The proposed path finding strategy in static unknown environments

O. Hachour

***Abstract*—** In this present work we present a strategy of navigation of autonomous mobile robots. The proposed path finding strategy is designed in a grid-map forming static unknown environment. We propose an algorithm that provides the robot the ability to move from the initial position to the specified target. First, the 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. We propose scheme for path finding from initial position to goal position. The problem can basically be divided into positioning and path planning. Navigation is a major challenge for autonomous mobile robots. Starting out from start position in the grid, the mobile robot can autonomously head for destination position in the grid to attend the target.The robot moves within the environment by sensing and avoiding the obstacles coming across its way towards the target. The proposed path finding provides the mobile robot more autonomy and intelligence. The proposed method improves the machine intelligence to bring the machine behaviour near the human one in thinking, reasoning and acting. The objective of intelligent mobile robots is to improve machine autonomy. 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 simulation part is an approach to the real expected result. The algorithms are implemented in visual basic programming language; whereby the environment is studied in a two dimensional coordinate system .The algorithm permits the robot to move from the initial position to the desired position following an estimated trajectory.

***Keywords*—** Autonomous Mobile robots, Navigation, Path Finding.

I. INTRODUCTION

THe use of autonomous robots can provide significant benefits in many fields. Robots can be used to reduce the risk involved with human physical intervention, especially in hazardous environments. They can approach the locations of interest to report sensory data and to show more detailed views of a suspicious area. Moreover they can perform long-time tedious tasks that require reliable execution, without lowering their level of efficiency. In recent years there has been a number of projects dealing with the problems involved in the use of autonomous robot to enhance a surveillance system.

A key prerequisite for a truly autonomous robot is that it can navigate safely within its environment. The problem of achieving this is one of the most active areas in mobile robotics research, which is stated as finding the answers to the three questions ''where am I?'', ''where do I go?'', and ''how do I get there?''. For an autonomous mobile robot these questions refer to the tasks of self-localization, map building, and path planning. The difficulty of this problem depends on the characteristics of the robot's environment, the characteristics of its sensors, and the map representation required by the application at the same time.

The autonomous robot navigation problem has been studied thoroughly by the robotics research community over the last years. The basic feature of an autonomous mobile robot is its capability to operate independently in unknown or partially known environments. The autonomy implies that the robot is capable of reacting to static obstacles and unpredictable dynamic events that may impede the successful execution of a task. To achieve this level of robustness, methods need to be developed to provide solutions to localization, map building, planning and control. The robot has to find a collision-free trajectory between the starting configuration and the goal configuration in a static or dynamic environment containing some obstacles.

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 [1,3,6,7].

The robot has to find a collision-free trajectory between the starting configuration and the goal configuration in a static or dynamic environment containing some obstacles. To this end, the robot needs the capability to build a map of the environment, which is essentially a repetitive process of moving to a new position, sensing the environment, updating the map, and planning subsequent motion. Most of the difficulties in this process originate in the nature of the real world: unstructured environments and inherent large uncertainties. First, any prior knowledge about the environment is, in general, incomplete, uncertain, and approximate [4]. For example, maps typically omit some details and temporary features; also, spatial relations between objects may have changed since the map was built. Second, perceptually acquired information is usually unreliable. Third, a real-world environment typically has complex and unpredictable dynamics: objects can move, other agents can modify the environment, and apparently stable features may change with time. Finally, the effects of control actions are not completely reliable, e.g. the wheels of a mobile robot may slip, resulting in accumulated zoometric errors.

Robot navigation can be defined as the combination of three basic activities:

• Map building: this is the process of constructing a map from sensor readings taken at different robot locations. The correct treatment of sensor data and the reliable localization of the robot are fundamental in the map-building process.

• Localization: this is the process of getting the actual robot's location from sensor readings and the most recent map. An accurate map and reliable sensors are crucial to achieving good localization.

