive way from a person's being.

The neural network approach may be a method for modeling intelligence. These networks are ready to solve problems in several areas. Their specialty lies in their brain. However, the approach suffers from a serious problem: the tactic of representing knowledge.

1.2.6.1. Recurrent neural network

A recurrent neural network (RNN) is a type of artificial neural network in which nodes are connected in a directed graph that follows a temporal sequence. This enables it to behave in a temporally dynamic manner. Derived from feed forward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs [22] [23] [24].

The name "recurrent neural network" is loosely applied to two broad kinds of networks that have a similar overall structure, one with limited impulse and the other with infinite impulse. The behavior of both types of networks is temporally dynamic [25]. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feed forward neural network, while an infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled.

1.2.6. 1.1.Back-propagation through time

Back-propagation through time (BPTT) is a gradient-based methodology for training recurrent neural networks of various types. Elman networks can be trained with it. Several researchers developed the method independently [33], [34], [35].

For training recurrent neural networks, BPTT is substantially faster than general-purpose optimization approaches like evolutionary optimization.[36]

1.2.6. 2.Multi-Layer Perceptron

The multilayer perceptron is a directed network of artificial neurons organized in layers and or the information travels in one direction only, from the input layer to the output layer.

Figure 1 shows an example of a network containing an input layer, two hidden layers, and an output layer. The input layer always represents a virtual layer associated with system inputs. It does not contain any neurons. The following layers are layers of neurons. In the example illustrates, there are 3 entries, 4 neurons on the first hidden layer, three neurons on the second and four neurons on the output layer. The outputs of the neurons of the last layer always correspond to the outputs of the system.

In the general case, a multilayer perceptron can have a number of layers of conque and a number of neurons (or inputs) per layer also any.

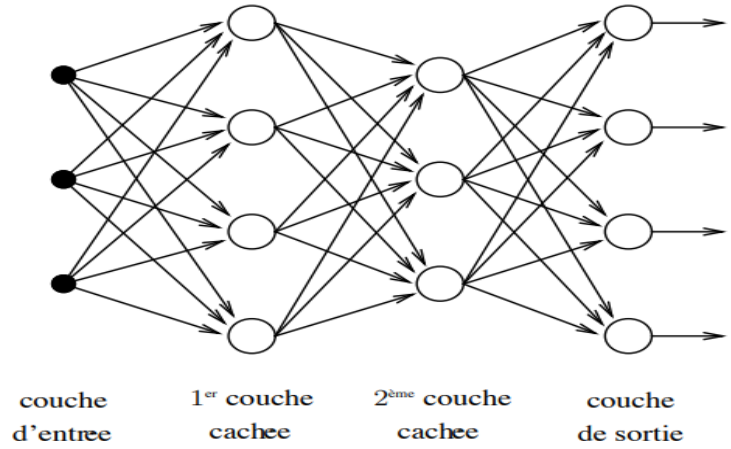


Figure 1.7: Example of a multilayer perceptron type network [26]

Bouhlassa also used this work offers a new technique for planning the trajectory of a robot, which is based on two methods of artificial intelligence: Neural networks. The proposed technique consists of modeling the environment of the robot by neural networks [1]

Araùjo and al, used this work "A Neural Network for Shortest Path Computation '' A new method to solve the shortest path problem was proposed using a two-layer Hopfield Neural Network. This solution aims to achieve an increased number of succeeded and valid convergences, which is one of the main limitations of previous solutions based on Neural Networks. Additionally, in general, it requires fewer neurons. The experimental results show that the main goal of the architecture is accomplished making it almost totally reliable (i.e., it achieves succeeded and valid convergences almost always) while considerably improving computational performance at the expense of the results, which are, only slightly worse. Two open issues deserve further work: first, the convergence to sub-optimal results; second, the adaptability to external varying conditions, in particular, to different graph topologies, which may not be trivial to achieve with the proposed architecture [32].

Belkhouche used this work trajectory planning and obstacle avoidance by neural networks for robots autonomous mobile [37].

1.2.6. 3. Reinforcement learning

Reinforcement learning (RL) is a branch of machine learning that studies how intelligent agents should operate in a given environment to maximize the concept of cumulative reward. [27] Reinforcement learning, along with supervised and unsupervised learning, is one of the three main machine learning paradigms.

Reinforcement learning (RL) is a branch of machine learning that studies how intelligent agents should operate in a given environment to maximize the concept of cumulative reward. [27] Reinforcement learning, along with supervised and unsupervised learning.

1.2.6. 3. 1. Q-learning

A model-free reinforcement learning algorithm is used to learn the value of an action in a certain state. It can handle problems with stochastic transitions and rewards without requiring adaptations and does not require a model of the environment (thus "model-free").

Q-learning provides an optimal policy for any finite Markov decision process (FMDP) by maximizing the anticipated value of the total reward across any and all successive steps, beginning from the present state. [29]

Given infinite exploration time and a partly-random policy, Q-learning can determine an optimal action-selection policy for any given FMDP. [29] The function that the algorithm computes – the expected rewards for an action taken in a given state – is referred to as "Q." [28].

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy derived from Q (e.g., ε-greedy)
    Take action a, observe r, s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$
    $s \leftarrow s'$;
  until s is terminal

Figure1.8: The Q-learning algorithm [49]

1.3-Comparison between Methods

Method	Advantages	Disadvantages
Fuzzy logic	- It is more simple and flexible. - Able to mange trouble with inaccurate data. - Not more expensive for development.	- The navigation path is not optimal because of the approximate reasoning method. - Requires the availability of an expert - Robot operation is limited by these rules
Neural networks	- A neural network can perform tasks that a linear program connot. - Ability to make machine learning.	-The neural network needs training to operate. -Requires high processing time for large neural networks.
Ant colony	- Can be used in dynamic applications. - Positive feedback leads to rapid discovery of good solutions.	- Convergence is guaranteed, but time to convergence is uncertain. - Coding is not straightforward. -Does not apply to all types of problems.
Bees Colony	-Effective in finding the best solutions and easy to implement - Get rid of the local optimum problem. - Sensitive to complex problems.	-It has a mechanism of evolution and diversification. - Big values may trap the algorithm for multiple cycles and it can eliminate a solution before exploitation.
Genetic Algorithm	- AGs use the encoding of parameters. - AGs are working on a population of points. - Same encoding – change the fitness function. - Have some GA; just write new chromosome to solve another problem.	-Computational time. -Slower than some other methods. - Choosing encoding and fitness function can be difficult.

Table1.1: Comparison of the Methods.

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 various steps; we've also provided the benefits and drawbacks of each strategy, and we've discovered that no navigation methodology has been observed that provides a solution that solves all difficulties. As a result, we can see that navigation is a very dynamic subject of study, with new approaches appearing on a regular basis. The necessity of applying heuristic approaches to treat the motion computation problem as an optimization issue in order to meet several constraints at the same time is revealed by this overview of the literature.

In the next chapter, we propose a new method of robot trajectory mapping based on artificial neural networks and the q-learning algorithm.

Chapter 2: System Design

2.1. Introduction:

Mobile robotics is the industry related to creating mobile robots, which are robots that can move around in a physical environment. Mobile robots are generally controlled by software and use sensors and other gear to identify their surroundings. Mobile robots combine the progress in artificial intelligence with physical robotics, which allows them to navigate their surroundings.

In terms of research, the development of applications requiring sharing of the robot's environment with its environment leads to look for reliable and scalable architectures far removed from the fixed architectures of classic robots whose immediate environment is closed to the human operator.

Automatic planning of the trajectory of collision less robots in Environments has been the subject of a very significant amount of research, in recent years.

Over the past twenty years, several planning techniques have been proposed, but very few of these techniques are effective in satisfactorily solving the problem of planning.

Our trajectory planning method, offers a safe and effective solution to the fundamental problem of planning and even the problem of optimization. In our work, the path is defined from the modeling of the robot environment by the neural networks artificial

2.2. Our Problem:

Our problem is to find the shortest path for an autonomous mobile robot, where the goal is to look for optimal plans to achieve the desired end state so that the cost is minimized. To find optimal solutions to large-scale planning problems and avoid obstacles, we study how artificial neural networks

How can a robot optimally move from a starting point to a breakpoint and avoid obstacles ?

2.3. The proposed solution:

The research focuses on adapting existing and constructing new efficient neural networks artificial to handle large scale optimization issues of tasks applied to a mobile robot in order to answer the research issue.

2.3.1. Problem definition

A robot must reach an end point from a start point, the robot has a starting position (Xs, Ys) and an end position (Xa, Ya), the robot does not know the environment.

A robot Training with neural networks artificial (RNN +MLP), we were interested in solving the following problems: planning a path for this robot to get from the starting point to the end point safely. We were interested in solving the following problems: Finding the optimal trajectory (the shortest path) and avoiding obstacles.

