

# The proposed hybrid intelligent system for path planning of Intelligent Autonomous Systems

O. Hachour

**Abstract**— In this paper, we discuss the ability to deal with a Hybrid Intelligent Systems (HIS) for Intelligent Autonomous Vehicles IAV in unknown environment. The aim of this work is to develop HIS combining Genetic Algorithms (GA), Fuzzy Logic (FL), Neural Networks (NN) and Expert Systems (ES). This project deals with a simulation program that allows a robot to identify a path to reach a specified target avoiding obstacles. The combination of (ES FL, NN, GA) offers design flexibility and robust integration and has the benefits of reduced communications overhead and improved runtime performance. This integration provides the robot the possibility to move from the initial position to the final position (target) without collisions. The robot moves within the unknown environment by sensing and avoiding the obstacles coming across its way towards the target. The algorithm permits the robot to move from the initial position to the desired position following an estimated trajectory. The proposed hybrid navigation strategy is designed in unknown environment with static unknown obstacles. This approach must make the robot able to achieve these tasks: to avoid obstacles, and to make ones way toward its target by *ES\_FL\_GA\_NN system* capturing the behavior of a human expert. The integration of these technologies (FL, NN, ES, and GA) has proven to be a way to develop useful real-world applications, and hybrid systems involving robust adaptive control. The proposed approach has the advantage of being generic and can be changed at the user demand. The results are satisfactory to see the great number of environments treated. The results are satisfactory and promising. the proposed method is computationally efficient and is suitable for more integration of hybrid intelligent systems.

**Keywords**— Intelligent Autonomous Vehicles (IAV), Fuzzy Logic (FL), Expert System (ES), and Neural Networks.

## I. INTRODUCTION

ROBOTICS systems capable of some degree of self-sufficiency is the overall objective of an IAS and are required in many fields. The focus is on the ability to move and on being self-sufficient to evolve in an unknown environment for example. Thus, IAS designers search to create dynamic systems to navigate and achieve intelligent behaviors like human in real time. Thus, the recent developments in autonomy requirements, intelligent components, multi-robot systems, and massively parallel computer have made the IAS very used, notably in the planetary explorations, mine industry, and highways. The objective of intelligent mobile robots is to improve machine autonomy. This improvement concerns three (03) essential aspects. First, robots must perform efficiently

some tasks like recognition, decision-making, and action which constitute the principal obstacle avoidance problems. They must also reduce the operator load by using natural language and common sense knowledge in order to allow easier decision making. Finally, they must operate at a human level with adaptation and learning capacities [1, 2, 3, 4].

Autonomous robots which work without human operators are required in robotic fields. In order to achieve tasks, autonomous robots have to be intelligent and should decide their own action. When the autonomous robot decides its action, it is necessary to plan optimally depending on their tasks. More, when a robot moves from a point to a target point in its given environment, it is necessary to plan an optimal or feasible path avoiding obstacles in its way and answer to some criterion of autonomy requirements such as : thermal, energy, time, and safety for example.

A robotic system capable of some degree of self-sufficiency is the overall objective of an Autonomous Mobile Robot AMR and is required in many fields. The focus is on the ability to move and on being self-sufficient to evolve in an unknown environment for example. Thus, the recent developments in autonomy requirements, intelligent components, multi-robot systems, and massively parallel computer have made the AMR very used, notably in the planetary explorations, mine industry, and highways [15,16].

The ability to acquire all faculties to treat and transmit knowledge constitutes the key of a certain kind of intelligence. Building this kind of intelligence is, up to now, a human ambition in the design and development of intelligent vehicles. However, the mobile robot is an appropriate tool for investing optional artificial intelligence problems relating to world understanding and taking a suitable action, such as, planning missions, avoiding obstacles, and fusing data from many sources[1,2,3,4,5,6,7,10,11].

A robotic vehicle is an intelligent mobile machine capable of autonomous operations in structured and unstructured environment, it must be capable of sensing (perceiving its environment), thinking (planning and reasoning), and acting (moving and manipulating). But, the current mobile robots do relatively little that is recognizable as intelligent thinking, this is because:

- 1) Perception does not meet the necessary standards.
- 2) Much of the intelligence is tied up in task specific behavior and has more to do with particular device and missions than with the mobile robots in general.

- 3) Much of the challenge of the mobile robots requires intelligence at subconscious level.

Therefore, the autonomous mobile robots must be able to achieve these tasks: to avoid obstacles, and to make one way towards their target. In fact, recognition, learning, decision-making, and action constitute principal problem of the navigation. One of the specific characteristics of mobile robot is the complexity of their environment. Therefore, one of the critical problems for the mobile robots is path planning, which is still an open one to be studying extensively. Accordingly, one of the key issues in the design of an autonomous robot is navigation, for which, the navigation planning is one of the most vital aspect of an autonomous robot.

Navigation is the ability to move and on being self-sufficient. Navigation is the science (or art) of directing the course of a mobile robot as the robot traverses the environment. Inherent in any navigation scheme is the desire to reach a destination without getting lost or crashing into any objects. The goal of the navigation system of mobile robots is to move the robot to a named place in a known, unknown, or partially known environment. The navigation planning is one of the most vital aspects of an autonomous robot. In most practical situations, the mobile robot can not take the most direct path from start to the goal point. So, path finding techniques must be used in these situations, and the simplest kinds of planning mission involve going from the start point to the goal point while minimizing some cost such as time spent, chance of detection, etc. When the robot actually starts to travel along a planned path, it may find that there are obstacles along the path, hence the robot must avoid these obstacles and plans a new path to achieve the task of navigation. Therefore, the space and how it is presented is an important role in the domain of moving an intelligent system. We can clarify this importance by the following reasons:

1. It provides the necessary information to do path panning.
2. It gives information for monitoring the position of the robot during the execution of the planned path.
3. It is essential that the mobile robot have the ability to build and use models of its environment that enable it to understand the environment's structure. This is necessary to understand orders, plan and execute paths.

The theory and practice of intelligence and robotic systems are currently the most strongly studied and promising areas in computer science and engineering which will certainly play a primary role in future. These theories and applications provide a source linking all fields in which intelligent control plays a dominant goal. Cognition, perception, action, and learning are essential components of such systems and their integration into real systems of different level of complexity (from micro-robots to robot societies) will help to clarify the true nature of robotic intelligence [11, 12, 13, 14, 15, 16].

This work deals with the intelligent navigation approach of IAV in an unknown environment combining Genetic Algorithms GA, Fuzzy Logic FL, Expert System ES and Neural Networks NN. The aim of this paper is to develop an IAV navigation approach for stationary obstacle avoidance to

provide them more autonomy and intelligence .These technologies (GA, FL,ES, NN) based on intelligent computing are becoming useful as alternate approach, may be able to replace the classical approaches such as: recognition, learning, decision-making , and action ( the principle obstacle avoidance problems).