• Path planning: This is the process of generating a feasible and safe trajectory from the current robot location to a goal based on the current map. In this case, it is also very important to have an accurate map and a reliable localization.

Recent research on intelligent autonomous robot 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 .

The path planning problem is in its most general form a geometric problem. It needs four ingredients:

- A description of the geometry of the moving entity (this is called the *robot*).
- A description of (the geometry of) the environment in which the robot moves or operates (also called the *workspace*)
- A description of the degrees of freedom of the robot's motion
- A start and a goal configuration in the environment, between which a path is to be planned for the robot.

Using this information, we can construct the *configuration space* of the robot, in terms of which the path planning problem is formulated generally. A *configuration* of the robot in its workspace is described using a number of parameters. A *configuration* of a manipulator robot can be described using a number of angles that the joints make between the consecutive links. The minimal number of parameters needed to uniquely describe a robot configuration equals the number of *degrees of freedom*(dofs) of the robot. The space of all robot configurations is called the *configuration space* of the robot.

Each degree of freedom of the robot accounts for one dimension of the configuration space. In realistic scenarios, the workspace of the robot contains obstacles. These obstacles cause some configurations to be *forbidden*. For instance, a configuration *C* is forbidden if the robot configured at *C* collides with any of the obstacles in the workspace. More generally, the configuration space *C* is partitioned into a set of forbidden configurations *Cforb*, and a set of free configurations *Cfree*. Configurations in *Cforb* not necessarily relate to collisions with workspace obstacles. For instance, if the robot is an industrial manipulator arm, the end-effector may collide with the base of the robot in some configurations. These are called *self-collisions*. In general, forbidden configurations can be the result of all kinds of internal robot constraints.

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 modelled geometrically and are controllable via a motion plan. A 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. Hence, 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 the mobile robots because it is essential that the mobile robots have 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.

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, behaviour-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.

To detect all possible obstacles, the robot is supposed to have vision system (camera). To operate in certain dynamic environments, the use of two or more sensors can guarantee to deliver acceptably accurate information all of he time. Thus the redundancy can be useful for autonomous systems as in the human sensory system

When an autonomous robot moves from a point to a target point in it given environment it is necessary to plan an optimal or feasible path avoiding obstruction in its way and answering to autonomy requirements such as: thermal, energy, Communication Management, Mechanical design, etc.

To evaluate the performances of vehicles one must answer to all factors to be embedded with robot when it executes its mission, this is summarized in how to perform all tasks, such as intelligence and autonomy requirements.

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 a researcher 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.

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.

The control task becomes more complex when the configuration of obstacles is not known a priori. The most popular control methods for such systems are based on reactive local navigation schemes that tightly couple the robot actions to the sensor information.

The multi-level structure of path planning and execution propounded in provides a basic framework for dealing with problems in the control of autonomous vehicles. Traditionally, motion planning and control have been separate fields within robotics. However, this historical distinction is at best arbitrary and at worst harmful to the development of practically successful algorithms for generating robotic motion. It is more useful to see planning and control as existing on the same continuum.

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. Many traditional working machines already used are going through changes to become remotely operated or even autonomous. Technology has made this feasible by using advanced computer control systems. The theory of planetary explorations includes vehicles as an infrastructure for mobility on the surface. Their use enables the transport of instruments for measurements and sampling to multiple locations of scientific interest like rocks and crevasses. In particular the robot science has shown the capabilities of robots including: power consumption, *autonomous reaction capabilities*, design complexity, learning, own decision, mission flexibilities, performing missing and reacting are done in efficient manner to deal with anthropomorphist principle. This is was a dream in all Artificial Intelligence world realizations.

When the robot is far from its objective only the mobile base moves; thus avoiding obstacles if necessary. When the objective is close to the robot, both mobile base and arm move and redundancy can be used to maximise a manipulability criterion. The partial results obtained with the real robot consolidate the results of simulation. The work does not propose an autonomous path planning and navigation of the mobile arm but assistance to the user for remote controlling it.