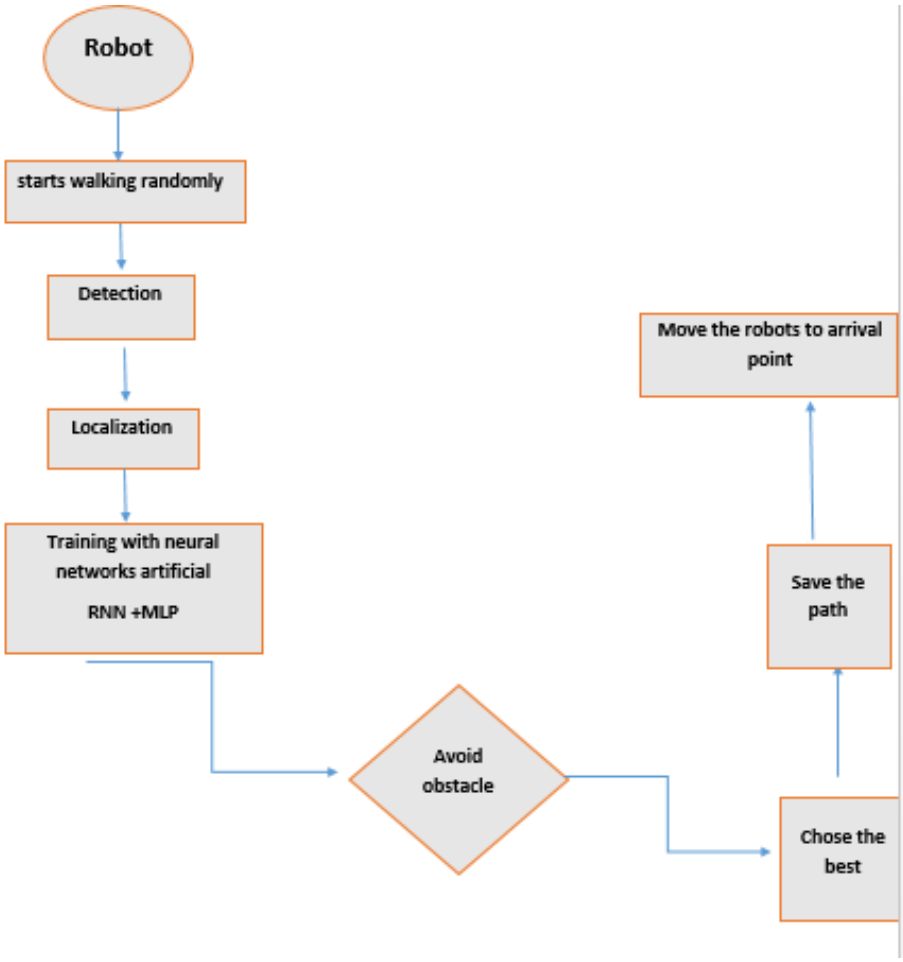


Figure 2.1: General Architecture of our system.

2.3.2. Use case diagram:

2.3.2.1. Robot use case diagram

A use case is a cohesive unit representing functionality visible from the outside. It provides an end-to-end service, with an initiation, an unfolding, and an end, for the actor who initiates it. Therefore, a use case models a service provided by the system, without imposing the embodiment of this service.

The robot has four main operations: search, move, avoidance and detection of obstacles:

- Research is a process of identifying goals.
- The movement is activities changes of positions on the instructions received from the controller.
- Obstacle avoidance is also a process of approaching goals.

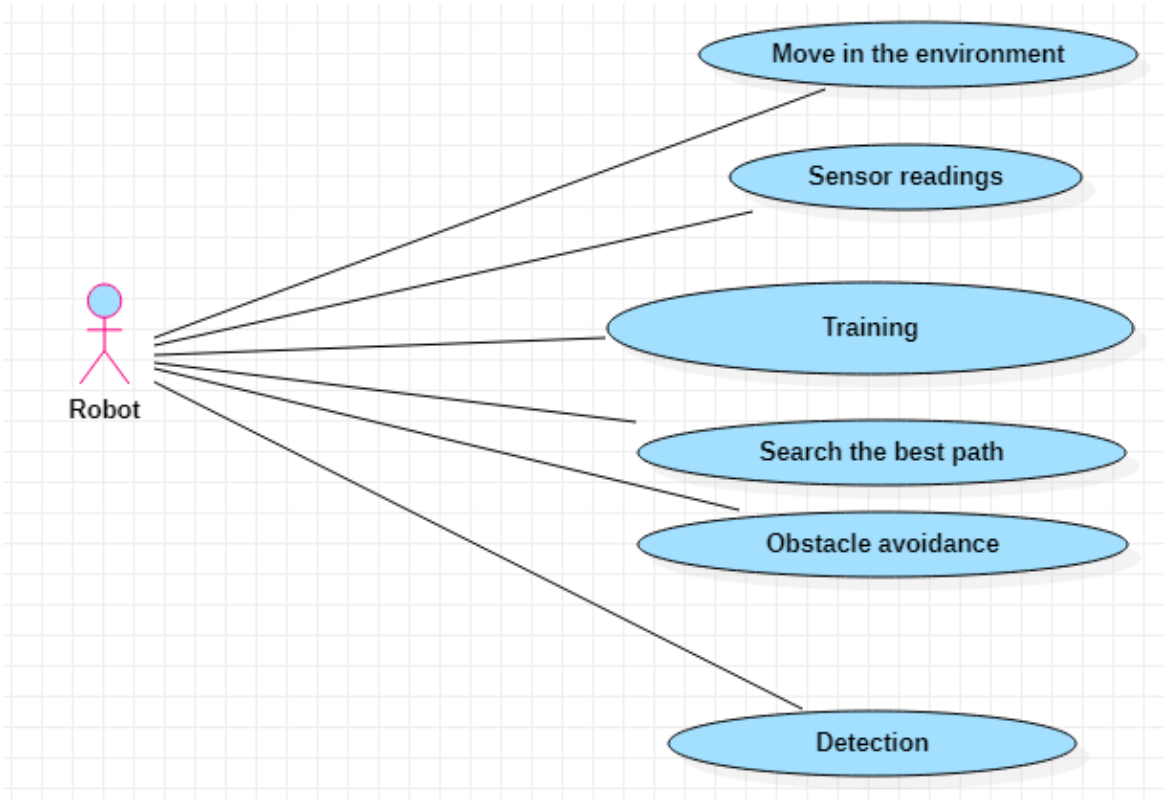


Figure 2.2: use case diagram

In our diagram, we have a single entity that symbolizes our robot as the main actor. the later has 6central operations to perform that being:

- ❖ The track of the path: our robot is situated at the starting position in an environment with an ending point to reach, it is then up to it to decide the best speed to apply .
- ❖ Move in the environment; with the ending point In record the robot begins to make its movement along the environment
- ❖ Obstacles avoidance: obstacles are randomly placed in the environment according to the start and ending point entered, therefore it is essential for the robot to avoid them when they cross its path.
- ❖ Training with neural networks artificial (RNN +MLP)
- ❖ Sensor readings.

2.3.2.2. Sensor use case diagram

Sensors are used to estimate a robot's condition and environment and measures the distance to the nearest wall/obstacle in each direction

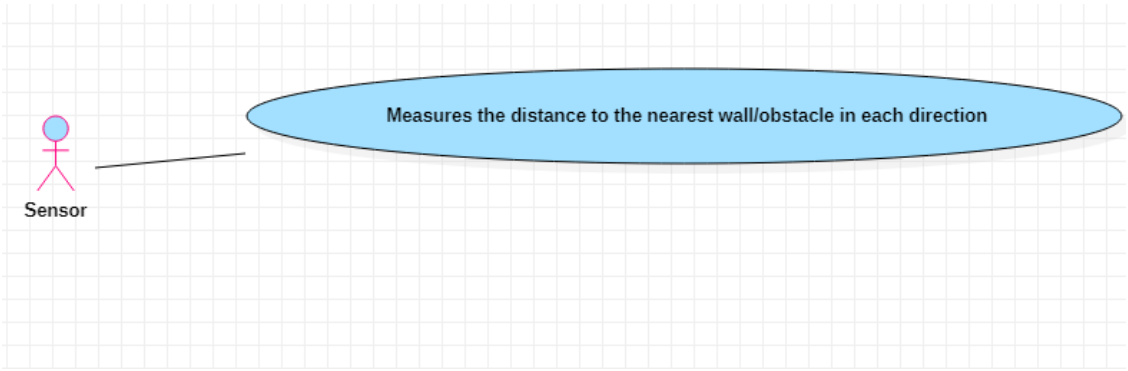


Figure2.3: Sensor use case diagram

2.3.2.3. Motors use case diagram

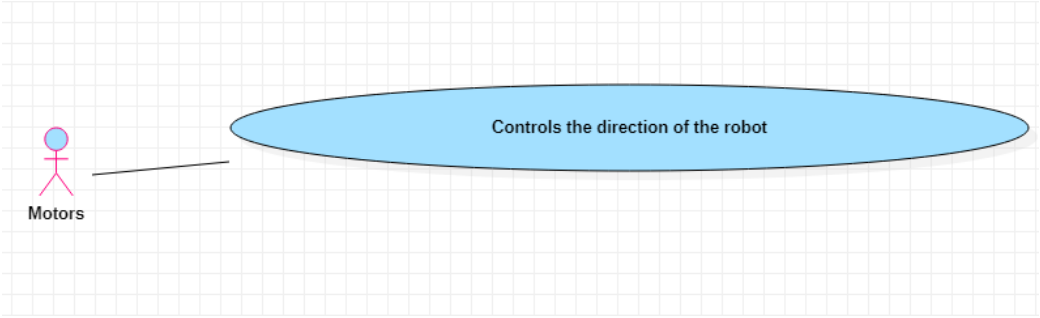


Figure 2.4: Motors use case diagram

In this use case diagram, the Motors control the direction of the robot (right, left)

2.3.3. The Class Diagram:

A class diagram is a type of diagram and part of a unified modeling language (UML) that defines and provides the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes.

Content: classes, subclasses, attributes and values, methods, links (multiplicity, generalization, composition), categories, and dependency. So that the robot does not halt between two trajectories, the robot class must control the sequence of trajectory generation and

commands. It has a specific behavior compared to other classes. UML expresses this behavior by the use of state charts which allow representing the parallelism well.