To deal with Artificial Intelligence and autonomy requirements, the proposed hybrid navigation approach ES\_GA\_FL\_NN can be realized in efficient manner and has proved to be superior to combinatorial optimization techniques, due to the problem complexity. This combination offers to our robot the ability to understand the structure of its environment to find a way towards its target without collisions. Artificial intelligence, including these technologies (NN,GA,FL,ES), has been actively studied and applied to domains such as automatically control of complex systems like robot. Hence, it is interesting to find some technical approaches based on intelligent computing technologies.

## II. THE NECESSITY OF IAV

The theory and practice of IAV are currently among the most intensively studied and promising areas in computer science and engineering which will certainly play a primary goal role in future. These theories and applications provide a source linking all fields in which intelligent control plays a dominant role. Cognition, perception, action, and learning are essential components of such-systems and their use is tending extensively towards challenging applications (service robots, micro-robots, bio-robots, guard robots, warehousing robots). Many traditional working machines already used e.g., in agriculture or construction mining are going through changes to become remotely operated or even autonomous. Autonomous driving in certain conditions is then a realistic target in the near future. A robotic vehicle is an intelligent mobile machine capable of autonomous operation in structured and unstructured environment. It must be capable of sensing (perceiving its environment), thinking (planning and reasoning), and acting (moving and manipulating). Please submit your manuscript electronically for review as e-mail attachments.

However, the mobile robot is an appropriate tool for investigating optional artificial intelligence problems relating to world understanding and taking a suitable action, such as, planning missions, avoiding obstacle, and finding data from many sources[9, 10]. Today, robotic occupies special place in the area interactive technologies. It combines sophisticated computation with rich sensory input in a physical embodiment that can exhibit tangible and expressive behaviour in the physical world. In this regard, a central question that occupies some research group - at -large: "what is an appropriate first role for intelligent human-robot interaction in the daily human environment?". The time is ripe to address this question. Robotic technologies are now sufficiency mature to enable interactive, competetent robot artefacts to be created. The study of human-robot interaction, while fruitful in recent year,

show great variation both in the duration of interaction and the roles played by human and robot participants. In care where human caregiver provides short-term, nurturing interaction to a robot, research has demonstrated the development of effective social relationships. Anthropomorphic robot design can help prime such interaction experiment by providing immediately comprehensible social cues for the human subjects. Technology has made this feasible by using advanced computer control systems. Also, the automotive industry has put much effort in developing perception and control systems to make the vehicle safer and easier to operate.

To perform all tasks in different environments, the robot must be characterized by more severe limits regarding mass volume, power consumption, autonomous reactions capabilities and design complexity. Particularly, for planetary operations severe constraints arise from available energy and data transmission capacities, e.g., the vehicles are usually designed as autonomous units with: data transfer via radio modems to rely stations (satellite in orbit or fixed surface stations) and power from solar arrays, batteries or radio-isotope thermo electric generators (for larger vehicles). A common application of mobile robot is the object manipulation. Examples include pick and place operation on the factory floor, package sorting and distribution. Some researchers are interesting in the simplest kind of object manipulation i.e. pushing. Pushing is the problem of changing the pose of an object by imparting a point contact force to it. For the simplicity, they constrain their self to the problem of changing the pose (in a horizontal plane). An early approach to robot pushing was implemented with two wheeled, cylindrical robots equipped with tactile sensors which implemented object reorientation and object translation. The strategy was to use two robots to push the object at its diagonally opposite corner. As a result of this off-center pushing a torque is applied to the box, rotating it roughly in place. This problem is addressed to detect and push stationary objects in a planar environment by using an environment-embedded sensor network and a simple mobile robot. The stationary sensors are used to detect push able objects. This way illustrates how the robot box-pushing with environment embedded Sensors.

The environment force prevents the robot from moving and turning towards obstacles by giving the user the distance information between the robot and the obstacle in a form of force. This force is similar to the traditional potential force field for path planning of mobile robot. However, the environment force is different from the potential force in some aspects. First there is no attention to a goal since we assume that the goal position is unknown. Secondly, only obstacles in the "relevant" area (according to the logical position of the interface) are considered, i.e. the obstacles that are for, or in the direction opposite to the movement of the robot are not relevant. In this context, a full range of advanced interfaces for vehicle control has been investigated by the researchers. These works demonstrate that obstacle detection and

collision avoidance is improved with good results.

### III. AUTONOMY REQUIREMENTS

Several autonomy requirements must be satisfied to well perform the tasks of IAV; this is summarized in some in the following section:

#### A. Thermal

To carry out tasks in various environments as in space applications, the thermal design must be taken into account, especially when the temperature can vary significantly. At ambient temperatures, the limited temperature -sensitive electronic equipment on-board must be placed in thermally insulated compartments.

#### B. Energy

For a specified period, IAV can operate autonomously, one very limited resource for underwater and space applications are energy. So, IAV usually carry a rechargeable energy system, appropriately sized batteries on-board.

#### C. Communication Management

The components on-board the vehicle and on-board the surface station must be interconnected by a two-way communication link. As in both underwater and space applications, a data management system is usually necessary to transfer data from IAV to terrestrial storage and processing stations by two-way communication link. Indeed, the data management system must be split between components of the vehicle and surface station. Thus, the vehicle must be more autonomous and intelligent to perform and achieve the tasks. Due to the limited resources and weight constraints, major data processing and storage capacities must be on the surface station. Although individual vehicles may have wildly different external appearances, different mechanisms of locomotion, and different missions or goals, many of the underlying computational issues involved are related to sensing and sensor modelling spatial data representation, and reasoning.

#### D. Best Navigation

Navigation is the ability to move and on being self-sufficient. The IAV must be able to achieve these tasks: to avoid obstacles, and to make one way towards their target. In fact, recognition, learning, decision-making, and action constitute principal problem of the navigation. One of the specific characteristic of mobile robot is the complexity of their environment. Therefore, one of the critical problems for the mobile robots is path planning, which is still an open one to be studying extensively. Accordingly, one of the key issues in the design of an autonomous robot is navigation, for which, the navigation planning is one of the most vital aspect of an autonomous robot. Therefore, the space and how it is represented play a pivotal role in any problem solution in the domain of the mobile robot, because:

- It provides the necessary information to do path planning.
- It gives information for monitoring the position of the robot

during the execution of the planned path.

Several models have been applied for environment where the principle of navigation is applied to do path planning. For example, A grid model has been adopted by many researchers, where the robot environment is dividing into many line squares and indicated to the presence of an object or not in each square. On line encountered unknown obstacle are modelled by piece of “wall”, where each piece of “wall” is a straight-line and represented by the list of its two end points. This representation is consistent with the representation of known objects, while it also accommodates the fact the only partial information about an unknown obstacle can be obtained from sensing at a particular location.

Many researches which have been done within this field, some of them used a “visibility graph” to set up a configuration space that can be mapped into a graph of vertices between which travel is possible in a straight line. The disadvantage of this method is time consuming. At the opposite, some researches have been based on dividing the world map into a grid (explained before) and assign a cost to each square. Path cost is the sum of the cost of the grid squares through which the path passes. A grid model has been adopted by many authors, where the robot environment is divided into many squares and indicated to the presence of an object or not in each square. A cellular model, in other hand, has been developed by many researchers where the world of navigation is decomposed into cellular areas, some of which include obstacles. More, the skeleton models for map representation in buildings have been used to understand the environment’s structure, avoid obstacles and to find a suitable path of navigation. These researches have been developed in order to find an efficient automated path strategy for mobile robots to work within the described environment where the robot moves.