In this present paper, we present a strategy of navigation of autonomous mobile robot. It is tested to navigate an unknown environment. The environment of navigation can be changed by user demand that means that the robot can move in another environment where a given shape is designed by the user: a square and rectangle.

A Robot is asked to navigate an unknown environment which can be changed. It is placed at a beginning position (the starting position) in an unknown environment and is asked to try to attend target position. Positions in the environments either will be open or blocked with an obstacle, it depends on the choice of the environments.

The proposed path finding strategy is designed in a grid-map form. The aim of this work is to develop an algorithm which allows a mobile robot to navigate through static obstacles, and finding the path in order to reach a specified target. We propose an algorithm that provides the robot a trajectory to be followed to move from the initial position to the specified target. The robot trajectory is designed in a grid-map form of an unknown environment with static obstacles. We propose scheme for path finding from initial position to goal position. The problem can basically be divided into positioning and path planning.

Navigation is a major challenge for autonomous mobile robots. Starting out from a predefined position and orientation in the grid, the mobile robot can autonomously head for destination position in the grid. The proposed path finding strategy is designed in a grid-map form of an environment with static unknown obstacles.

The robot moves within the 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 simulation part is an approach to the real expected results where the algorithm is implemented in visual basic programming language; whereby the environment is studied in a two dimensional coordinates system. The algorithms permit the robot to move from the initial position to the desired position following an estimated trajectory.

II. Path planning

Planning is a term that means different things to different groups of people. Robotics addresses the automation of mechanical systems that have sensing, actuation, and computation capabilities (similar terms, such as autonomous systems are also used). A fundamental need in robotics is to have algorithms that convert high-level specifications of tasks from humans into low-level descriptions of how to move [2,5].

The terms motion planning and trajectory planning are often used for these kinds of problems. Imagine giving a precise computer-aided design (CAD) model of a factory and a machine as input to an algorithm. The algorithm must determine how to move the machine from one room to another in the factory without hitting anything. Most of us have encountered similar problems when moving a mattress up a set of stairs. Robot motion planning usually ignores dynamics and other differential constraints and focuses primarily on the translations and rotations required to move the machine. Recent work, however, does consider other aspects, such as uncertainties, differential constraints, modelling errors, and optimality.

Trajectory planning usually refers to the problem of taking the solution from a robot motion planning algorithm and determining how to move along the solution in a way that respects the mechanical limitations of the robot. Natural questions at this point are what is a plan? How is a plan represented? How is it computed? What is it supposed to achieve? How is its quality evaluated? Who or what is going to use it? These questions must be answer in the design and the development. Regarding the user of the plan, it clearly depends on the application. Imagine, for example, that the planning algorithm provides you with an investment strategy.

A plan in general, a plan imposes a specific strategy or behaviour on a decision maker. A plan may simply specify a sequence of actions to be taken; however, it could be more complicated. If it is impossible to predict future states, then the
plan can specify actions as a function of state. In this case, regardless of the future states, the appropriate action is determined. Using terminology from other fields, this enables feedback or reactive plans. It might even be the case that the state cannot be measured. In this case, the appropriate action must be determined from whatever information is available up to the current time. This will generally be referred to as an information state, on which the actions of a plan are conditioned.

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.
The multi-level structure of path planning and execution propounded in provides a basic framework for dealing with problems in the control of autonomous vehicles. There are three basic levels of path planning and execution:

A. *The Planner*

A global path planner uses priori knowledge (a map) to plan a plausible road.

B. *The Navigator*

A local path planner, uses the plan of the planner a guide, but provides more precision routing according to obtained terrain information locally.

C. *The Pilot*

It is the execution of simple vehicle movement routines. Path planning is one of the key issues in mobile robot navigation.

Path planning is traditionally divided into two categories: global path planning and local path planning. In global path planning, prior knowledge of the workspace is available which means that the robot navigates in known environment. Local path planning methods use ultrasonic sensors, laser range finders, and on-board vision systems to perceive the environment to perform on-line planning in this case a prior knowledge of the environment is not necessary and the robot navigate in an unknown environment.