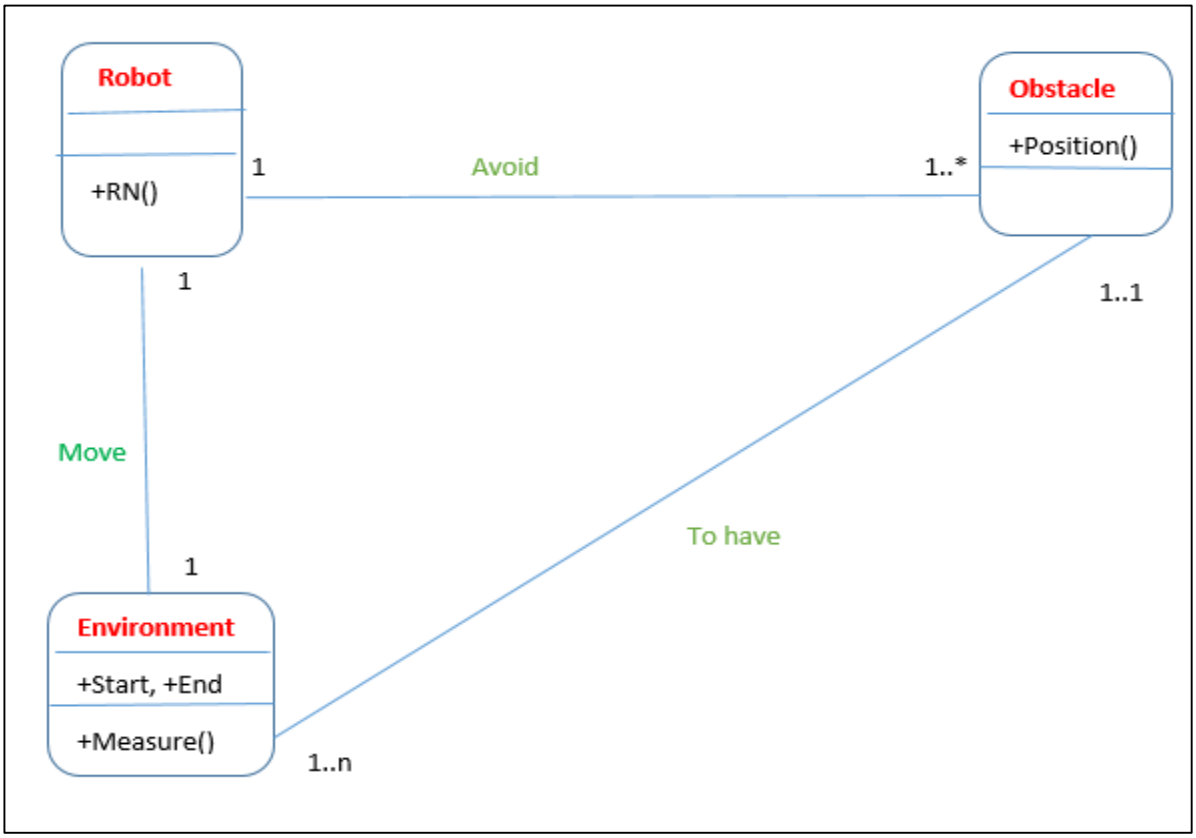


Figure 2.5:Class Diagram

- Environment Class It manages the entry of the starting and ending points, and the measurements of the path chosen.
- Obstacles Class It manages all operations of the obstacles including positions. user class it manages all operation of the user
- Robot class it manages all the neural networks artificial.

2.3.4. Sequence diagram:

A sequence diagram, in the context of UML, represents object collaboration and is used to define event sequences between objects for a certain outcome. A sequence diagram is an essential component used in processes related to analysis, design and documentation. Communication between these objects is modeled by the sequence diagrams in Figure 2.6.

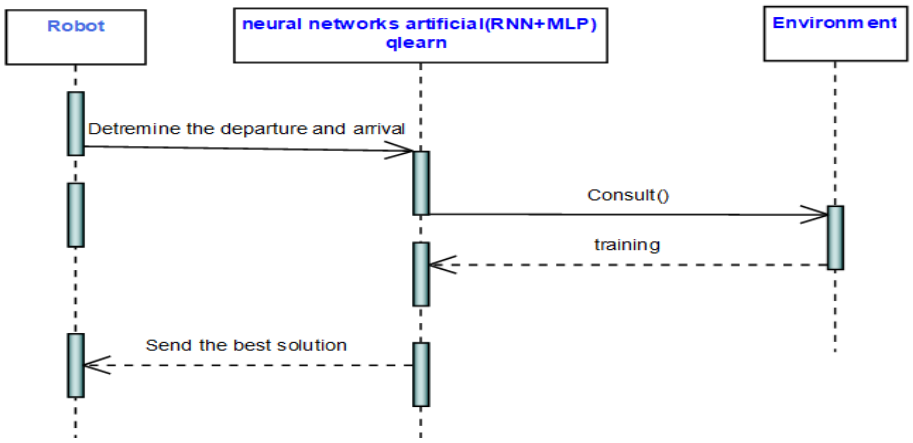


Figure 2.6: Sequence diagram illustrating the robot action

2.3.5. The Activity Diagram:

In UML, an activity diagram provides a view of the behavior of a system by describing the sequence of actions of a process. Activity diagrams are similar to information processing flowcharts because they show the flows between actions in an activity. Activity diagrams can, however, also show simultaneous parallel flows and replacement flows.

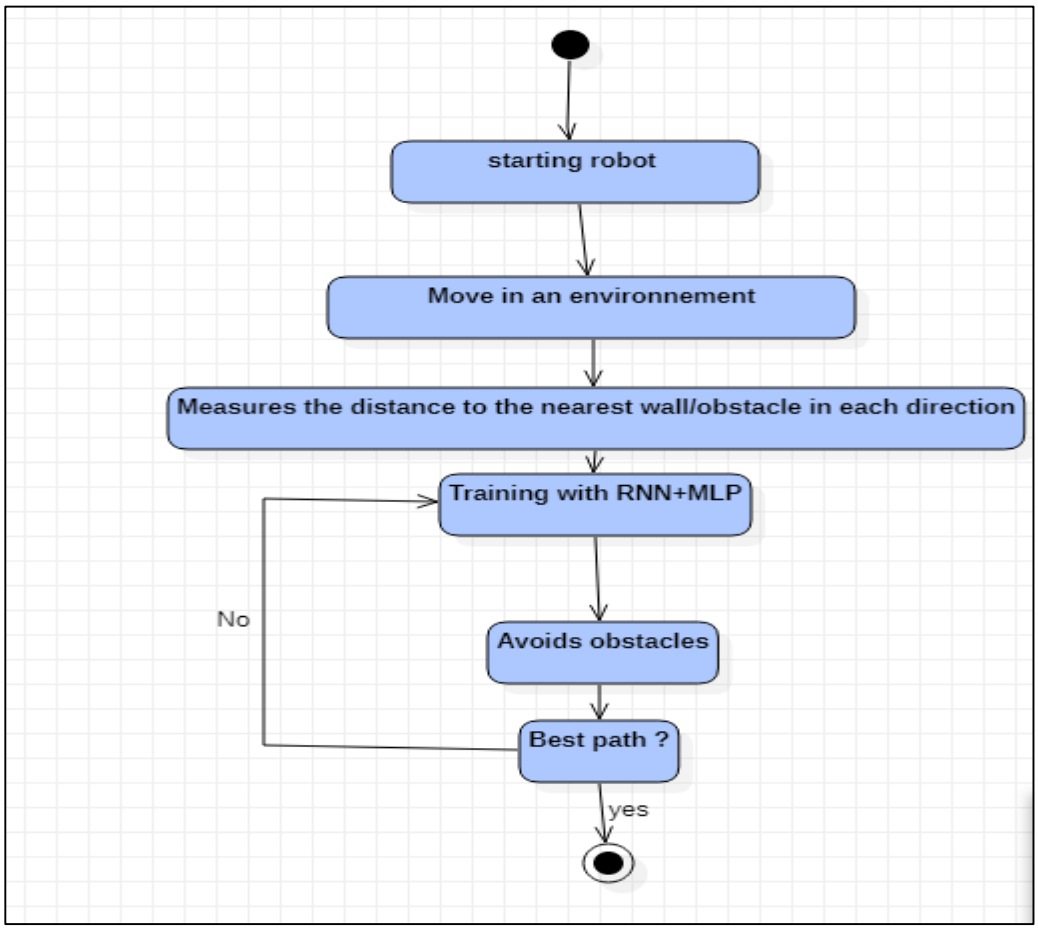


Figure 2.7: Activity Diagram.

Begin from the starting robot and then it move in an environment to measures the distance to the nearest wall/obstacle in each direction ,at this moment it does his training and avoids obstacles and finally it choose the best path .

