Path planning of Autonomous Mobile robot

O.Hachour

***Abstract*—**in this present work, we present an algorithm for path planning to a target for mobile robot in unknown environment. The proposed algorithm allows a mobile robot to navigate through static obstacles, and finding the path in order to reach the target without collision. This algorithm provides the robot the possibility to move from the initial position to the final position (target). The proposed path finding strategy is designed in a grid-map form of an unknown environment with static unknown obstacles. The robot moves within the unknown environment by sensing and avoiding the obstacles coming across its way towards the target. When the mission is executed, it is necessary to plan an optimal or feasible path for itself avoiding obstructions in its way and minimizing a cost such as time, energy, and distance. The proposed path planning must make the robot able to achieve these tasks: to avoid obstacles, and to make ones way toward its target. The algorithms are implemented in Borland C++, afterwards tested with visual basic and DELPHI programming language; whereby the environment is studied in a two dimensional coordinate system. The simulation part is an approach to the real expected result; this part is done using C++ to recognize all objects within the environment and since it is suitable for graphic problems. Taking the segmented environment issued from C++ development, the algorithm permit the robot to move from the initial position to the desired position following an estimated trajectory using visual basic and Delphi language.

***Keywords*—**Intelligent Autonomous Systems (IAS), navigation, Path planning.

I. INTRODUCTION

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, it is necessary to plan a collision free path minimizing a cost such as time, energy and distance. When an autonomous 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. Therefore, the major main work for path planning for autonomous mobile robot is to search a collision free path.

Many works on this topic have been carried out for the path planning of autonomous mobile robot. Motion planning is one of the important tasks in intelligent control of an autonomous mobile robot. It is often decomposed into path planning and trajectory planning. Path planning is to generate a collision free path in an environment with obstacles and optimize it with respect to some criterion [6,9]. Trajectory planning is to schedule the movement of a mobile robot along the planned path. Several approaches have been proposed to address the problem of motion planning of a mobile robot. If the environment is a known static terrain and it generates a path in advance it said to be off-line algorithm. It is said to be on-line if it is capable of producing a new path in response to environmental changes.

A robotic systems capable of some degree of self-sufficiency is the overall objective of an Autonomous Mobile robot and are required in many fields [1,2,4,5,7,8]. The focus is on the ability to move and on being self-sufficient while trying to imitate the biology. Indeed, biological models are of major interest since living systems are prototypes of autonomous behaviours. IAS have many possible applications in a large variety of domains, from spatial explorations to handling material, and from industrial tasks to the handicapped helps. In fact, recognition, learning, decision-making, and action constitute principal problems of the obstacle avoidance of IAS. Three levels are required to recognition namely: inaccurate data processing (issued from sensors), construction of knowledge base, and establishment of an environment map. To solve these problems and remedy insufficiencies of classical approaches related to real-time, autonomy, and intelligence, current approaches are based on hybrid intelligent systems.

IAS designers search to create dynamic systems o navigate and perform purposeful behaviours like human in real environments where conditions are laborious. However, the environment complexity is a specific problem to solve since the environments can be imprecise, vast, dynamical, and partially or not structured. Then, IAS must then be able to understand the structure of these environments. To reach the target without collisions, IAS must be endowed with recognition, learning, decision-making, and actions capabilities.

The ability to acquire these 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.

Recent research on IAS has pointed out a promising direction for future research in mobile robotics where real-time, autonomy and intelligence have received considerably more weight then, for instance, optimality and completeness. Many navigation approaches have dropped the explicit knowledge representation for an implicit one based on acquisitions of intelligent behaviours that enable the robot to interact effectively with its environment, they have to orient themselves, explore their environments autonomously, recover from failure, and perform whole families of tasks in real-time.

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). Thus, the recent developments in autonomy requirements, intelligent components, multi-robot system, and massively parallel computer have made the IAS very used, notably in the planetary explorations, mine industry, and highways [10,11,12,13,14]. But, the current mobile robots do relatively little that is recognizable as intelligent thinking, this is because:

- Perception does not meet the necessary standards.
- Much of the intelligence is tied up in task specific behavior and has more to do with particular devices and missions than with the mobile robots in general.
- Much of the challenge of the mobile robots requires intelligence at subconscious level.

The motion of mobile robots in an unknown environment where there are stationary unknown obstacles requires the existence of algorithms that are able to solve the path and motion planning problem of these robots so that collisions are avoided. In order to execute the desired motion, the mobile robot navigates intelligibly and avoids obstacles so that the target is reached. The problem becomes more difficult when the parameters that describe the model and /or the workspace of the robot are not exactly known.