III. Autonomous Mobile Robots

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.
A fully autonomous robot has the ability to:

- Gain information about the environment.
- Work for an extended period without human intervention.
- Move either all or part of it self throughout its operating environment without human assistance.

An autonomous robot may also learn or gain new capabilities like adjusting strategies for accomplishing its tasks or adapting to changing surrounding Autonomous robots still require regular maintenance, as do other machines.

Navigation is the science of directing the course of a mobile robot as the robot traverses the Environment. Inherent in any navigation schema is the desire to reach a destination without getting lost or crashing into any objects (see the figure1). In this figure, a map start position of the robot is defined, and a destination is assigned. To reach the destination, the robot will explore the environment, looking at the floor and locating the first obstacles. Once a new obstacle is detected, the robot changes the path to reach the destination.

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 track perfectly. Early path planners were such off-line planners or were only suitable for such off-line planning. However, the limitation 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.

Autonomous robots are robots which can perform desired tasks in unstructured Environment without continuous human guidance. Many kinds of robots can be autonomous. In different ways. A high degree of autonomy is particularly desirable in fields such as space exploration, cleaning floors, moving lawns, and waste water treatment. .Some modern factory robots are "autonomous " within the strict confines of their direct environment .it may not be that every degree of freedom exists in their surrounding environment but the factory robots workplace is challenging and can often contain chaotic ,unpredicted variables. The exact orientation and position of the next object and the required task must be determined. This can vary unpredictably at least from the robots point of view. One important area of robotics research is to enable the robot to cope with its environment whether this be on land, underwater, in the air, underground, or in space for example.

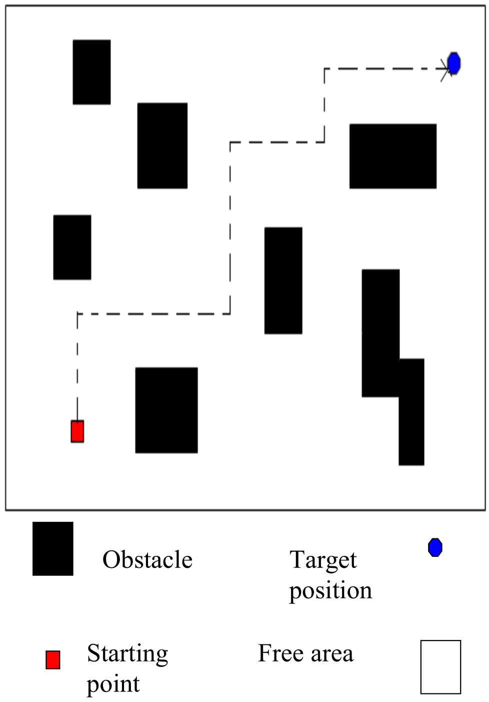

Fig. 1 Robot trajectory

In the example of figure1, a map start position of the robot is defined, and a destination is assigned. To reach the destination, the robot will explore the environment, looking at the floor and locating the first obstacles. Once a new obstacle is detected, the robot changes the path to reach the destination.

IV. Navigation Models of Mobile Robots

Nowadays, Intelligent Autonomous robots can carry out task in various environments by themselves like human. These intelligent autonomous systems are very useful in many fields as industry and planetary explorations. However, they are all semi-autonomous and need some human operators. In fact, the future factory needs all flexible and robust intelligent autonomous mobile robots.

Additionally, Intelligent Autonomous Vehicles (IAV) are becoming more and more interesting for underwater, terrestrial, and space applications. These mechanical systems are constructed to respond any traditional working as in construction and agriculture. Previously, certain industrial operations required human skills may be tedious and exceptionally hardly ever. Above all, repetitive operations can result in reductions in quality control, as in visual inspections tasks. Also, these repetitive actions may be hazardous health risks as exposure to unsafe materials like radioactive and high pressure in underwater applications. So, the presence of human workers in these environments may be perilous which need the necessity to be replaced by intelligent systems, these systems can move, react, and carry out tasks in various environments by themselves like human.