2.4 Artificial neural networks (ANNs) Architecture

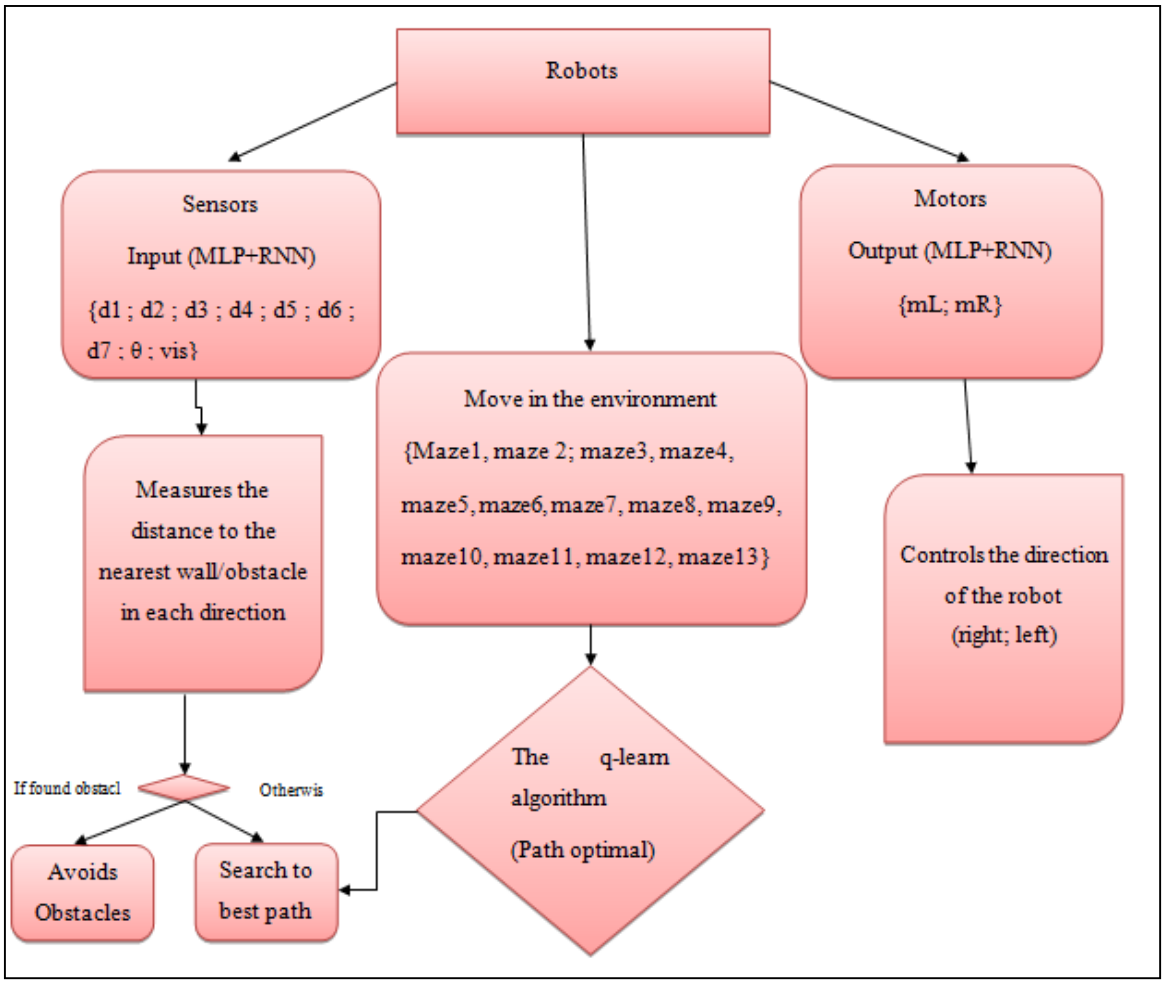


Figure 2.8: General Flowchart of neural networks artificial

2.5. Multilayer Perceptron

- A fully-connected MLP
  - Trained with back-propagation.
- Input and output
  - Input: Sensory Measurements {d1 ;d2 ;d3 ;d4 ;d5 ;d6 ;d7 ;θ ; vis }.
  - Hidden layer 1, Hidden layer 2.
  - Output: Motor Actions {mL; mR}.

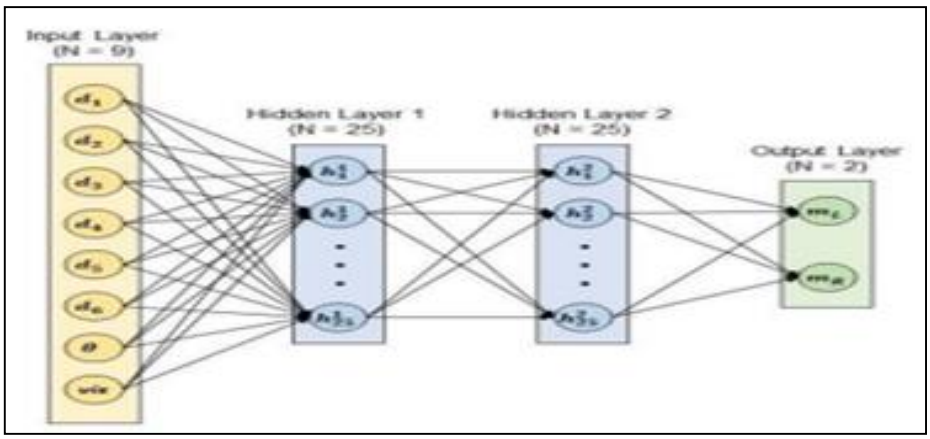


Figure2.9: Architecture of the multilayer perceptron neural network

Parameter	Value
Learning Rate, η	0.05
Number of Samples per Epoch	650
Number of Epochs	100000

Figure2.10: MLP training parameters

2.6. Elman-type Recurrent Neural Network

A fully-connected Elman-type RNN

- Trained with Back-propagation through Time

- Input and output
  - Input: Sensory Measurements{d1 ;d2 ;d3 ;d4 ;d5 ;d6 ;d7 ;θ ; vis }.
  - Output: Motor Actions {mL ;mR }.

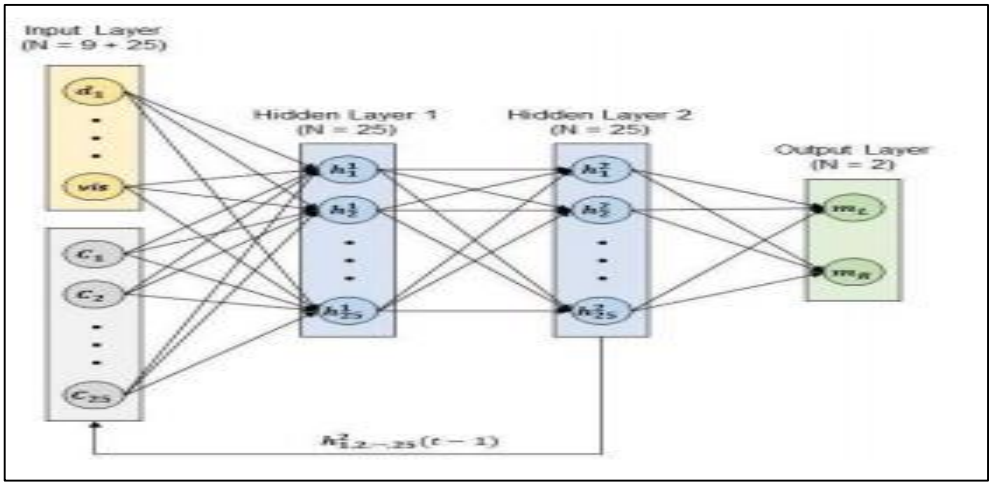


Figure2.11: Architecture of the Recurrent Neural Network.

Parameter	Value
Learning Rate, η	0.0005
Length of a Path Segment	50
Number of Epochs	100000

Figure 2.12: RNN training parameters

2.7. Result of the maze with model RNN and MLP

Test 1

Test with the Default Goal Positions in the 13 Mazes

Test with original goal points in all 13 mazes

○ Tested with 10 trials per maze

● Number of successful trials on each maze

Maze	1	2	3	4	5	6	7	8	9	10	11	12	13	Total
RNN	8	6	0	6	10	0	0	0	0	5	0	1	0	36
MLP	0	0	0	0	1	8	10	0	0	10	4	0	0	33

Table 2.1: Comparison of the mazes between RNN and MLP.

- RNN > MLPin 5 mazes and MLP >RNN in 4 mazes.
- Each model has certain mazes with good performance than the other model.

Test 2

Test with two new Goal Positions in the 13 Mazes.

Test 1) Test with original goal points in all 13 mazes.

○ Tested with 10 trials per maze.

● Number of successful trials on each maze.

Maze		1	2	3	4	5	6	7	8	9	10	11	12	13	TOTAL
Goal 1	RNN	2	5	4	4	10	3	0	10	10	10	4	5	10	77
	MLP	0	0	0	0	10	0	2	0	0	1	0	10	10	33
Goal 2	RNN	3	9	0	0	5	5	10	3	0	0	0	10	10	55
	MLP	3	0	0	0	1	0	10	0	0	0	0	0	0	14

Table 2.2: Comparison of the mazes between RNN and MLP

- RNN > MLP in 15 different goals and MLP > RNN in 2 different goals.

Conclusion:

In this chapter, we propose a new method of robot trajectory mapping; it is based on artificial neural networks and the q-learning algorithm.

The method is used to model the environment of the robot by MLP and RNN neural networks to avoid obstacles; The Q-Learning algorithm is applied to find the optimal path in an environment.

We have presented the different diagrams: diagram of cases of initiation, sequence diagrams, activity diagram and class diagram. These diagrams made our work easier to understand the problem; to obtain a solution of the problem of movement of the mobile robot in spaces with obstacles has been proposed.

In the next chapter we will present the tools and languages used in our application.

Chapter 3: Realization and Implementation

3.1. Introduction:

In order to show the autonomous control of the mobile robot, we used an application developed under Matlab which is an intelligent system with several functions used to simulate the behavior of the robot in obstacle avoidance to reach the target and to assess the performance of the techniques used.

Optimization issues are becoming more complex and the rapid development of technology has made the use of a neural approach increasingly necessary.

In addition, the cost / performance ratio in parallel IT systems continues to decrease. The neural approach is used in the design and implementation of meta-heuristics to accelerate research, and to improve the quality of the solutions obtained, to improve robustness and solve problems on a large scale.

In this thesis, our research is done on the problem of finding the shortest path of an autonomous mobile robot. We have proposed a new neural approach for solving the problem of finding the shortest path for an autonomous mobile robot, based on the artificial neural networks.

3.2. Tools and working environments:

3.2.1. Software environment:

Matlab is digital calculation software marketed by the company MathWorks. He was initially developed in the late 1970s by Cleve Moler, professor of mathematics at the University of New Mexico and then Stanford, to allow students to work from of a high-level programming tool and without learning FORTRAN or C. Matlab stands for Matrix laboratory. It is a language for scientific computing, data analysis, their visualization, the development of algorithms. Its interface offers, on the one hand, a windowinteractive console type for executing commands, and on the other hand, aintegrated development (IDE) for application programming.

Matlab finds its applications in many disciplines. It is a digital toolpowerful for modeling physical systems, simulating mathematical models, design and validation (simulation and experimentation tests) of applications. The basic software can be supplemented with multiple toolboxes, i.e. toolboxes. These are function libraries dedicated to particular domains. We can cite for example: Automation, signal processing, statistical analysis, optimization..etc.