#### *E. Motion planning*

The goal of the navigation process of mobile robots is to move the robot to a named place in a known, unknown or partially known environment. In most practical situations, the mobile robot can not take the most direct path from the start to the goal point. So, path planning techniques must be used in this situation, and the simplified kinds of planning mission involve going from the start point to the goal point while minimizing some cost such as time spent, chance of detection, or fuel consumption.

Often, a path is planned off-line for the robot to follow, which can lead the robot to its destination assuming that the environment is perfectly known and stationary and the robot can rack perfectly. Early path planners were such off-line planners or were only suitable for such off-line planning. However, the limitations of off-line planning led researchers to study on-line planning, which relies on knowledge acquired from sensing the local environment to handle unknown obstacles as the robot traverses the environment. Moreover, when a robot moves in a specific space, it is necessary to select a most reasonable path so as to avoid collisions with

obstacles. Several approaches for path planning exist for mobile robots, whose suitability depends on a particular problem in an application. For example, behavior-based reactive methods are good choice for robust collision avoidance. Path planning in spatial representation often requires the integration of several approaches. This can provide efficient, accurate, and consistent navigation of a mobile robot.

The major task for path-planning for single mobile robot is to search a collision –free path. The work in path planning has led into issues of map representation for a real world. Therefore, this problem considered as one of challenges in the field of mobile robots because of its direct effect for having a simple and computationally efficient path planning strategy. For path planning areas, it is sufficient for the robot to use a topological map that represents only the different areas without details such as office rooms. The possibility to use topological maps with different abstraction levels helps to save processing time. The static aspect of topological maps enables rather the creation of paths without information that is relevant at runtime. The created schedule, which is based on a topological map, holds nothing about objects which occupy the path. In that case it is not possible to perform the schedule. To get further actual information, the schedule should be enriched by the use of more up-to date plans like egocentric maps.

Topological path planning is useful for the creation of long – distance paths, which support the navigation for solving a task. Therefore, those nodes representing for example, free region space are extracted from a topological map, which connect a start point with a target point. The start point is mostly the actual position of the robot. To generate the path, several sophisticated and classical algorithms exist that are based on graph theory like the algorithm; of the shortest path. To give best support for the path planning it could be helpful to use different abstraction levels for topological maps. For example, if the robot enters a particular room; of an employee for postal delivery, the robot must use a topological map that contains the doors of an office building and the room numbers.

Topological maps can be used to solve abstract tasks, for example, to go and retrieve objects whose positions are not exactly known because the locations of the objects are often changed. Topological maps are graphs whose nodes represent static objects like rooms, and doors for example. The edges between the nodes are part’s relationships between the objects. For example, an abstract task formulated Systems that control the navigation of a mobile robot are based on several paradigms. Biologically motivated applications, for example, adopt the assumed behavior of animals. Geometric representations use geometrical elements like rectangles, polygons, and cylinders for the modeling of an environment. Also, systems for mobile robot exist that do not use a representation of their environment. The behavior of the robot is determined by the sensor data actually taken. Further approaches were introduced which use icons to represent the

environment.

Several approaches for path planning exist for mobile robots, whose suitability depends on a particular problem in an application. For example, behavior-based reactive methods are good choice for robust collision avoidance. Path planning in spatial representation often requires the integration of several approaches. This can provide efficient, accurate, and consist navigation of a mobile robot. It is sufficient for the robot to use a topological map that represents only the areas of navigation (free areas, occupied areas of obstacles). It is essential the robot has the ability to build and uses models of its environment, which enable it to understand the environment's structure. This is necessary to understand orders, plan and execute paths.

Path planning in spatial representation often requires the integration of several approaches. This can provide efficient, accurate, and consist navigation of a mobile robot. It is sufficient for the robot to use a topological map that represents only the areas of navigation (free areas, occupied areas of obstacles). It is essential the robot has the ability to build and uses models of its environment that enable it to understand the environment's structure. This is necessary to understand orders, plan and execute paths. In this paper, a simple and efficient navigation approach for autonomous mobile robot is proposed in which the robot navigates, avoids obstacles and attends its target. Note that, the algorithm described here is just to find a feasible and flexible path from initial area source to destination target area, flexible because:

- The user can change the position of obstacles it has no effect since the environment is unknown.
- The robust method can deal a wide number of environments.
- The algorithm gives the best decision, recognition, and acting.
- Avoiding obstacles and attending target are done in accurate way.

#### IV. THE PROPOSED HYBRID INTELLIGENT SYSTEM ES\_GA\_FL\_NN FOR NAVIGATION PLANNING OF IAV

Today, researchers have at their disposal, the required hardware, software, and sensor technologies to build IAV. More, they are also in possession of a computational tool such as FL,NN,ES and GA that are more effective in the design and development of IAV than the predicate logic based methods of traditional Artificial Intelligence. Fuzzy Logic FL, Neural Networks NN, Genetic Algorithms GA and Expert System (or another combination such as Hidden Markov Model HMM, etc.). ES are well established as useful technologies that complement each other in powerful hybrid system. The first and most advanced integration of intelligent technologies is the hybrid GA and FL; it has been shown after several demonstrated works that the efficacy of this combination is very successful. On the other hand, Hybrid intelligent systems are now part of the repertoire of computer systems developers and important research mechanisms in the study of Artificial

Intelligent [4,5]. The integration of these technologies (for example this combination : NN, FL,ES,GA) has proven to be a way to develop useful real-world applications, and hybrid systems involving all possible complementary integrations are well established as useful technologies that complement each other in powerful hybrid system. For our case for example, the major thrust of our type of combining ES,NN,FL,GA is to synthesis the capability of ES to capture expert domain knowledge in an inference –based system.

##### A. *Evolutionary Genetic Algorithm (GA)*

GAs are search algorithms based on the mechanics of natural genetics. A genetic algorithm for an optimization problem consists of two major components. First, GA maintains a population of individual corresponds to a candidate solution and the population is a collection of such potential solutions [5]. In GA, an individual is commonly represented by a binary string the mapping between solutions and binary strings is called a “coding”. GA has been theoretically and empirically proven to provide robust search capabilities in complex spaces offering a valid approach to problems requiring efficient and effective searching.

Before the GA search starts, candidates of solution are represented as binary bit strings and are prepared. This is called a population . A candidate is called a chromosome (in our case: the path is a “chromosome” and positions are the “genes”). Also, an evolution function, called fitness function, needs to be defined for a problem to be solved in order to evaluate chromosome. As fitness function, we should define distance for each chromosome to give an evaluation function. This evaluation is the goal of the GA search and goes as follows: two (02) chromosomes are chosen randomly from a population are mated and they go through operations like the crossover to yield better chromosomes for next generations. This is repeated until about twelve populations with new chromosomes. To determine execution of the GA, we must specify a stopping criterion, in our case, it could be determine, by a probabilistic function: as we have four chromosomes and we choose randomly two chromosomes, to combine and to compare one path with itself. The crossover is the comparison operator, see the figure. Therefore, after several generations of GA search (The problem of mutation) relatively low fitness of chromosomes remain in a population and some of them are chosen as the solution of the problem (the most preferable path), see the figure2.