The autonomous mobile systems capable of some degree of self-sufficiency are required in many fields and are the primary goal of IAS. The focus is on the ability to move and on being self-sufficient while trying to imitate the biology. Indeed, biological models are of major interest since living systems are prototypes of autonomous behaviours. IAS have many possible applications in a large variety of domains, from spatial exploration to handling material, and from industrial tasks to the handicapped helps. In fact, recognition, learning, decision-making, and action constitute principle problems of the obstacles avoidance of IAS. Three levels are required to recognition namely: inaccurate data processing (issued from sensors), construction of knowledge base, and establishments of an environment map. To solve these problems and remedy in sufficiency of classical approaches related to real-time, autonomy, and intelligence, current approaches are based on hybrid intelligent systems.

IAS designers search to create dynamic systems able to navigate and perform purposeful behaviours like human in real environments where conditions are laborious. However, the environment complexity is a specific problem to solve since these environments can be imprecise, vast dynamical, and partially or not structured. IAS must then be able to understand the structure of these environments. To reach the target without collisions, IAS must be endowed with recognition, learning, decision-making, and action capabilities. The ability to acquire these 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.

Recent research on IAS has pointed out a promising direction for future research in mobile robotics where real-time, autonomy and intelligence have received considerably more weight than, for instance, optimality and completeness. Many navigation approaches have dropped the explicit knowledge representation for an implicit one based on acquisitions of intelligent behaviours with its environments, they have to orient themselves, explore their environments autonomously, recover from failure, and perform whole families of tasks in real-time. However, the mobile robot is appropriate tool for investigating 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.

This paper deals with the intelligent path planning of IAS in an unknown environment. The aim of this paper is to develop an IAS algorithm for the AMR stationary obstacle avoidance to provide them more autonomy and intelligence. A robotic vehicle is an intelligent mobile machine capable of autonomous operation in structured and unstructured environment, it must be able of sensing (perceiving its environment) thinking (planning, reasoning) and acting (moving and manipulating). However, mobile robots are appropriate tools for investigating optional artificial intelligence problems relating to word understanding and taking a suitable action, such as, planning missions, avoiding obstacles, and fusing data from many sources[3]. One of the specific characteristics of mobile robots 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.

The most important key issue in the design of an autonomous robot is the navigation process, which is one of the most vital aspects of an autonomous mobile robot. 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:

- It provides the necessary information to do path panning.
- It gives information for monitoring the position of the robot during the execution of the planned path.

- 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.

A robot is a "device" that responds to sensory input by running a program automatically without human intervention. Typically, a robot is endowed with some artificial intelligence so that it can react to different situations it may encounter. The robot is referred to be all bodies that are modeled geometrically and are controllable via a motion plan.\q robotic vehicle is an intelligent ;mobile machine capable of autonomous operations in structured and unstructured environment. It must be capable of sensing thinking and acting. 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 obstacles, and fusing data from many sources.

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't 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 rely 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.

One of the specific characteristics of mobile robots 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 on an autonomous robot is navigation. The 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 aspect 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.

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. One of the specific characteristics of mobile robots is the complexity of their environment, therefore, one of the critical problem for the mobile robots is path planning. 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 that enable it to understand the environment's structure. This is necessary to understand orders, plan and execute paths.

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 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 [6, 9]. 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.

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. This robust method can deal a wide number of environments and gives to our robot the autonomous decision of how to avoid obstacles and how to attend the target. More, the path planning procedure covers the environments structure and the propagate distances through free space from the source position. For any starting point within the environment representing the initial position of the mobile robot, the shortest path to the goal is traced. The algorithm described here therefore is to develop a method for path planning by using simple and computationally efficient-way to solve path planning problem in an unknown environment without consuming time, lose energy, un-safety of the robot architecture. This paper describes 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.

1) The user can change the position of obstacles it has no effect since the environment is unknown and answers for any user demand.
2) The proposed navigation approach can deal a wide number of environments and gives to our robot the autonomous decision of how to avoid obstacles and how to attend the target.

More, the navigation planning procedure covers the environments structure and the propagate distances through free space from the source position. For any starting point within the environment representing the initial position of the mobile robot, the shortest path to the goal is traced. The algorithm described here therefore is to develop a method for navigation planning by using simple and computationally efficient-way to solve path planning problem in an unknown environment without consuming time, lose energy, un-safety of the robot architecture. The shortest path is obtained without collision. This proposed method has an advantage of adaptivity such that the algorithm works perfectly even if an environment is unknown.

II. Necessity of intelligent Autonomous robot

The theory and practice of IAS 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.

Industrial robots used for manipulations of goods; typically consist of one or two arms and a controller. The term controller is used in at least two different ways . in this context, we mean the computer system used to control the robot, often called a *robot work-station* controller. The controller may be programmed to operate the robot in a number of way; thus distinguishing it from hard automation. The controller is also responsible for the monitoring of auxiliary sensor that detect the presence, distance, velocity, shape, weight, or other properties of objects. Robots may be equipped with vision systems, depending on the application for which they are used. Most often, industrial robot are stationary, and work is transported to them by conveyer or robot carts, which are often called autonomous guided vehicles (AGV). Autonomous guided vehicles are becoming increasingly used in industry for materials transport. Most frequently, these vehicles use a sensor to follow a wire in the factory floor. Some systems employ an arm mounted on an AGV.