Chapter 3: Realization and Implementation

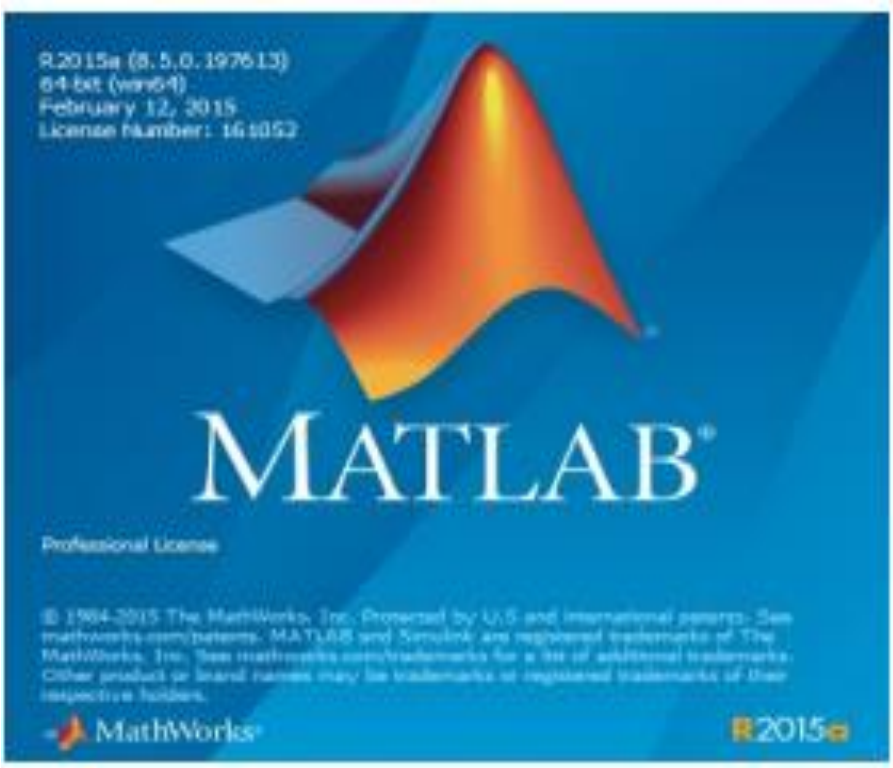
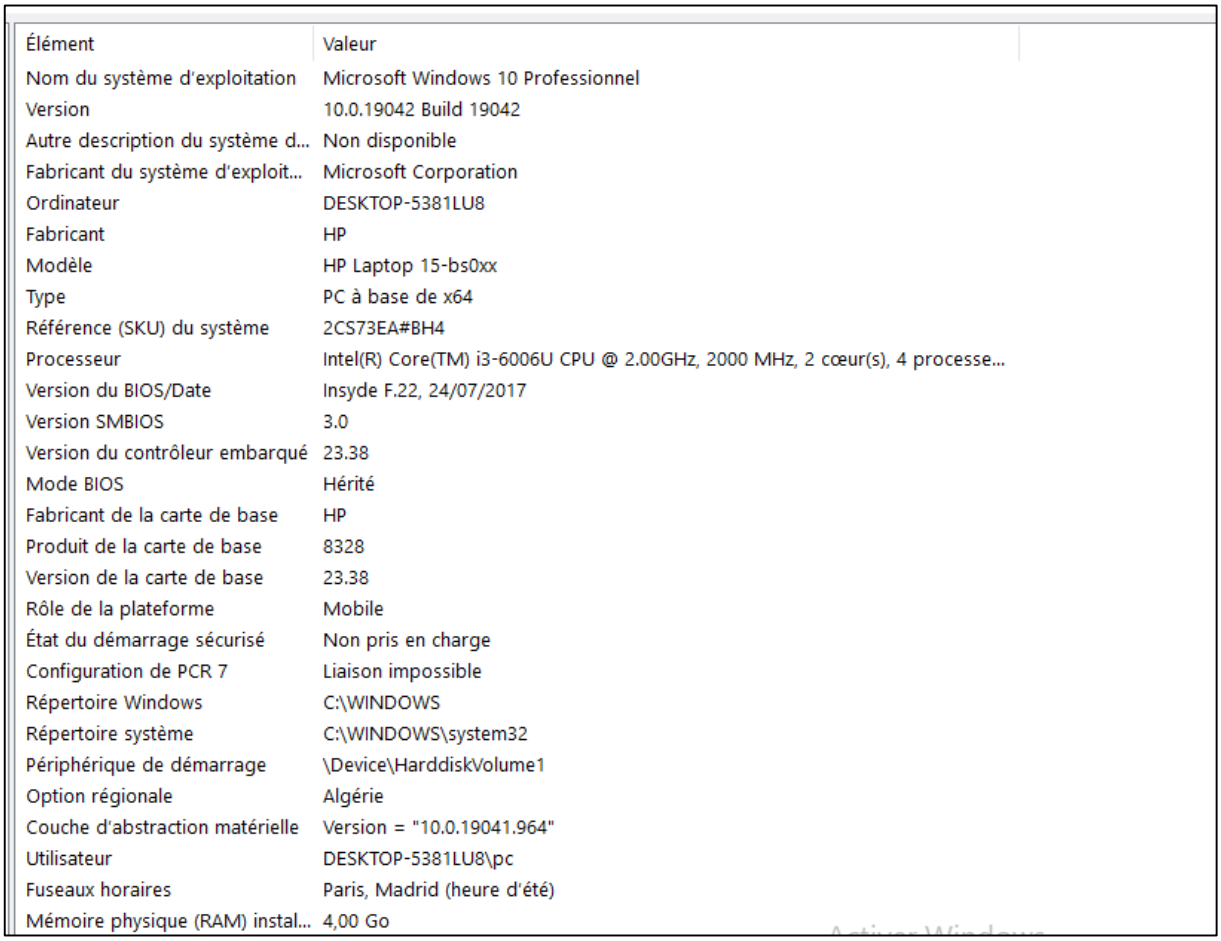


Figure 3.1: Matlab logo [38]

StarUML is a UML modeling program that was "transferred as open source" by its publisher at the conclusion of its commercial exploitation (which obviously continues...), under a modified GNU GPL license. StarUML V3 maintains the majority of the diagrams specified in the UML 2.0 standard, and is currently only accessible under a proprietary license. StarUML is written in Delphi and is based on the Delphi language. StarUML is a Delphi application that relies on proprietary Delphi components (not open-source).

Figure 3.2: Star UML logo [39].

3.2.2. Materials Used:



Élément	Valeur
Nom du système d'exploitation	Microsoft Windows 10 Professionnel
Version	10.0.19042 Build 19042
Autre description du système d...	Non disponible
Fabricant du système d'exploit...	Microsoft Corporation
Ordinateur	DESKTOP-5381LU8
Fabricant	HP
Modèle	HP Laptop 15-bs0xx
Type	PC à base de x64
Référence (SKU) du système	2CS73EA#BH4
Processeur	Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz, 2000 MHz, 2 cœur(s), 4 processe...
Version du BIOS/Date	Insyde F.22, 24/07/2017
Version SMBIOS	3.0
Version du contrôleur embarqué	23.38
Mode BIOS	Hérité
Fabricant de la carte de base	HP
Produit de la carte de base	8328
Version de la carte de base	23.38
Rôle de la plateforme	Mobile
État du démarrage sécurisé	Non pris en charge
Configuration de PCR 7	Liaison impossible
Répertoire Windows	C:\WINDOWS
Répertoire système	C:\WINDOWS\system32
Périphérique de démarrage	\Device\HarddiskVolume1
Option régionale	Algérie
Couche d'abstraction matérielle	Version = "10.0.19041.964"
Utilisateur	DESKTOP-5381LU8\pc
Fuseaux horaires	Paris, Madrid (heure d'été)
Mémoire physique (RAM) instal...	4,00 Go

Figure 3.3: Materials Used

3.3. Description of the different modules:

3.3.1. The environment:

A- Direction of the robot

Nine (09) Input Sensors { d1 ; d2 ;d3 ;d4 ;d5 ;d6 ;d7 ; θ ; vis}, seven (07) lasers for wall/obstacle detection.

- Directions = (-90º, -60º, -30º, 0º, 30º, 60º, 90º).
- Each laser measures the distance to the nearest wall/obstacle in each direction.

<u>One laser for goal tracking:</u>

- Measures the angular difference between the robot's heading direction and the goal.
- Checks if the goal is visible (1) or not (0).

2 Output Motors {mL ,mG}

- {1, 0} will make the robot turn right.

Chapter 3: Realization and Implementation

Robot Navigation:

● When the goal is not visible, the robot wanders around the maze while avoiding obstacles based on the distance laser readings.

● When the goal is visible, the robot moves toward the goal. The robot avoids obstacles only when a collision is imminent.

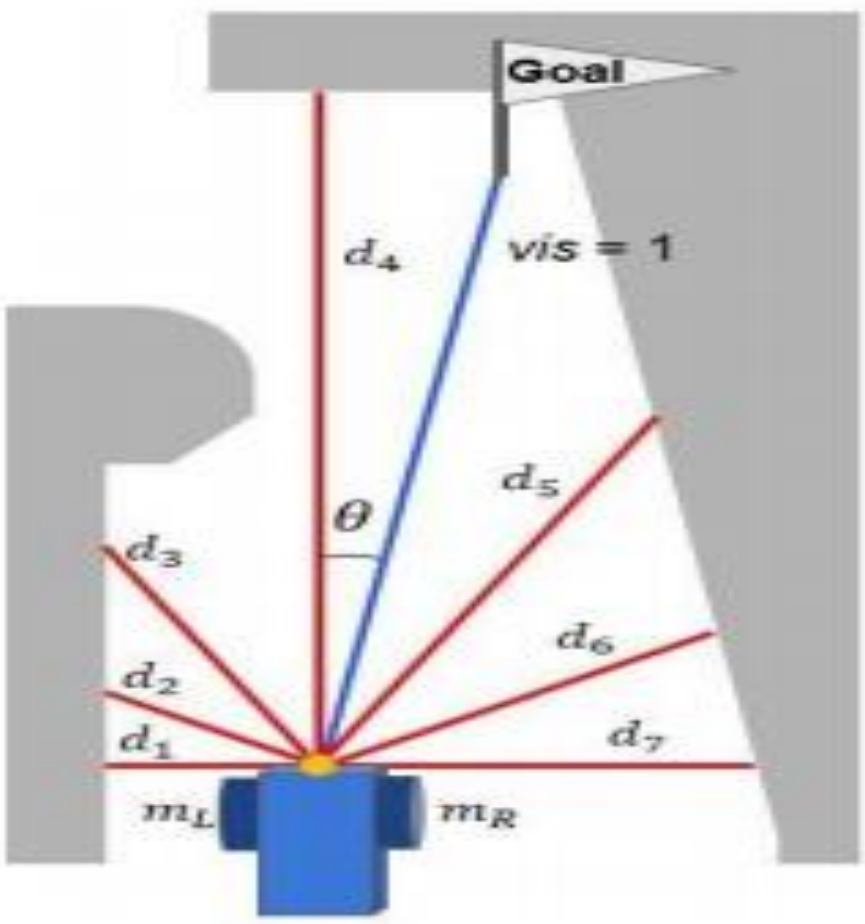


Figure 3.4**:** The directions of our robot.