Gas are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. An occasional new part is tried for good measure avoiding local minima. While randomized, GA is no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance. In the figure 3 we present genetic algorithm chart.

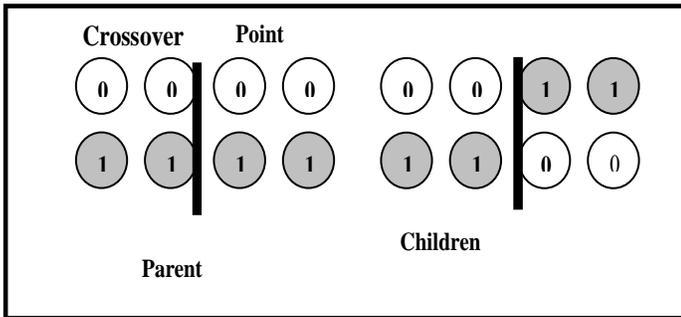


Fig. 1 Example of Crossover on singlepoint

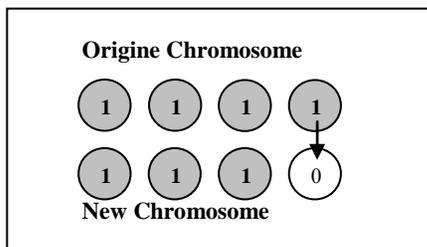


Fig. 2 Example of Mutation in the fourth bit

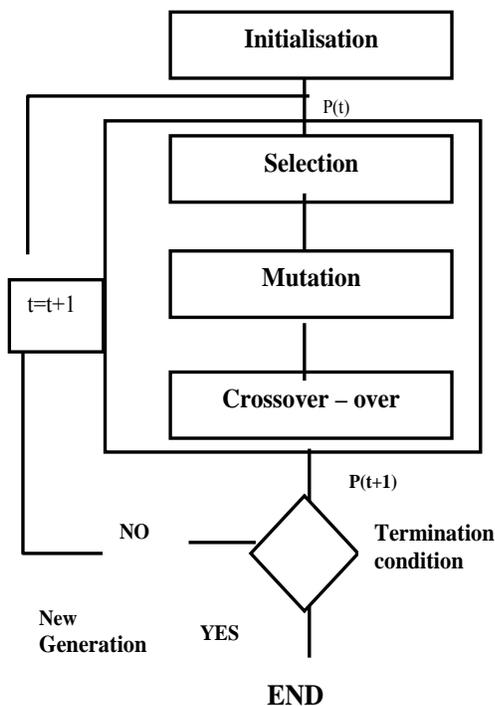


Fig. 3 Genetic Algorithm Chart

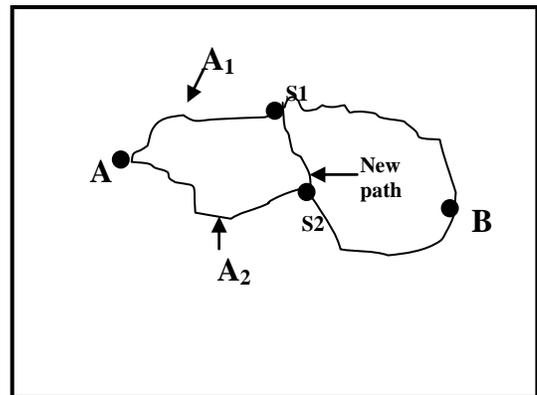


Fig. 4 Crossover operation

over various environments, we observed simulation of the GA for N 5 (we start by no obstacle until we get N MAX Maximum Number) of obstacles environments. We can change the position of obstacles so we get other different environments. These environments were randomly generated. To find a new optimal path after insertion of deletion of an obstacle, we measure the number of generations of candidates. The coding of GA is to affect label 0 for free cell and 1 for hazardous cell. This way of work is very useful later if the substring is inherited to new generations by genetic operators:

*Crossover process*

We have designed an operator that swaps two substrings of two genitors. However two children paths are generated which inherit some of the properties (substrings) of their two parents. Let two paths  $A_1$  and  $A_2$ . Two points  $S_1 \in A_1$  and  $S_2 \in A_2$  are randomly Selected, from which two new paths  $A_3$  and  $A_4$  from start to goat are generated by using the linking and concatenation operators (see the figure 4).

We note that: the path is chromosome and the positions are genes. In theory any path of the  $S_{AB}$  can be obtained by the crossover operator. However, due to the chosen probabilistic distribution in the calculus. The children have a great probability of lying in the region bounded by the parents that demonstrates the need of mutations to explore new regions. Let us to explain more:

*path  $A_3$*

- 1- Path  $A_1$  from A to  $S_1$ .
- 2- New path from  $S_1$  to  $S_2$ .
- 3- Path  $A_2$  from  $S_2$  to B.

*Path A<sub>4</sub>*

- 1- Path A<sub>2</sub> from A to S<sub>2</sub>.
- 2- New path from S<sub>2</sub> to S<sub>1</sub>.
- 3-Path A<sub>1</sub> from S<sub>1</sub> to B

*Selection process*

The selection is done according to a probability proportional to the performance (fitness in the classical literature). The selection is given by P:

$$p = \frac{2^n - 4}{2^m} \quad (1)$$

Performance index may tend to include paths neighboring a local minimum. then the population iteratively Where **n** is the code number of paths designed to be candidates of selection of two paths, **m** is the code number of all paths. At the beginning, a population of paths is created by the mutation operator between A1 = A and A2= B. the size **n** of the population is a critical parameter of all the genetic algorithms: if the number of individuals is small, the region of terrain explored at the beginning of the search are limited, and then the population iteratively generated for optimizing a performance index may tend to include paths neighboring a local minimum. On the other hand, a large population allows the generation of many individuals covering most of the terrain, and has a good chance to find all the optimal and near – optimal solution, but the population will also include less interesting solutions and furthermore the computing time may be high. The better chromosome which has less cost of path (the shortest path) yields after progenitor (several mutations and generation). The criteria of progenitor is stopped after (2<sup>n</sup>-4) of generations, where **n** is the number of paths, we subtract with number 4 because we cannot compare and combine a path with itself. The fitness function for each path is the number of pixels belonging to this path. (in the literature the fitness function is the performance that evaluates and gives a meaning of each chromosome). For improving iteratively the performances of the individuals in the population, the best individuals are preferred to serve as parents in the next crossover operations.