Robot programmability provides major advantages over hard automation. If there are to be many models or options on a product, programmability allows the variations to be handled easily. If product models change frequently; as in the automotive industry, it is generally far less costly to reprogram a robot than to rework hard automation. A robot workstation may be programmed to perform several tasks in succession rather than just a single step on a line. This makes it easy to accommodate fluctuations in product volume by adding or removing workstations. Also; because robots may be reprogrammed to do different tasks; it is often possible to amortize their first cost over several products. Robots can also perform many applications that are poorly suited to human abilities. These include manipulation of small and a large object like electronic parts and turbine blades, respectively. Another of these applications is work in unusual environments like clean rooms, furnaces, high-radiation areas, and space. Japan has led the world in the use of robots in manufacturing. The two sectors making heaviest use of robots are the automotive and electronics industries. Interest in legged locomotion has been stimulated by application in traversing rough terrain and in unmanned exploration of unknown environment. Aside from electronic motivation, there are many unanswered scientific question about how biological organism produce the remarkable sensor motor behaviour that we observe. Finally, the notion of simulating biological organism has a certain instinctive reproductive appeal and offers the possibility of satisfying our curiosity as to how come to be as we are.

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. 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. 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 nature 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 sever limits regarding mass volume, power consumption, autonomous reactions capabilities and design complexity. Particularly, for planetary operations sever 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 he 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 consider, 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 demonstrates that obstacle detection and collision avoidance is improved with good results.

Classical artificial intelligence systems presuppose that all knowledge is stored in a central database of logical assertions or other symbolic representation and that reasoning consist largely of searching and sequentially updating that database. While this model has been successful for disembodied reasoning system, it is problematic for robots. Robots are distributed systems; multiple sensory, reasoning, and motor control processes run in parallel, often on separate processor hate rate only loosely coupled with one another. Each of these procure necessarily maintains its own separate, limited representation of the world and task; requiring them to constantly synchronize with the central knowledge base is probably unrealistic. Automated reasoning systems are typically built on a transaction-oriented model of computation .knowledge of the world is stored in a database of assertion in some logical language, indexed perhaps by predicate name. For one problem discussed, for example, the robot should have reflective pools that give the robot access to its own internal state:

The behavioral pool

The behavioural pool holds binding between tag and specific robot behaviour. Each behaviour continually compare it tag to the tag on a global call signal. Whenever a behavioural detects a match, it activates itself. Active behaviour also drive a global running signal with bit-vector of their tag. The signal therefore hold the tag of all running behaviour, allowing any part of the system monitoring the signal to determine whether the behaviour bound to a given tag is running.

The proposition pool

The proposition pool holds bindings between tag and specific binary-valued signal in the system. The pool generates a true? Signal comprised of the set of all tag bound to proposition that are presently true. This allow one component of the system to “pass” a signal to another component by binding it to tag that has been agreed upon in advance. The receiving component can then monitor the signal by inspecting the appropriate bit of the true signal.

The predicate pool

The predicate pool holds binding between tag and unary predicate. The predicate pool generates vector of signal, indexed by role, whose elements hold the extension of all bound predicates – role 0 in element 0 , role 1 in element 1, etc. again, this provides an indirection facility for passing signal between components . In this context, we can include a marker- passing semantic net. Node within the net can be bound to role tag and then propagated a marker along links in the net to perform retrieval and inference from long-term memory.

It is important to understand that a given object or concept might be represented in several of these pools simultaneously, with each pool representing different aspect of the object. This is supported in part by allowing element of different pool to share a sing tag register. For example, the lexicon pool entry of the word “show” , the behaviour SHOW, and the semantic net node representing information about the behaviour all share a common tag register. Therefore, when the parser bind “show” to a role, the behaviour that can implement the verb is automatically bound to the same role at the same time. A several works were demonstrated in this domain, many researchers have attended this problem to give a successful reasoning system. They have discussed a lot of an alternate class of architectures-tagged behaviour-based systems- that support a large subset of the capabilities of classical artificial intelligence architecture, including limited quantified inference, forward and backward-chaining, simple natural language question answering and command following, reification, and computational reflection, while allowing object representation, to remain distributed across multiple sensory and representational modalities.

Navigation

Navigation is the ability to move and on being self-sufficient. The IAS 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.

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.

Besides, the most important key of the navigation system of mobile robot is to move the robot to a named place in known, unknown or partially known environments. In most practical situations the mobile robot can not take the most direct path from the start to the goal point. So, path finding techniques must be used in this situation, and the simples 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. When the robot actually starts to travel along a planned path, it may find that there are surmountable obstacles along the path that were not on the map. When this happens, the robot must chart the obstacle and, if no local avoidance manoeuvre is possible, it must plan a new path with the modified map and its current position as the new staring location. Accordingly, we can divide the task of moving a mobile robot within its environment in a two step process:

1) Planning paths which are optimal by certain criteria's.
2) Controlling the robot to execute the planed paths.