The goal of the navigation system of mobile robots is to move the robot to a named place in known, un known or partially environment in most practical of 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 navigation, when the robot actually starts to travel along a planned path, it may find that:

There are an 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 move is possible, it must plan a new path with the modified map and its current position as the new starting location.

The robot must then be able to understand the structure of the environment to find a way towards its target without collisions. In general and summary of some works, there are two models of navigation according to recent development navigation approach: a grid model and a cellular model. In the first model, the robot environment is divided into many small squares by which each square indicates either the obstacle is present or not. The cellular model is the one where the environment of navigation has been decomposed into cellular areas, some of which include obstacles.

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

1-It provides the necessary information to do path planning.
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

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. 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. 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 the; mobile robots because it is essential that the mobile robots have 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.

One example is presented in the figure 2, where the Range profile is obtained from the frontal side of the robot. The robot moves by tracking the maximum in the range profile. A local maximum appears when the robot reaches location and it perceives a new open area to the left. At this moment of branching the navigation level decides whether to transfer the focus to the new local maximum or to remain with the current one. The robot decides to transfer the focus to the new maximum and proceeds to the left. In the same manner, the decision process is repeated at point where the focus transfers to a new local maximum on the left-hand side, and at point where it stays with the current branch by traveling forwards, and at where it transfers to a new branch to the Left these binary branching decisions generate the trajectory shown, see the figure 2.

Another example of model of navigation of autonomous mobile robots, we present a Maximum-Clearance Roadmaps algorithm that search for clearance path, the shortest path roadmaps looks for the shortest one. This algorithm is also known as the reduced visibility graph in some articles. Let's define a graph G consisting of a set of vertices V and a set of edges E. The vertices of G are reflex vertices (i.e. polygon vertices for which the interior angle in *Cfree* is greater than Π). While edges of G are formed even by Consecutive reflex vertices, i.e. edge between two reflex vertices which are the endpoints of an edge of *Cobs*, or by bitangent edges, i.e. a bitangent line drawn through a pair of reflex vertices which must be mutually visible from each other (see figure 3).

The figure 4 illustrates this method. To solve a query, q_I and q_G are connected to all roadmap vertices that are visible. If each edge is given a cost that corresponds to its path length, then the resulting solution path is the shortest path between q_I and q_G; this is shown in Figure 5.

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, behaviour-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.

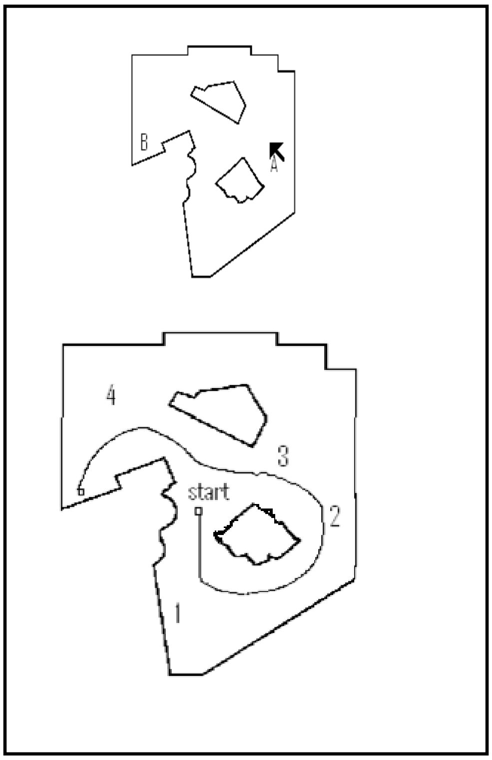

Fig. 2 An example of travel and its associated sensory flow.