*Mutation Process*

Two selected chromosome are cut in some positions and glued together: the first part of the first chromosome with the second part of the second chromosome, and the first part of the second chromosome with the second part of the first chromosome (see fig1). The mutation is responsible for fine turning values of coordinates of the genes listed in the chromosome: if a gene (sub position) of a chromosome (path) is selected for the mutation, its coordinates are modified to get the best chromosome. The coordinates are changed in the following way:

$$\begin{aligned} x &= x_0 \pm tx_0 \\ y &= y_0 \pm ty_0 \end{aligned} \quad (2)$$

This is done for every (X<sub>0</sub>,Y<sub>0</sub>) initial coordinate of gene to be permuted to give (x', y') new coordinate of gene *t* is the current

generation number of the evolution process. The ± changes on depend of randomly bit *r*:

$$\begin{aligned} \text{if } r = 1 : x' &= x_0 - tx_0 \\ \text{if } r = 0 : x' &= x_0 + tx_0 \end{aligned} \quad (3)$$

$\Delta tx_z$  returns the small piece difference of [0,z] such that the probability of  $\Delta(t, z)$  being close to Zero as **t** increases. The principle is based on three operators:

*Insertion*

This operator inserts a new chromosome (new part) into existing path; every place between **n** positions has the same probability of such insertion.

*Deletion*

This operator deletes a gene (sub position) from the chromosome (path).

*Swap*

This operator splits the selected chromosome into two parts (splitting point is determined at random) and swaps their parts.

*B. The proposed GA work*

Assume that path planning is considered in a square terrain and a path between two locations is approximated with a sequence of adjacent cells in the grid corresponding to the terrain. The length A(α,β) from cell "α" to its adjacent cell "β" is defined by the Euclid distance from the center cell "α" of one cell to the center cell "β" of another cell. Each cell in this grid is assigned of three states: *free*, *occupied*, or *unknown* otherwise. A cell is *free* if it is known to contain no obstacles, *occupied* if it is known to contain one or more obstacles. All other cells are marked *unknown*. In the grid, any cell that can be seen by these three states and ensure the visibility constraint in space navigation.

We denote that the configuration grid is a representation of the configuration space. In the configuration grid starting from any location to attend another one, cells are thus belonging to reachable or unreachable path. Note that the set of reachable cells is a subset of the set of free configuration cells, the set of unreachable cell is a subset of the set of occupied configuration cells. By selecting a goal that lies within reachable space. we ensure that it will not be in collision and it exists some "feasible path" such that the goal is reached in the environment. Having determined the reachability space, the algorithm works and operates on the reachability grid. This one specifies at the end the target area. The detection of the three states is done by the different color of pixels of those belonging to the area obstacle. Generally, the detected different colors of pixels have the same luminous intensity for every free path (a less difference). The other color neighbors are belonging to obstacle area. This detection is based on the game of every detected color of pixel. We separate between the set of luminous intensity of free path of pixels with those belonging to the set of luminous intensity of obstacle and unknown area. This separation is very useful to get a meaning of segmentation. A grid of (i X j) dimension of free path is denoted by "X," an occupy grid of (l X k) is denoted by "Y". An obstacle is collection of hazardous cells in the "Y" grid. A

path from start cell “C” to destination cell “D” that the detected color of “X” does not interest any detected color “Y”. the path is said to be monotone of free cells “X” with respect to i-coordinates if no lines parallel to k-axis cross the j-axis.

The proposed algorithm here relies on number of cells and iterates, as follows:

- 1) i by j grid, start cell a in the grid.
- 2) Detect free destination in the grid (free cells).
- 3) Detect the collection of cells in the grid corresponding to obstacle area (hazardous occupied cells area) and unknown cells.
- 4) A path from “C” to “D” such that the total of neighboring cells is detected free.
- 5) If the collection of free cells is continuous, detect all neighboring on the same destination until the target is reached.
- 6) If the collection of free cells is discontinuous, change the direction and continue on another free continuous collection of cells.

To maintain the idea; we have created several environments which contain many obstacles. The search area (environment) is divided into square grids. Each item in the array represents one of the squares on the grid, and its status is recorded as walkable or unwalkable area (obstacle). The robot can identify two colors inside our environment: dark and white. The dark color is interpreted as an obstacle area (Also affected for the neighbouring pixel area which each pixel has the same approximate luminous intensity value of those pixels belonging to obstacle area); whereas the white color represents the free trajectory to attend the given target area (Also affected for the neighbouring pixel area which each pixel has the same approximate luminous intensity value of those pixels belonging to free trajectory area); The robot starts from any position then it must move by sensing and avoiding the obstacles. The trajectory is designed in form of a grid-map, when it moves it must verify the adjacent case by avoiding the obstacle that can meet to reach the target. We use an algorithm containing the information about the target position, and the robot will move accordingly.

To determine the nature of space of navigation, and as we have illustrated before, cells are marked as free or occupied; otherwise unknown. We can therefore divide our search area into free and occupied area. Note that all free space cells represent the walkable space and unwalkable in occupied space. Each free cell is able of laying all the neighbor free cell within a certain distance “d”. This distance “d” is usually set to a value greater than or equal to the size of cell. Note that the set of free cells is a subset of free cells, which is in turn a subset of the set of free occupancy cells. Thus, by selecting a goal that lies within free space, we ensure that the free sub-path will not be in collision with the environment, and that there exists some sub-paths to get the target. Note that, we determine the free resultant cells within free space to get a feasible path during navigation. For unwalkable space (occupied space) we just develop a procedure of avoiding

d  
u

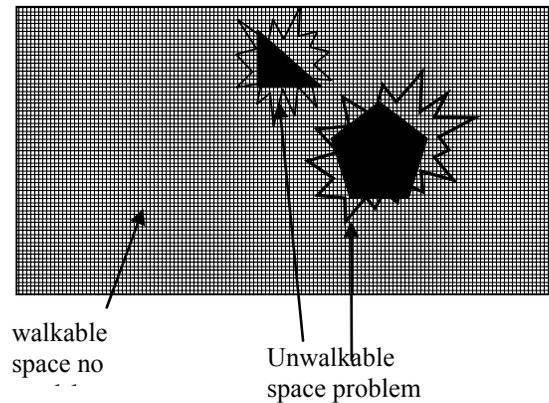


Fig. 5 Example of walkable space and walkable space

For unwalkable space, we compute the total size of free cells around danger (obstacle) area. This total may be at least greater or equal than to the length of architecture of robot. This is ensure the safety to our robot to not be in collision with the obstacle, and that the path P has enough security SE to attend it target where it is given by  $P \pm SE$  (S is size of security). In principle, we generate a plan for reaching safety area for every neighboring danger area. The safety distance is generated to construct the safety area building to the navigation process, to be near without collision within this one.

### C. Expert System

An ES is a computer program that functions, is in a narrow domain, dealing with specialized knowledge, generally possessed by human experts. ES is able to draw conclusions without seeing all possible information and capable of directing the acquisition of new information in an efficient manner.