A. Autonomy requirements

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

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 [9]. At ambient temperatures, the limited temperature -sensitive electronic equipment on-board must be placed in a thermally insulated compartments [6] . The thermal environment of Mars challenges the thermal control system. In the course of a Martian day the temperature can vary from 140K to 300K for example.

Energy

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

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 IAS 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.

Mechanical design

The mechanical design of Intelligent Autonomous Robots is the result of an integration approach considering several criteria related with perception, control, and planning issues in addition to structural design and other mechanical requirements.

B. Criteria to satisfy by vehicles to be autonomous and intelligent

To evaluate the performance of IAS which are intelligent and autonomous vehicle, the robot must perform the following criteria:

Intelligence

A robotic system capable of some degree of self-sufficiency is the primary goal of Intelligent Autonomous Vehicles**.** Thus, the robot must achieve his task with more autonomy and intelligence. Also, the vehicle reacts to unknown static and dynamic obstacles with safety not to endanger itself or other objects in the environment. Near Safety, the reliability is taken into account in the field of robotics; it is the probability that the required function is executed without failure during certain duration.[13,14]

Navigation

Navigation is the ability to move and on being self-sufficient. The AMR 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.

III. The Proposed Navigation Approach Procedure for IAS

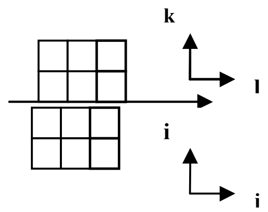
A. Path planning

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,) 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.

In this present work, we consider a grid of (i X) dimension. This grid is divided into : free path denoted by "*X Cell (k*L)*" and occupied path denoted by "*Y* Cell (i*J)" . 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 (see figure 1).



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 are 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 the 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 danger. The figure 2 shows an example of walkable or unwalkable space.

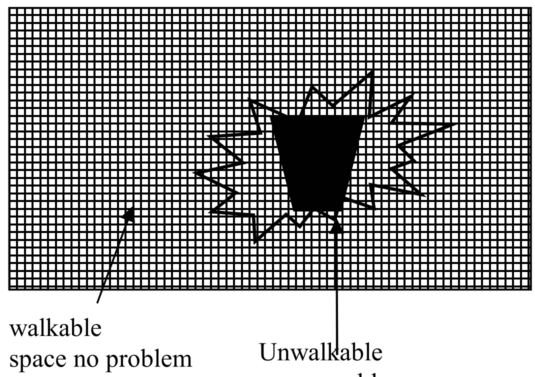


Fig. 2 an example of walkable space and walkable space

For unwalkable space, we compute the total size of free cells around danger (obstacle) area. This total may be at least 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. In the figure 3 we present one example of navigation approach using a square cellule grid for the movement

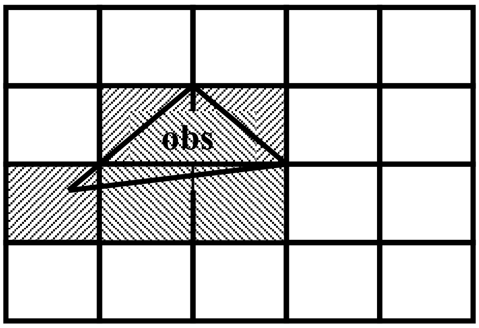


Fig.3 an example of detection area obstacle

Another example is presented in the figure 4 to find an optimal path to navigate intelligibly avoiding the obstacles. This example shows the way on which the scene of navigation is decomposed.

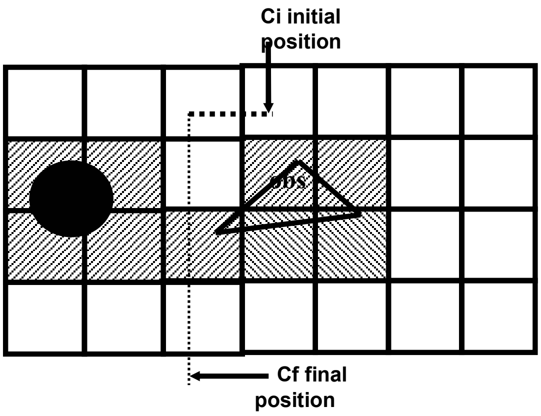


Fig. 4 example of the navigation finding an optimal path

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.

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 primary role in any problem solution in the domain of mobile robots because it is essential that the mobile robot has the ability to build and use models of its environment that enable it to understand the scene navigation's structure. This is necessary to understand orders, plan and execute paths.

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.

B. Delphi Language

To design a software program, two elements must be taken into consideration: the first one is the structure of the program i.e. the flowchart, and the second is the language in which the program will be written.