Nine (09) Input Sensors { d1 ; d2 ;d3 ;d4 ;d5 ;d6 ;d7 ; θ ; vis}, seven (07) lasers for wall/obstacle detection.

■ Directions = (-90º, -60º, -30º, 0º, 30º, 60º, 90º).

■ Each laser measures the distance to the nearest wall/obstacle in each direction.

One laser for goal tracking:

■ Measures the angular difference between the robot's heading direction and the goal.

■ Checks if the goal is visible (1) or not (0).

2 Output Motors {mL ,mG}

- {0, 1} will make the robot turn left.

Robot Navigation:

● When the goal is not visible, the robot wanders around the maze while avoiding obstacles based on the distance laser readings.

● When the goal is visible, the robot moves toward the goal. The robot avoids obstacles only when a collision is imminent.

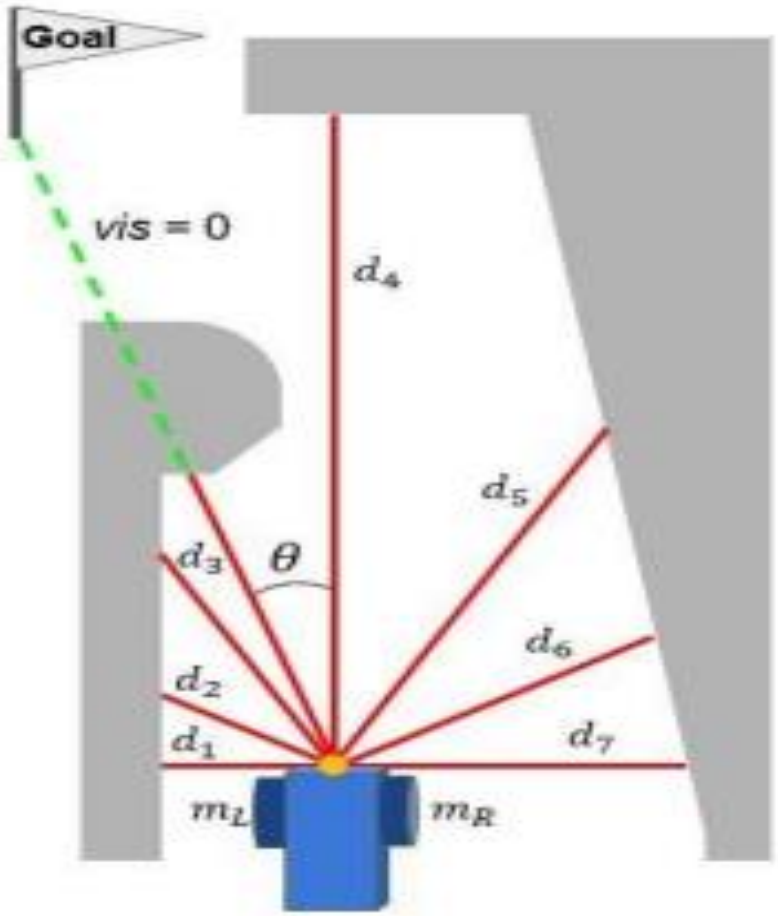


Figure 3.5: The Robot with new goal position

B-Navigation of the robot

The fixed starting point for each maze

The fixed goal point for each maze

A maze (or an environment) is consisted of walls and obstacles

Chapter 3: Realization and Implementation

Figure3.6: Representation of the mazes.

Chapter 3: Realization and Implementation

The robot has to reach an end point from a starting point, the robot has a starting position (Xs, Ys) and an end position (Xa, Ya), then it avoids the obstacles and chooses the best path

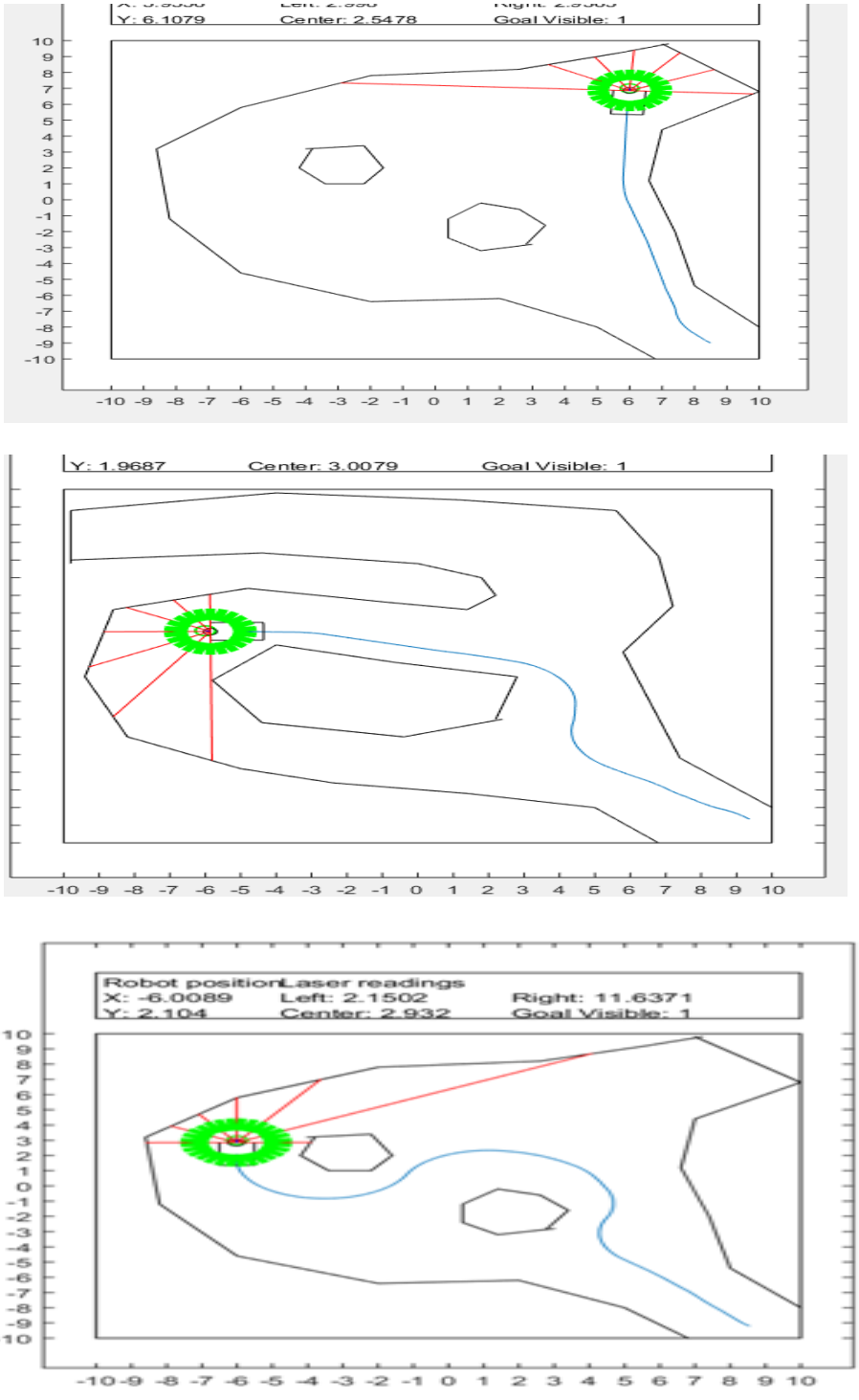


Figure3.7: Navigation of the robot for different mazes

C-Test with the Default Goal Positions in the 13 Mazes

- A sample result.
  - Tested maze: Maze.

- Model: RNN
  - Result: Goal reached

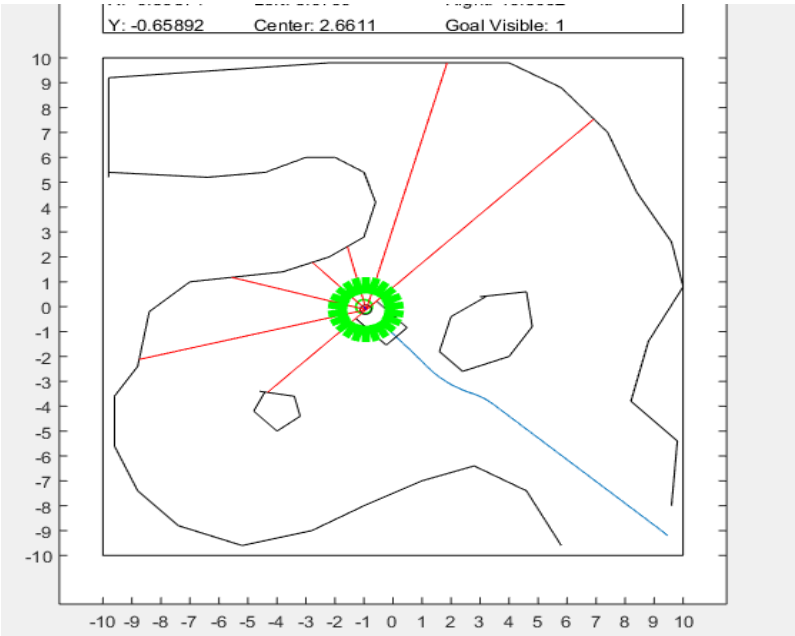


Figure 3.8: Tested maze 11

- A sample result
  - Tested maze: Maze5
  - Model: MLP
    - Result: Goal reached