### D. Fuzzy Logic

The implied natural language is represented through fuzzy sets involving classes with gradually varying transition boundaries [8]. As human reasoning is not based on the classical two-valued logic, this process involves fuzzy truths, fuzzy deduction rules, etc. This is the reason why FL is closer to human thinking and natural language than classical logic. Furthermore, to build machines that are able to perform complex task requiring massively parallel computation, knowledge of the environment structure and interacting with it involves abstract appreciation of natural concepts related to, the proximity, degree of dangers, etc. Also, FL can be viewed as an attempt to bring together conventional precise mathematics and human-like decision-making concepts.

The fuzzy model, treated with *ES\_GA\_FL\_NN* is presented in Figure 6. where :

- $A_1$  : The static danger degree for the first path.
- $B_1$ : The safety degree corresponding to the first path.
- $A_2$ : The static danger degree for the second path
- $B_2$ : The safety degree corresponding to the second path.
- $A_n$  : The static danger degree for the  $n$  path.
- $B_n$ : The safety degree corresponding to the  $n$  path.
- $D_n$ : Intermediate distance
- $C_n$ : The avoidance direction for the  $n$  best path.
- $P$ : the best path.

At first, from a given detected position  $P_1$  (Pixel by pixel), and from a given visual position  $P_g$  the vehicle gets absolute obstacle position  $P_1$  and  $P_g$  from current point  $P_1$  to calculate the angles  $A_n$  and  $B_n$ . Then, after updating its position ( $P_1$  becomes  $P_g$ ) and the target position ( $P_g$  becomes the target one) based on its Cartesian coordinates, the vehicle must avoid the obstacle to get the target position. Afterwards, the vehicle recognizes the static danger degree  $A_n$  and safety degree  $B_n$  between itself and the obstacle using a fuzzy reasoning and inference. Second, using  $A_n$  and  $B_n$  the vehicle decides the avoidance direction  $C_n$  by a decision table written with productions rules and then avoidance direction vector as it is shown in figure 10. The main problem of this approach is that does not encounter several obstacles at the same time and does not take into account the obstacles sizes. The static danger degree  $A$  and safety degree  $B$  are given by:

$$A_n = \tan^{-1}((Y_i - Y_1)/(X_i - X_1)) \tag{4}$$

$$B_n = \tan^{-1}((Y_g - Y_1)/(X_g - X_1))$$

Where : the points  $P_1 (X_1, Y_1)$ ,  $P_i (X_i, Y_i)$  and  $P_g(X_g, Y_g)$  are the co-ordinate of respectively to current point, intermediate point and visual point (we calculate point to point until the visual point becomes the target one). In the Figure 7, we present the membership labels for distance  $D_n$  which are defined as Near Danger (ND), and Far Danger (FD). The membership functions of directions  $A_n$  and  $B_n$  are represented in Figure 8, where fuzzy labels are defined as left Danger Near (LDN), Right Danger Near (RDN), Left Safety (LS), Right Safety (RS). The membership function of direction  $C_n$  are shown in Figure9, where fuzzy labels are defined as : Left Danger (LD), Left Safety Small (LSS), Left Danger Far (LDF), Left Safety Big (LSB), Right Danger (RD), Right Safety Small (RSS), Right Danger Far (RDF), Right Safety Big (RSB). The final results are given as a guide and decision of steering vector.

The final decision is given by:

$$G = \frac{\sum(\mu_i * \beta_i)}{\sum \mu_i} \tag{5}$$

Where

- $1 \leq i \leq m$ .
- $m$ : number of rule.
- $B$ : centroid of the backend of membership function correspond for each rule.
- $\mu$ : factor of membership correspond for each rule .

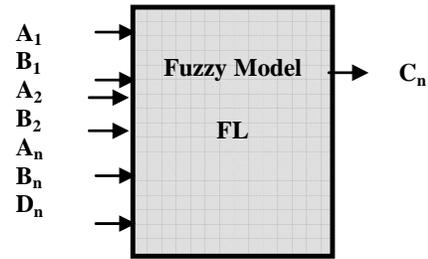


Fig. 6 Fuzzy Model

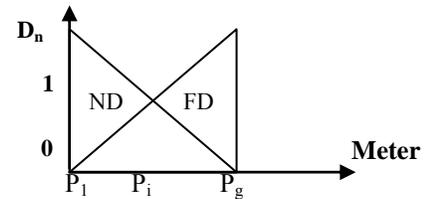


Fig. 7 Membership functions of distance  $D_n$

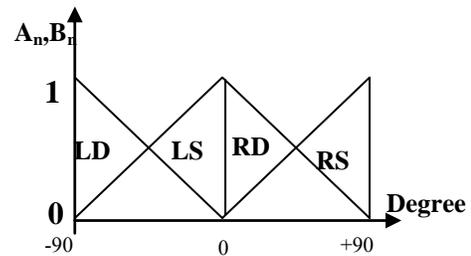


Fig. 8 Membership functions of distance  $D_n$

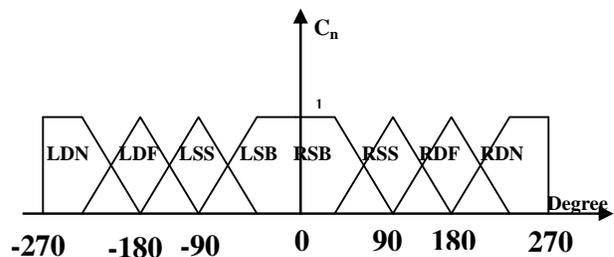


Fig. 9 Membership functions of distance  $D_n$

IF (A<sub>n</sub> or (B<sub>n</sub>) is LD and D is ND ) Then (C<sub>n</sub> is LDN )  
 IF (A<sub>n</sub> or (B<sub>n</sub>) is LD and D is FD ) Then (C<sub>n</sub> is LDF )  
 IF (A<sub>n</sub> or (B<sub>n</sub>) is LS and D is ND ) Then (C<sub>n</sub> is LSS )  
 IF (A<sub>n</sub> or (B<sub>n</sub>) is LS and D is FD ) Then (C<sub>n</sub> is LSB )  
 IF (A<sub>n</sub> or (B<sub>n</sub>) is RD and D is ND ) Then (C<sub>n</sub> is RDN )  
 IF (A<sub>n</sub> or (B<sub>n</sub>) is RD and D is FD ) Then (C<sub>n</sub> is RDF )  
 IF (A<sub>n</sub> or (B<sub>n</sub>) is RS and D is ND ) Then (C<sub>n</sub> is RSS )  
 IF (A<sub>n</sub> or (B<sub>n</sub>) is RS and D is FD ) Then (C<sub>n</sub> is RSB )

Fig. 10 Fuzzy Rule Base

Neural Networks

Historically, interest in NN stems from the wish to understand principles leading in some manner to the comprehension of the human brain functions and to build machines that are able to perform tasks requiring massively parallel computation[6]. Essentially, NN deal with cognitive tasks such as learning, adaptation, generalization and optimization. Networks of neurons can achieve complex classification based on the elementary capability of each neuron to distinguish classes through its activation function. NN must provide Robot with capacities to successfully navigate during the navigation. The robot must learn to build a map ( i.e. target, obstacles, and free spaces from sensors). The robot must learn to decide the angle C<sub>n</sub> avoidance formulation using NN from fuzzy linguistic formulation of human expert knowledge. This FL\_NN is trained to capture the human expert behavior in the decision- making operation C<sub>n</sub>. Our Neural networks NN are built of four (04) layers as shown in the figure 11; which are

Layer 1

This layer is the input layer with two angles [An, Bn ] . it transmits inputs to their corresponding me membership functions in the next layer.