The Borland Delphi environment is a fine development environment, which is widely used by computing professionals throughout the world. Delphi programming permits to develop the applications for Windows. It is a tool that makes a visual conception for functions, to the programming object. Besides, it maintains the part of the source code automatically. This language offers to the user a friendly interface to the program. All the common parts of the Windows graphical user interface, like forms, buttons and lists objects, are included in Delphi as components. This means that it is not needed to write any code when adding them to an application. Delphi has many advantages comparing to other languages that work under DOS. As Windows is a multitasking operating system, many applications may run at the same time without affecting the Delphi programming environment, this is considered as an ideal software in our project that needs simultaneous operations of reading, processing and displaying, we have chosen Delphi language for this process because of many advantages and applications it offers to the navigation planning.

C. Visual Basic language

A programming language and environment developed by Microsoft. Based on the BASIC language, Visual Basic was one of the first products to provide a graphical programming environment and a paint metaphor for developing user interfaces. Instead of worrying about syntax details, the Visual Basic programmer can add a substantial amount of code simply by dragging and dropping *controls*, such as buttons and dialog boxes, and then defining their appearance and behaviour. Although not a true object-oriented programming language in the strictest sense, Visual Basic nevertheless has an object-oriented philosophy. It is sometimes called an *event-driven* language because each object can react to different events such as a mouse click.

Since its launch in 1990, the Visual Basic approach has become the norm for programming languages. Now there are visual environments for many programming languages, including C, C++, Pascal, and Java. Visual Basic is sometimes called a Rapid Application Development (RAD) system because it enables programmers to quickly build prototype applications.

IV. Simulation Results

The algorithms are implemented in Borland C++, afterwards tested with visual basic and DELPHI programming language; whereby the environment is studied in a two dimensional coordinate system. The simulation part is an approach to the real expected result; this part is done using C++ to recognize all objects within the environment and since it is suitable for graphic problems. Taking the segmented environment issued from C++ development, the algorithm permit the robot to move from the initial position to the desired position following an estimated trajectory using visual basic and Delphi language.

A. General Flowchart

Our general flowchart is presented in the figure 5, where the main work is described in order to get the target.

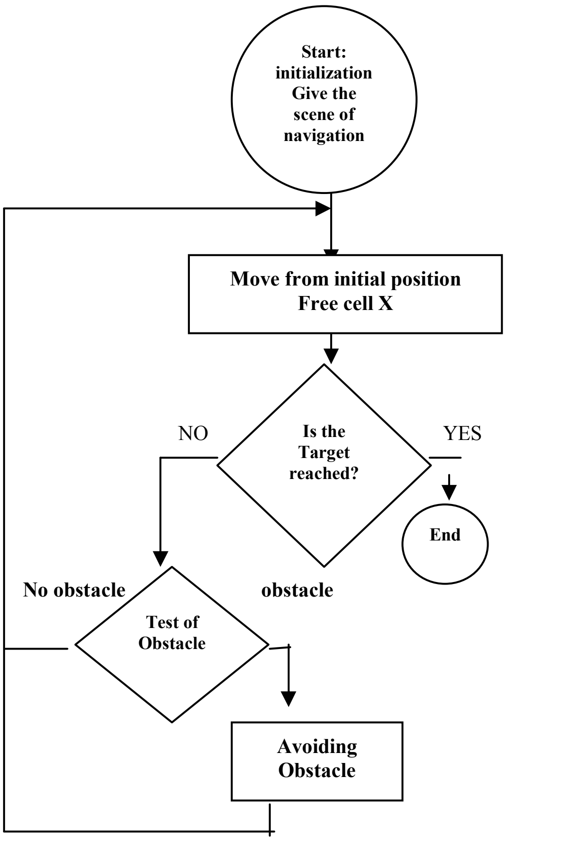


Fig.5 the general flowchart of the process of navigation of IAS

B. First implementation of navigation approach using C++

To reflect the vehicle behaviors acquired by learning and to demonstrate generalization and adaptation abilities of our approach, the robot is simulated in different static environments. In this context, we have created N unknown environments containing static obstacles; (complexity order of theses creations is limited at the last environment one, until now we have tested 56 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.

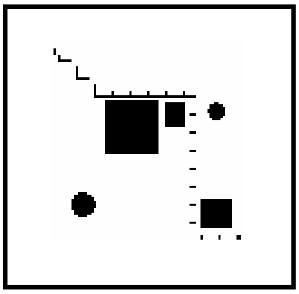
Tested in different unknown environments with static obstacles, we present simulation results which provide the most preferable path between another one treated. As it is illustrated In Figure 6.a where S: Robot and B: Target, the vehicle succeeds to avoid obstacles and reaches its target. In this case, 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, it is replaced by point material and the path is a set of positions of all points of navigation.

The shortest /optimal path is essential for the efficient operation of mobile robot. For any starting point within the environment representing the initial position of the mobile robot, the shortest path to the goal is traced by walking, avoiding obstacles, taking a correct decision, recognizing and the best reasoning. However, ambiguity of optimal paths exists where there exist two or more cells to choose the same least distance transform.