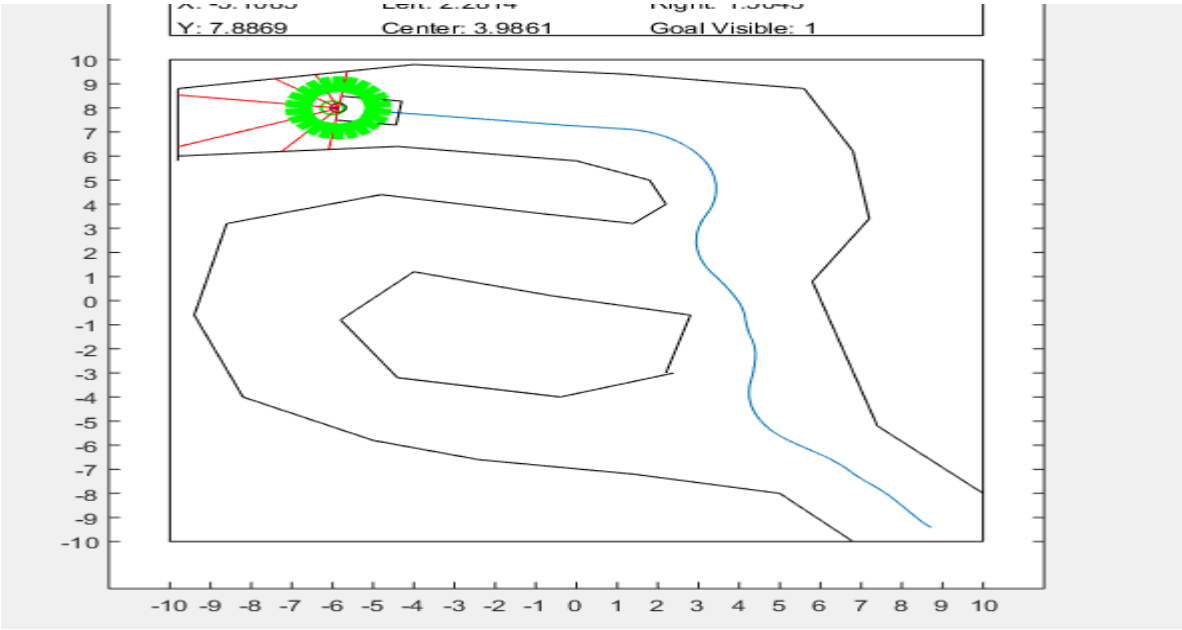


Figure 3.9: Tested maze 5

3.3.2. Matlab Code:

A-Multi-layer Perceptron

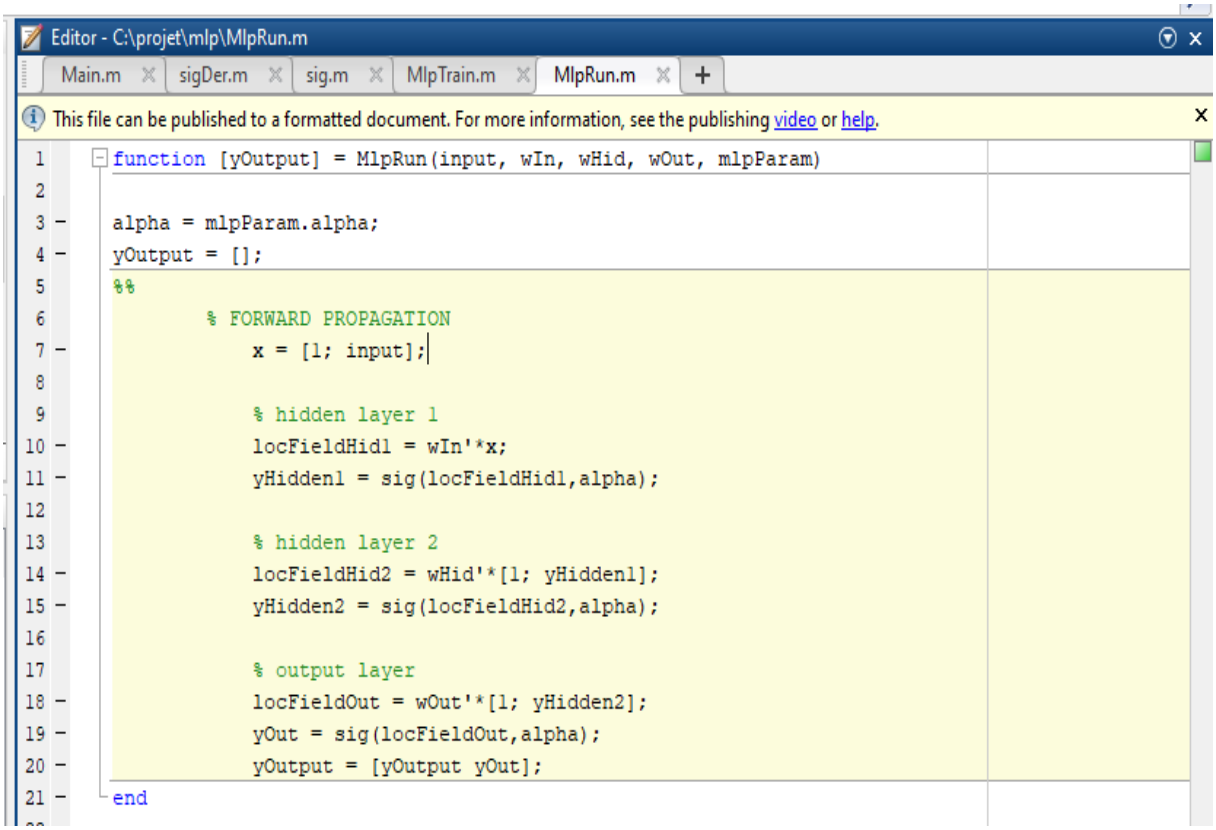


Figure 3.9:Matlab code Multi-layer perceptron source code

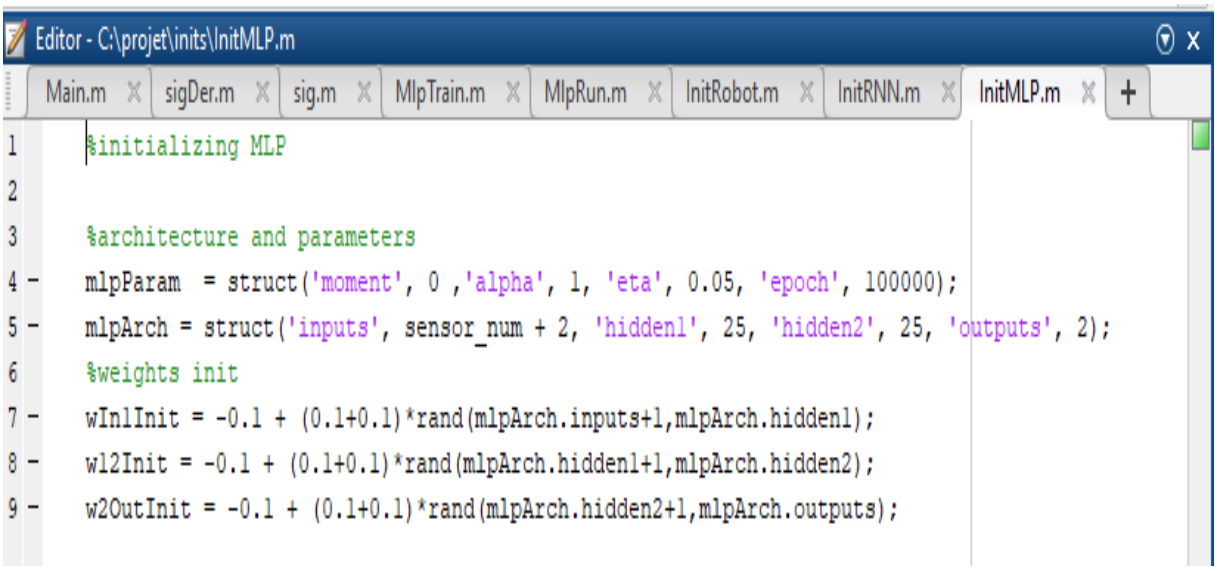


Figure 3.10:Matlab code Parameters multi-layer percptron

B- Recurrent Neural Network

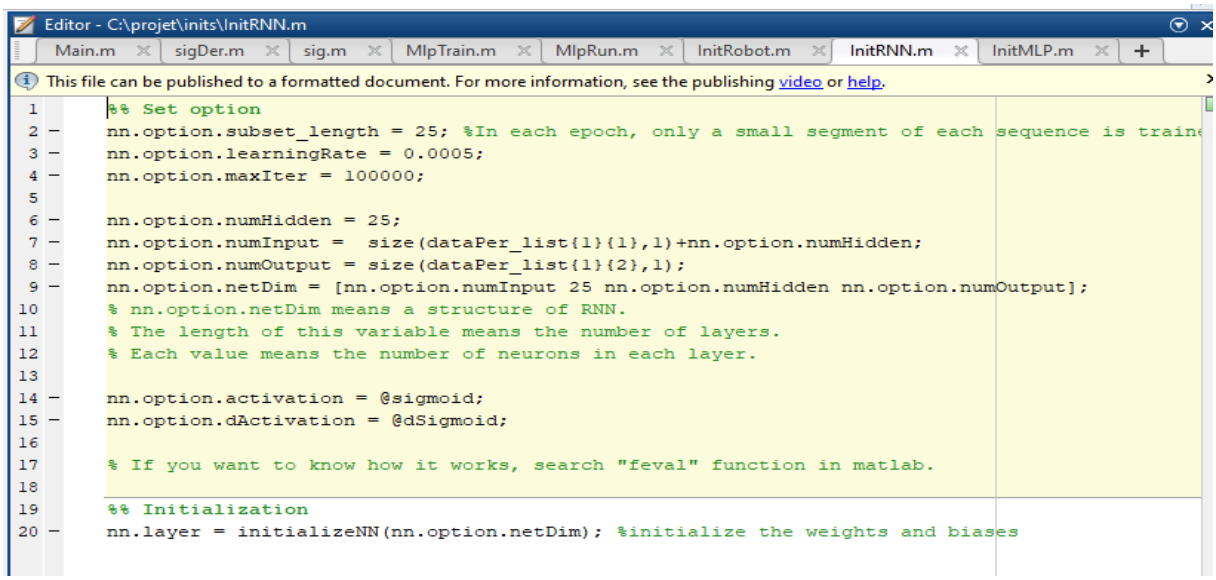


Figure 3.11**:** Matlab code Recurrent Neural Network

3.4. Discussion

The navigation domain has been enriched by the diversity of effort and viewpoints. As a result, a set of criteria for ensuring the quality of such a solution can be identified. We also did a tiny comparison between two other methods: ant colony [40] and bee colony [41] and genetic algorithm [42], to demonstrate the performance of artificial neural networks in solving the shortest path problem, and we noticed that the rate of convergence in neural networks artificial is more interesting than other methods.