Layer 2

This second layer performs the fuzzification operation and contains input membership functions according to the respective linguistic input with four (04) nodes. Each node calculate the degree of the measure data belonging to the n<sup>th</sup> membership function for the input variable, e.g. for the input variable [D<sub>n</sub>], we obtain ( u<sub>N</sub> [D<sub>n</sub>], u<sub>F</sub>[D<sub>n</sub>] ) with u<sub>N</sub> (D<sub>n</sub>) and u<sub>F</sub> [D<sub>n</sub>] the membership degrees of fuzzy sets N and F respectively. Idem for the angle An or Bn , we compute { u<sub>N</sub> (LD), u<sub>N</sub> (LS), u<sub>N</sub> (RD), u<sub>N</sub> (RS), u<sub>F</sub> (LD), u<sub>F</sub> (LS), u<sub>F</sub> (RD), u<sub>F</sub> (RS)}.

Layer 3

It is the fuzzy rule base with the eight (08) nodes “number of rule” representing fuzzy rules. The rule association is done among membership function of different inputs. Each node receives only two connections, from the corresponding node { LS, LD, Rs, Rd} of the inputs An or Bn and the one of {N,F} of the second input which is distance.

Layer 4

This final layer with the nodes corresponding to output membership function performs the computing of angle C<sub>n</sub> { LDN, LDF , LSS, LSB, RSB, RSS, RDF, RDN}. This layer corresponds to perform the sum operation ai=Σ Wi where Wi : the weights of the output layer.

$$w_1(t + 1) = w_1(t) + \Delta w_1 \tag{6}$$

With ΔW1= ηδ y, Where η : learning rate such as 0< η<1 (η it is chosen when we compute degree membership function) and Y : hidden output.

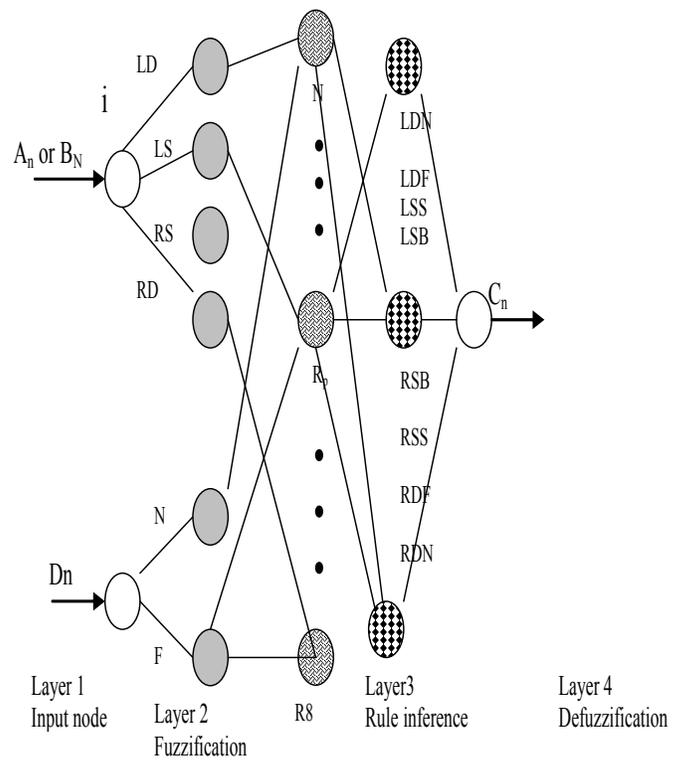
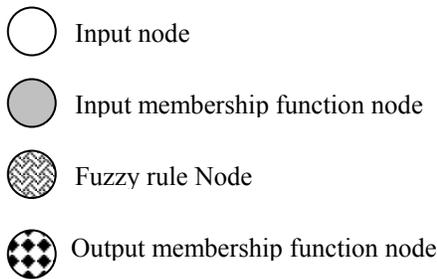


Fig. 11 the hybrid neural fuzzy system

Where



## V. SIMULATION RESULTS

To reflect the vehicle behaviors acquired by learning and to demonstrate generalization and adaptation abilities of *ES\_GA\_FL\_NN*, vehicle is simulated in different static environments. In this context, we have created N unknown environments containing static obstacles; (complexity order of these creations is limited at the last environment one, until now we have tested 64 environments), we start with no obstacle until the complexity order is done. As there is no information at advance, this creation can give another configurations of environments, that means that, the user of this concept can change the positions of all objects as he want in the scene and can change the shapes of obstacle (big, small, different sizes, ...), this have no effect since the environment is unknown, the robot success, in satisfactory manner, to avoid suitably the static obstacles while it makes one's way toward its target, we can give different infinite environment complexity, in order to achieve the desired task.

Tested in different unknown environments with static obstacles, we present simulation results which provides the most preferable path between another one treated. As it is illustrated In Figure 11.a where S: Robot and B: Target, the vehicle succeeds to avoid obstacles and reaches its target. In this case, we present virtually the best optimum path, e.g. the robot doesn't endanger itself or other objects in the environment. At advance, the robot navigates virtually to structure the environment, and one or more camera are used for the perception which can guarantee to deliver acceptably accurate information all of the time. Also, the redundancy is useful (sensor data fusion), the robot receives a good deal of attention and recognizes all elements of the scene of navigation and learned where are situated the safety section to evolve and where the danger sections to avoid. After learning, the final decision is given as guide of steering vector. In this case, the robot is supposed not as square (8 X 8pixels), it is replaced by point material and the path is a set of positions of all points of navigation.

The user can change the shape (body) of robot to execute the final path by gravity center (but the size of the vehicle is taken into account). We replace the body of vehicle by gravity center (material point) to execute the path truly. Before, the optimum path has been calculated and the accurate avoidance

direction is known (Figure12), so now the robot knows at advance how to evolve and where is situated from the target (Figure13). The final decision is taken and the best path to execute is selected, the robot can evolve without risk. These results display the *ES\_GA\_FL\_NN* ability making IAV able to intelligently avoid obstacles with different architectures. In the figure 14 we present another environment where the navigation is done in complex environment. The robot knows at advance how to evolve and where is situated from the target (Figure15). The final decision is taken and the best path to execute is selected, the robot can evolve without risk.