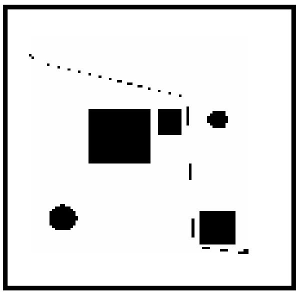
The user can change the shape (body) of robot to execute the final path by gravity center (but the size of the vehicle is taken into account). We replace the body of vehicle by gravity center (material point) to execute the path truly. Before, the optimum path has been calculated and the accurate avoidance direction is known (Figure6.a), so now the robot knows at advance how to evolve and where is situated from the target (Figure6.b). The final decision is taken and the best path to execute is selected, the robot can evolve without risk. These results display the approach ability making IAS able to intelligently avoid obstacles with different architectures. In the figure 7 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 (Figure7.b). The final decision is taken and the best path to execute is selected, the robot can evolve without risk.

C. Using Visual Basic

In the beginning, we have implemented a simulation program using visual Basic language. The environment set up is shown in the figure 6.The squares are small enough to permit the robot land in the next square horizontal at just one step of robot. The path is found by figuring all the squares. Once the path is found, the robot moves from one square to the next until the target is reached, once we have simplified our search area into a convenient number of nodes, as we have done with the grid design, the next step is to conduct a search to find the path. We do this by starting at point A, checking the adjacent squares, and generally searching outward until we find our target. We start the search by the following steps: we have fixed the starting position, it moves forward horizontally as shown above in figure 8.The robot meets an obstacle; it moves a step down then back until it meets another obstacle. The robot keeps navigation in this manner until the target is found, as shown in figure 9and figure 10, shown the robot