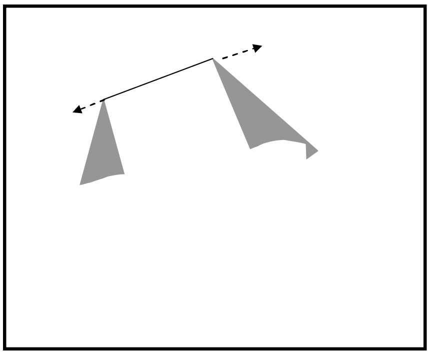

Fig. 3 The search graph and the query solution

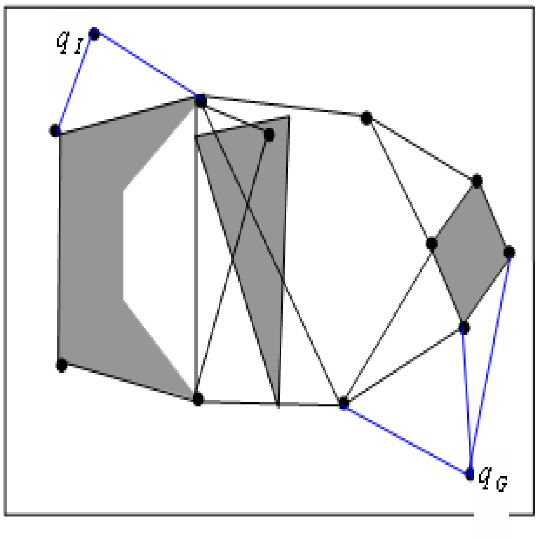

Fig. 4 The search graph and the query solution

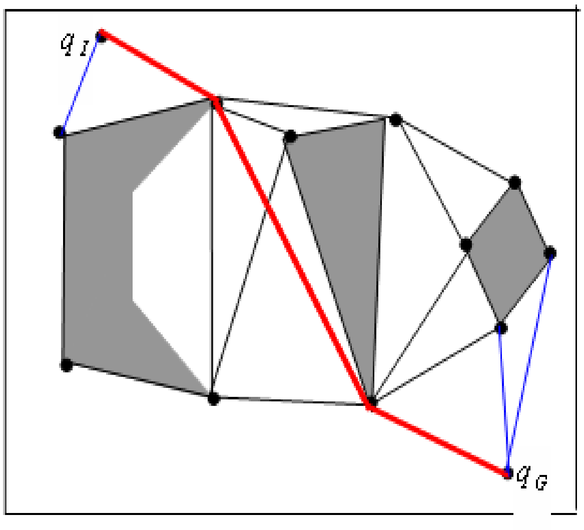

Fig. 5 The shortest path in the extended roadmap

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.

V. THE PROPOSED STRATEGY OF NAVIGATION APPROACH

In this present work, we present a strategy of navigation of autonomous mobile robot. It is tested to navigate an unknown environment. The environment of navigation can be changed by user demand that means that the robot can move in another environment where a given shape is designed by the user: a square and rectangle.

The algorithm searches the short possible path between a set of paths. For this context we present our method of work in detail, but let us starting to present our language programming which is Visual Basic.

Visual Basic programming language and environment were developed by Microsoft. The 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 programming 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.

It is sometimes called an event-driven language because each object can to different events such as a mouse click. Since its launch in 1990, the Visual Basic approach has become the nom for programming language, now there are visual environments for many programming language, 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 application.

This section presents the method of work using the visual basic. Conceptually speaking, it is an algorithm of finding a path on the ground from a starting location to a target location, avoiding the obstacles and minimizing the cost. The concept is to find the low cost for each sub path (short path) taking into account the avoiding principle. For any contribution, the algorithm tries to find the possible path; otherwise it returns a message indicating that no path in the program the message box is path not found. Two set up environments are tested in this simulation where the principle of searching the best path is applied for each one. As we can see the use of map path planning and the definition of the world map are the main elements that can be taken into account in navigation of autonomous mobile robot. We have presented some navigation models that have been camped by research kind as for example: grid and cellular models. All developed models are reaching to find best navigation respecting autonomy requirements such as energy consumption and thermal dissipation.