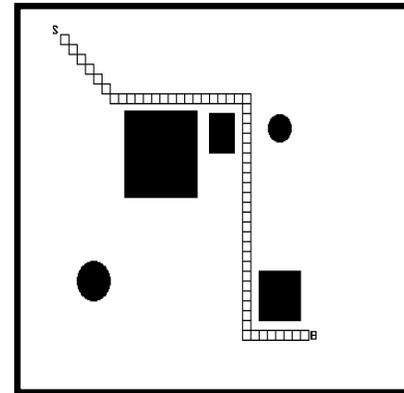


Fig. 12 Example of simple environment of navigation of an intelligent autonomous system : The reached best path

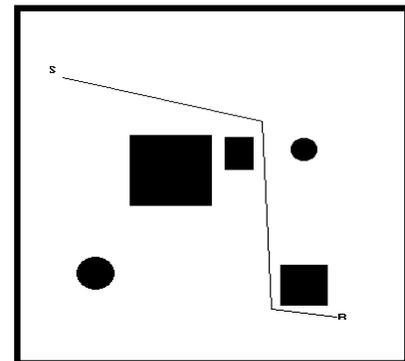


Fig. 13. Example of simple environment of navigation of an intelligent autonomous system : The final decision to be taken to execute the best path

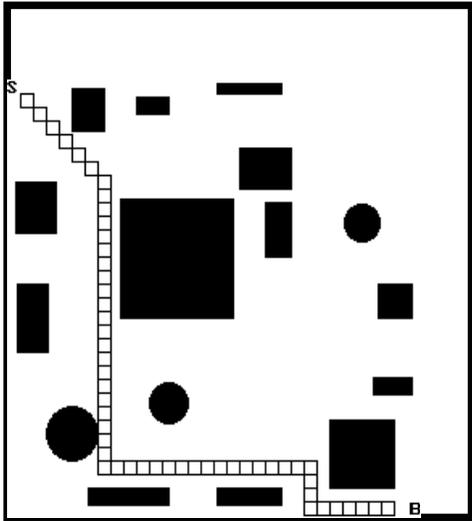


Fig. 14 Example of complex environment of navigation of an intelligent autonomous system: The reached best path

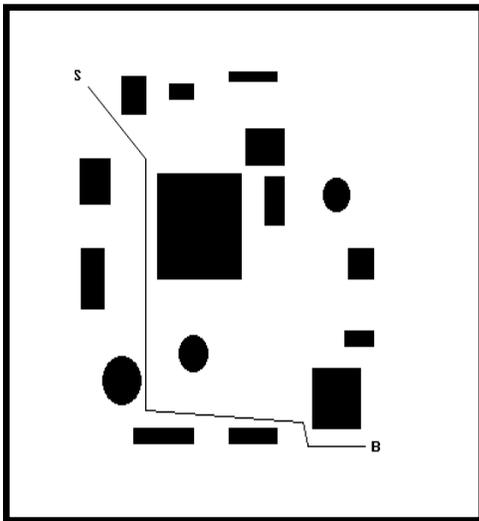


Fig. 15 Example of complex environment of navigation of an intelligent autonomous system: The final decision to be taken to execute the best path

## VI. CONCLUSION

The theory and practice of Intelligent autonomous systems are currently among the most intensively studied and promising areas in computer science and engineering which will certainly play a primary goal role in future. These theories and applications provide a source linking all fields in which intelligent control plays a dominant role. Cognition, perception, action, and learning are essential components of

such-systems and their use is tending extensively towards challenging applications (service robots, micro-robots, bio-robots, guard robots, warehousing robots). In this paper, we have presented a software implementation of navigation approach of an autonomous mobile robot in an unknown environment using hybrid intelligent. Indeed, the main feature of *ES\_GA\_FL\_NN* is the use of the best path of biological genetic principle combined with networks in the task fuzzy reasoning and inference capturing human expert knowledge to decide about the best avoidance direction getting a big safety of obstacle danger. Besides, the proposed approach can deal a wide number of environments. This system constitutes the knowledge bases of *ES\_GA\_FL\_NN approach* allowing recognizing situation of the target localization and obstacle avoidance, respectively.

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

## REFERENCES

- [1] D. Estrin, D.Culler, and K.Pister, *PERVASIVE Computing IEEE*, 2002, pp. 59-69G.
- [2] T. Willeke, C. Kunz, and I.Nourbakhsh, "The Personal Rover Project : the comprehensive Design Of a domestic personal robot", *Robotics and Autonomous Systems (4)*, Elsevier Science, 2003, pp.245-258.
- [3] L. Moreno, E.A Puente, and M. A.Salichs, " World modelling and sensor data fusion in a non static environment : application to mobile robots", in *Proceeding International IFAC Conference Intelligent Components and Instruments for control Applications*, Malaga, Spain, 1992, pp.433-436.
- [4] S. Florczyk, "Robot Vision Video-based Indoor Exploration with Autonomous and Mobile Robot", WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2005.
- [5] A. Gonzalez and R. Perez. , "SLAVE : A genetic learning System Based on an Iterative Approach", *IEEE, Transaction on Fuzzy systems*, volume7, number2, April 1999, pp.176-191.
- [6] J. A. Freeman and D.M. Skapura , " Neural networks : algorithms , application, and programming techniques", Addison-Wesley, New York ,1992.
- [7] L. R. Medsker, *hybrid intelligent systems*, Kluwer Academic Publishers, 1995.
- [8] W. Pedrycz " Relevancy of fuzzy models", *Information Sciences 52* (1990), pp.285-302.
- [9] K. Schilling and C.Jungius," Mobile robots for planetary exploration", in *proceeding second international conference IFAC*, Intelligent Autonomous.
- [10] S. Florczyk,"Robot Vision Video-based Indoor Exploration with Autonomous and Mobile Robots", WILEY-VCH Verlag GmbH & Co. KGaA, 2005,Weinheim.
- [11] O. Hachour and N. Mastorakis, "IAV: A VHDL methodology for FPGA implementation", *WSEAS transaction on circuits and systems*, Issue5, Volume3, ISSN 1109-2734, 2004, pp.1091-1096.
- [12] O. Hachour AND N. Mastorakis, "FPGA implementation of navigation approach", *4<sup>th</sup> WSEAS international multicongference robotics, distance learning and intelligent communication systems (ICRODIC 2004)*, in Rio de Janeiro Brazil, October 1-15 , 2004, pp2777-2784.
- [13] O. Hachour AND N. Mastorakis, "Avoiding obstacles using FPGA –a new solution and application" ,*5<sup>th</sup> WSEAS international conference on automation & information (ICAI 2004)* , *WSEAS transaction on systems* , issue9 ,volume 3 , Venice , Italy15-17 , November 2004 , ISSN 1109-2777, pp2827-2834.
- [14] O. Hachour AND N. Mastorakis, "Behaviour of intelligent autonomous ROBOTIC IAR", *IASME transaction*, issue1, vol.1, ISSN 1790-031x WSEAS, January 2004, pp 76-86.

- [15] O. Hachour, "The proposed Genetic FPGA Implementation for path planning of Autonomous Mobile Robot", *International Journal of Circuits, Systems and signal processing*, ISSN: 1998-4464, Issue 2, Vol. 2, 2008, PP151-167.
- [16] O. Hachour, path planning of Autonomous Mobile Robot", *International Journal of Systems Applications, Engineering & Development*, Issue4, vol.2, 2008, pp178-190.