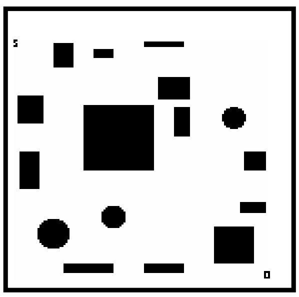


a. The reached best path

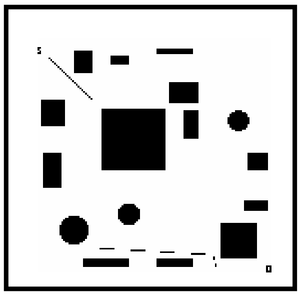


b. The final decision to be taken to execute the best path

Fig.6 an example of simple environment of navigation of an IAS



a .The reached best path



b. The final decision to be taken to execute the best path

Fig. 7 an example of complex environment of navigation of an IAS

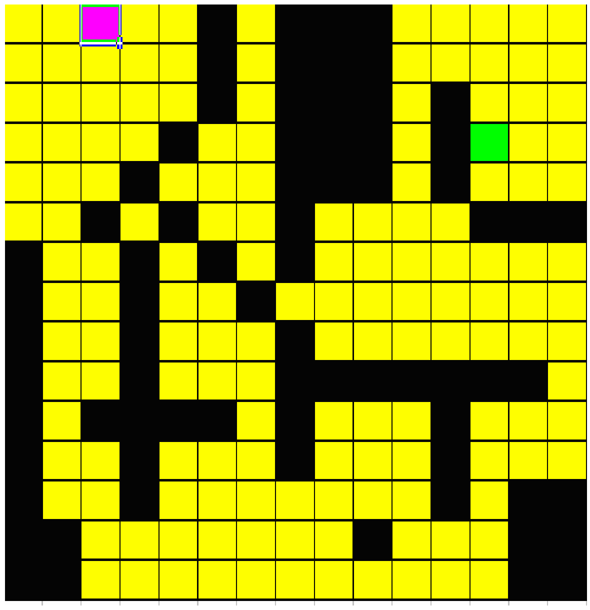


Fig.8 Assumed initial environment set up

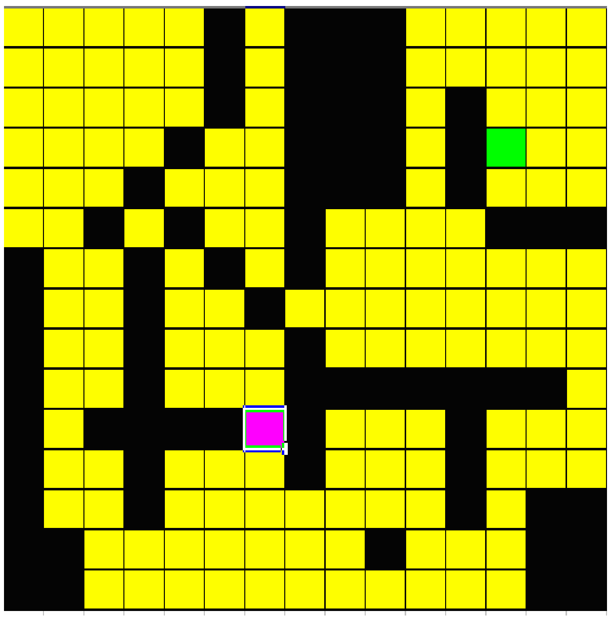


Fig.9 Intermediate condition

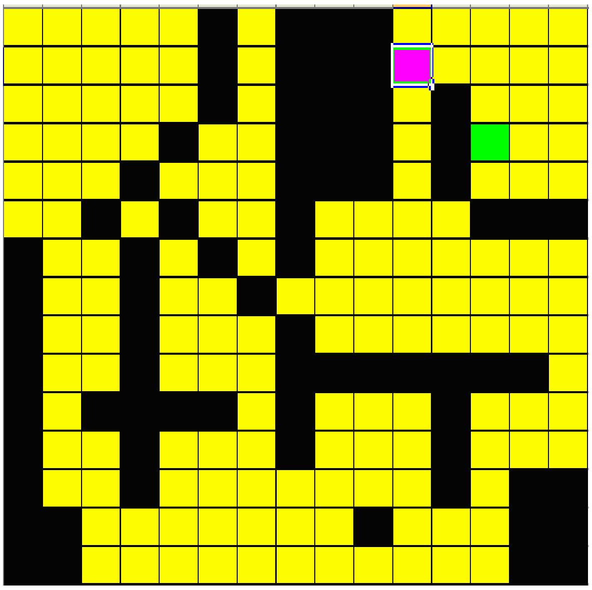


Fig.10 The robot approaches the target

D. using Borland Delphi

Dealing with more principal and more details of navigation of autonomous mobile robot, we write a program using DELPHI version 7 that allows the robot to take the shortest path. We used Delphi 7 to clarify and see how we can approach real expected results. The path is found by figuring out the least number of squares it can take to reach the target. As soon as the path is found, the robot moves from one square to next, when the robot moves for number of squares it meets an obstacle then, it changes the direction to avoid the obstacle but takes the shortest path until the target is reached as shown in the following figures :