Training with multiple mazes:

- The performance of each network varied up to the mazes
- RNN could navigate better than MLP toward the goals not used in training
- The robot can be taking the good short way when using RNN then MLP
- The robot can be taking the best short way when using RNN and MLP

RNN showed overall better performance than MLP did:

- Since path data is time-series data, the navigation problem can better be solved by using the dynamics of RNN.

Neural network-based navigation is a good option for mobile robot:

- Can deal with the flexible situation and the environment change.
- Can deal with start and target position changes.

Chapter 3: Realization and Implementation

Conclusion:

In this chapter, we have tried to present in a simple way the different stages through which we went. A new neural approach has been shown to find the shortest path of an autonomous mobile robot. This approach is able to guide the robot to move autonomously. The proposed approach improves the performance of navigation on several levels. In general, solutions based on this approach allow the resolution of our problem.

General Conclusion

The study of neurons and their networks has allowed us to understand (in part) their functioning. Indeed, a neuron has a morphology adapted to its function, to circulate information in our brain with the aim of a good coordination of our organism. Moreover, we now know the way in which they operate and the fact that only one of them is connected to a hundred thousand of his fellows, which constitutes a network. These studies have enabled the development of two types of so-called “artificial” neurons: laser micro-pillars and electronic chips. These “artificial” neurons want to be as close as possible to the real functioning of biological ones. In addition, scientists come to imagine entirely artificial networks. This is why the development of these artificial neural networks could well lead one day to the creation of various artificial intelligences (intelligent computer, humanoid robot, etc.). We can then wonder about the danger of artificial intelligences on the future of the human race. It would no longer be science fiction, but just science. As part of this work, we have carried out a new trajectory planning technique based on artificial intelligence networks neurons.

In recent years, roboticists have become increasingly interested in the subject of autonomous navigation of mobile robots in natural environments. They hope to steadily raise the degree of autonomy of their robots until they approach complete autonomy and are capable of long missions.

Fully autonomous mobile robot mobility in dynamic situations is still a tough subject to tackle. It necessitates the development of functions that allow the perception, decision, and action cycle to be completed. To be able to deal with a wide range of scenarios that the robot may encounter while navigating.

In a general way, by this study we highlighted the interest of the intelligent system not only for the autonomous control of mobile robot, we used the MATLAB. The implementation of artificial neural networks shows that the goal has been achieved by avoiding obstacles.

Several perspectives are possible following our work. So, it would be interesting to introduce "dynamics" into our study of autonomous navigation, taking into account mobile obstacles, which will raise particularly delicate problems.

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] Bouhalassa, L.,«Robot trajectory planning based on neural networks and genetic algorithms», end-of-studies project dissertation for the diploma of magister electronics Faculty of electrical engineering Department of electronics University of sciences of technology of oran 2010

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

[3] Latombe, J, C., Robot Motion Planning, *Kluwer Academic Publishers*, Norwell.

[4] 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 25-28,1991

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

[6] Brooks, R, A.,A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1,pp. 14-23,1986.

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

[8] Adachi, Y., Saito, H., Matsumoto, Y., Ogasawara, T., 2003. Memory-based navigation using data sequence of laser range finder, Computational Intelligence in Robotics and Automation, Proceedings IEEE International symposium, Vol 1, pp. 479 – 484.

[9] Lettvin, J, Y., Maturana, H, R., McCulloch, W, H., 1959. What the Frog's Eye Tells the Frog's Brain, *Proceedings of The Institute of Radio Engineers*, New York, Vol. 47, No. 11, pp. 1940-51.

[10] Lebedev, D, V., Steil, J, J., Ritter, H, J., 2005. The dynamic wave expansion neural network model for robot motion planning in time-varying environments, *Neural Networks*, Volume 18, Issue 3, pp. 267-285.

[11] Belker, T., Schulz, D., 2002. Local Action Planning for Mobile Robot Collision Avoidance, intelligent Robots and System, *IEEE/RSJ International Conference on Volume 1*, 30, Page(s):601 - 606 vol.1.

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

[13] 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.

References

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

[15] 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.

[16] 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).

[17] Lucic, c., Teodorovic, D., 2001.Optimisation des colonies d'abeilles (BCO) Université de Belgrade, Faculingénieur des transports et du traficg, Vojvode Stepe 305 11000 Belgrade, Serbeia dusan@sf.bg.ac.rs

[18] 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.

[19] 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*

[20] 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).

[21] 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

[22] Dupond, S.,"A thorough review on the current advance of neural network structures". Annual Reviews in Control. **14**: 200–230. (2019).

[23]Abiodun, O-I., Jantan, O., A-E.; Dada, K-V., Mohamed, N –A., Arshad, H.,"State-of-the-art in artificial neural network applications: A survey". Heliyon. **4** (11): e00938.doi:10.1016/j.heliyon.2018.e00938.ISSN24058440. PMC 6260436. PMID 305153. (2018-11-01).

References

[24]Tealab, A.,"Time series forecasting using artificial neural networks methodologies: A systematic review". Future Computing and Informatics Journal. **3** (2): 334–340. doi:10.1016/j.fcij.2018.10.003. ISSN 2314-7288(2018-12-01).

[25]Miljanovic, M.,"Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction" (PDF). Indian Journal of Computer and Engineering. (Feb–Mar 2012).

[26]Marc, P.,“ The multilayer perceptron and its backpropagation algorithm“ department of electrical engineering and computer engineering lava university September 10, 2004

[27] Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F. (2020). "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning". IEEE Transactions on Vehicular Technology. 69 (12): 14413–14423.

[28] Kaelbling, Leslie P.; Littman, Michael L.; Moore, Andrew W. (1996). "Reinforcement Learning: A Survey". Journal of Artificial Intelligence Research. **4**: 237–285. arXiv:cs/9605103. doi:10.1613/jair.301. S2CID *1708582. Archived from the originalon 2001-11-20.*

[30] *Melo, Francisco S. "Convergence of Q-learning: a simple proof" (PDF).*

[31] Jump up to:[a] [b] *Matiisen, Tambet (December 19, 2015). "Demystifying Deep Reinforcement Learning". neuro.cs.ut.ee. Computational Neuroscience Lab. Retrieved 2018-04-06.*

[32] Filipe, A., Bernardete, R.,Luıis, R.,« A Neural Network for Shortest Path Computation » Faculty of Science Department of informatique Lisboa Campo Grande, 1700 Lisbon Portugal April 6, 2000

[33] Mozer, M. C. (1995). "A Focused Backpropagation Algorithm for Temporal Pattern Recognition". In Chauvin, Y.; Rumelhart, D. (eds.). Backpropagation: Theory, architectures, and applications. ResearchGate. Hillsdale, NJ: Lawrence Erlbaum Associates. pp. 137–169. Retrieved 2017-08-21.

[34] Robinson, A. J. & Fallside, F. (1987). The utility driven dynamic error propagation network (Technical report). Cambridge University, Engineering Department. CUED/F-INFENG/TR.1.

References

[35] Werbos, P.,"Generalization of backpropagation with application to a recurrent gas market model". Neural Networks. **1** (4): 339–356. doi:10.1016/0893-6080(88)90007-x.] (1988).

[36] Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P-Y., Hjalmarsson, H., Juditsky, A., "Nonlinear black-box modeling in system identification: a unified overview". *Automatica*. **31** (12): 1691–1724. CiteSeerX 10.1.1.27.81. doi:10.1016/0005-1098(95)00120-8(1995).

[37] Belkhouche, I.,« Trajectory planning and obstacle avoidance by neural networks for robots autonomous mobile » end of studies thesis for obtaining the Master's degree in Computer Science University Abou Bakr Belkaid– Tlemcen 26 September 2011

[38] Cadieux, J., Ariba, Y. , A book « Manuel Matlab » Départements GEL et Mécanique version,0.1 page 6

[39] MKLab. The software was licensed under a modified version of GNU GPL until 2014, when a rewritten version 2.0.0 was released for beta testing under a proprietary license.

[40] Hdimi, k., « Approches bio-inspirées pour la coordination de robots dans un milieu hostile: colonie des fourmis », Master Académique SII, université chadli Bendejdid el Tarf.2019

[41] Belkaid, N., « Approches bio-inspirées pour la coordination de robots: colonie d'abeilles », Master Académique SII, université chadli Bendejdid el Tarf. September 2019

[42] Melouk, A., « An evolutionary approach to searching the shortest of a self-contained mobile robot » , Master Académique SII, université chadli Bendejdid el Tarf. October 2020

[43] http://eavr.u-strasbg.fr/-bernard

[44] huahua, l., and hugh, l., and cheng, c., and yawei, z., and xingqun, z., "Navigation information augmented artificial potential field algorithm for collision avoidance in UAV formation flight" ,Aerospace Systems ,"3(3), 229-241. Doi : 10.1007/s42401-020-00059-6 27 august 2020.

[45] Songpo, l., and Fernando, j., and Rodrigo, d., and Diedra, G., "Attention –Aware Robotic Laparoscope Based On Fuzzy Interpretation of eye-Gaze Patterns" journal of medical devices may 2015 .

[46] Ping, D.,and Yong-Heng? A., « Research on an Improved Ant Colony Optimization Algorithm and its Application » international journal of hybrid information technology 2016.

References

[47] Ameur, d., and David, l., and Kamel, s., « Genetic-based decoder for sratistical machine translation » book december 2016.

[48] Golam, k., « Comparative Analysis of Artificial Neural Networks and Neuro-Fuzzy Models for Multicriteria Demand Forecasting »International Journal of Fuzzy System Applications 3(1):1-24 January 2013.

[49] kao-shing, h., and chia-yue, l., « Cooperation Between Multiple Agents Based on Partially Sharing Policy » Conference: Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, Third International Conference on Intelligent Computing, ICIC 2007, Qingdao, China, August 21-24, 2007.