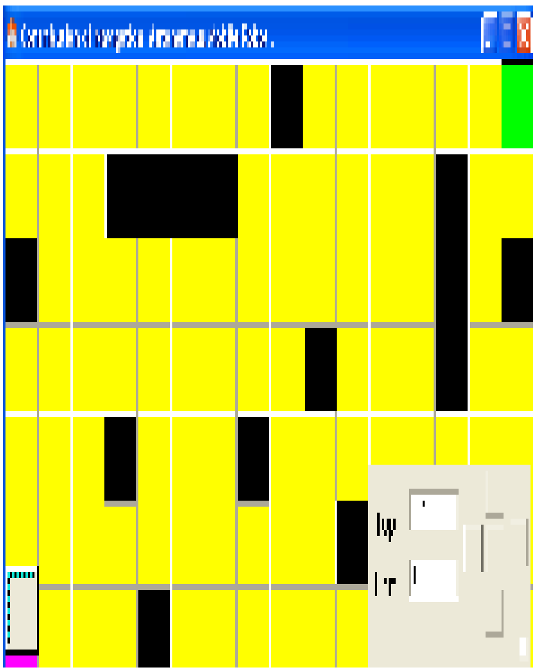


Fig.11 Initial condition set up

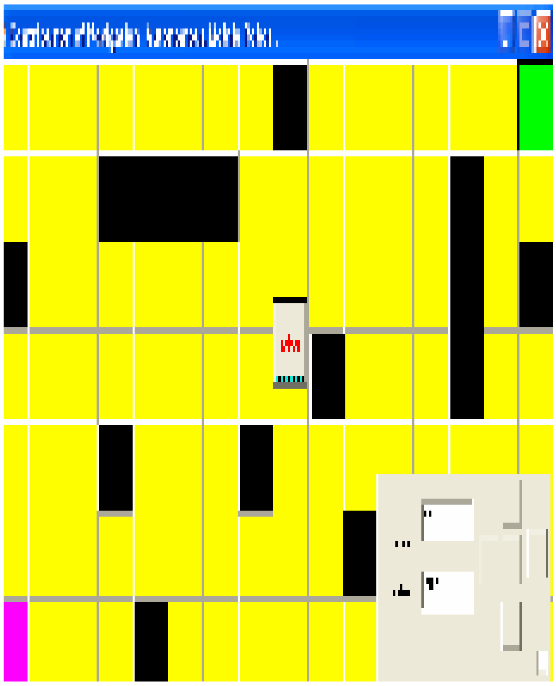


Fig. 12 Intermediate condition

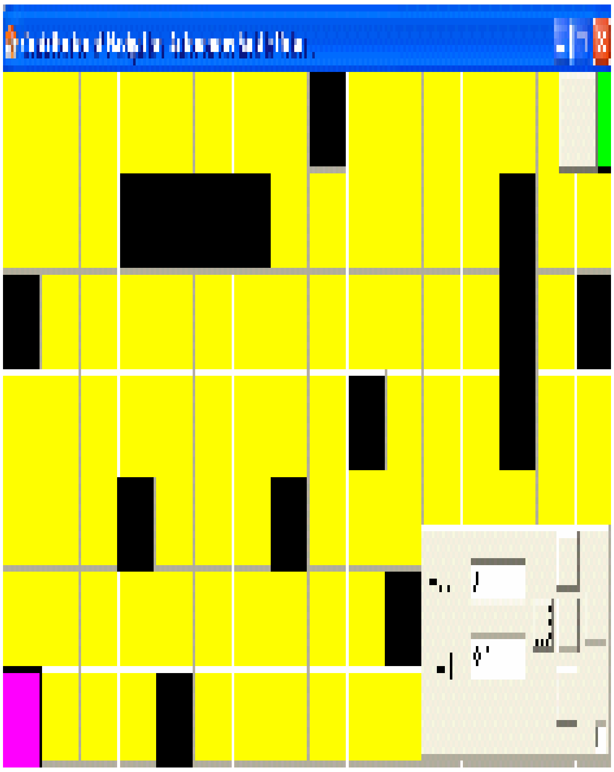


Fig.13 Final condition showing the robot reaching the target

V. Conclusion

The theory and practice of IAS 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 hardware implementation of navigation approach of an autonomous mobile robot in an unknown environment using hybrid intelligent. Indeed, the main feature of 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 our *approach* allowing recognizing situation of the target localization and obstacle avoidance, respectively. Also, the aim work has demonstrated the basic features of navigation of an autonomous mobile robot simulation using visual Basic and DELPHI programming languages. We have run our simulation using the two programming languages: in the basic programming language the robot reaches the target by avoiding obstacles regardless of the number of squares that it takes but in Delphi the robot takes the shortest path to reach the target. The proposed approach can deal a wide number of environments. This navigation approach has an advantage of adaptivity such that the AMR approach works perfectly even if an environment is unknown. This proposed approach has made the robot able to achieve these tasks : avoid obstacles, deciding, perception, recognition and to attend the target which are the main factors to be realized of autonomy requirements. Hence; the results are promising for next future work of this domain.

References

[1] B.P. Gerkey, M.J Mataric, Principled Communication for Dynamic Multi-
Robot Task Allocation, *Experimental Robotics VII, LNCIS 271,* Springer
Verlag Berlin Heidelberg, 2001,pp.353-362.

[2] S Saripalli, G. S. Sukhatme, and J.F.Montgomery, an Experimental Study Of The Autonomous Helicopter Landing Problem, *the Eight International Symposium on Experimental Robotics*, July 2002,pp. 8-11.

[3] B. P. Gerkey , M.J MatariĆ, and G.S.Sukhatme, Exploiting Physical Dynamics for concurrent control of a mobile robot, *In Proceeding of the IEEE International Conference on Robotics and Automation (ICRA 2002)*, Washington, DC , May 2002, pp. 3467-3472.

[4] D. Estrin, D. Culler, K. Pister, PERVASIVE Computing IEEE, 2002,pp. 59-69.

[5] T. Willeke, C. Kunz, 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.

[6] A. Howard, M.J MatariĆ and G.S.Sukhatme, An Incremental Self-Deployment Algorithm for mobile Sensor Networks, autnomus robots, special Issue on Intelligent Embedded Systems, 13(2), September 202, pp. 113-126.

[7] L. Moreno, E.A Puente, and M.A.Salichs, : World modeling 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.

[8] K. Schilling and C. Jungius : Mobile robots for planetary explorations, *in Proceeding second international conference IFAC*, Intelligent Autonomous

[9] S. Florczyk, *Robot Vision Video-based Indoor Exploration with Autonomous and Mobile Robots*, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2005.

[10] O. Hachour and N. Mastorakis, IAV : A VHDL methodology for FPGA implementation, *WSEAS transaction on circuits and systems,* Issue5, Volume3,ISSN 1109-2734, pp.1091-1096.

[11] O. Hachour AND N. Mastorakis, FPGA implementation of navigation approach, *WSEAS international multiconference 4th WSEAS robotics, distance learning and intelligent communication systems (ICRODIC 2004),* in Rio de Janeiro Brazil, October 1-15 , 2004, pp2777.

[12] O. Hachour AND N. Mastorakis, Avoiding obstacles using FPGA –a new solution and application *,5th 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 .

[13] O. Hachour AND N. Mastorakis Behaviour of intelligent autonomous ROBOTIC IAR", *IASME transaction*, issue1, volume 1 ISSN 1790-031x WSEAS January 2004,pp 76-86.

[14] O. Hachour AND N. Mastorakism Intelligent Control and planning of IAR, 3rd WSEAS International Multiconfrence on System Science and engineering, in Copacabana Rio De Janeiro, Brazil, October 12-15,2004.www.wseas.org.