For both map building, it works on a rectangular workspace, called the search region or map, divided in little squares, called cells. Two possible situations can occur: walkable path or unwalkable path. The map building depends on the use and checking colours that are identified in the environments set up. The robot can identify three colors inside our environment: dark, yellow and green. The dark color is interpreted as an obstacle area; whereas the yellow color represents the free trajectory to attend the given target, and the green color refers to the area target (this is the first part of the main project to be after developed more, i.e., we start by this principle to give after more intelligence to our mobile robot).

For the second environment set up: four colours are assigned in the simulation:

- The red colour designs the robot body.
- The green colour is the target position.
- The dark colour is the obstacles area.
- The white colour designs the free area .

For each set up execution procedure: 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. 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 cells, 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. For each set up: a Robot is asked to navigate intelligibly this environment which can be changed. It is placed at a beginning position (the starting position) and is asked to try to attend target position. Positions in the environment either will be open or blocked with an obstacle, it depends on the choice of the environments. The game of colour is just to check the level of reasoning of the robot, i.e, if the free space (unoccupied area) is yellow or white it has no effect since the robot must recognize the difference luminous intensity levels between the dangerous area and the safety area. The robot must recognize by learning the difference between the free area and the occupied area and decide how to get the free feasible path without collision. The Robot comes only move on the free positions (free area without obstacles) and must stay within the environment searching its way from the starting position to the target position (a solution path) until it finds one or until it exhausts all possibilities (no possible paths).

The figure 6 shows an example of walkable or unwalkable 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±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.

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.

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 either 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.

If the algorithm does not converge, an error is returned in the variable Error Message. If there is no possible way to get the target, the program returns (no way) as message box response indicating that after much computing there is no possible configuration to the target. This is very important in all navigation process, because instead of stopping the program or waiting without issue, the user gets the answer which is logical as answer to show that no way towards the target.

Our General flowchart of work is presented in the figure7.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 9 and figure 10, shown the robot close to the target.

The input parameter Map contains the ground information. It is an (N x M) array. Each element represents a single square of the ground. The user provides the starting cell (Is, Js) and the target cell (It, Jt) by two arrays: Path Start and Path End. The routine searches for the best path and returns the cells of the path, in reverse order, in the Path array; it has two columns [i, j] and several rows, depending to the path-lenght.

In order to evaluate, the performance of navigation algorithm of autonomous mobile robots over various environments, we observed simulation of the navigation in different environments. We can change the position of obstacles so we get other different environments. These environments were randomly generated. To find a new path after insertion of deletion of an obstacle. Hence, a mobile robot detects unknown hazardous obstacle on the path and find its free path without collision. More, after the generation of several paths given by the process of navigation, the robot reaches its target intelligibly by deciding itself how to navigate, how to avoid obstacle, and how to reach carefully his goal. 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.

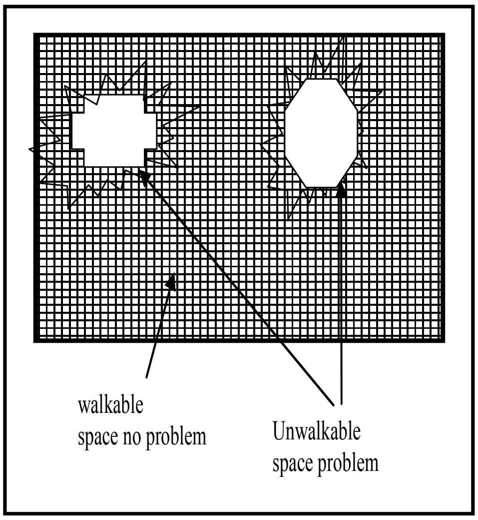

Fig. 6 an example of walkable space and walkable

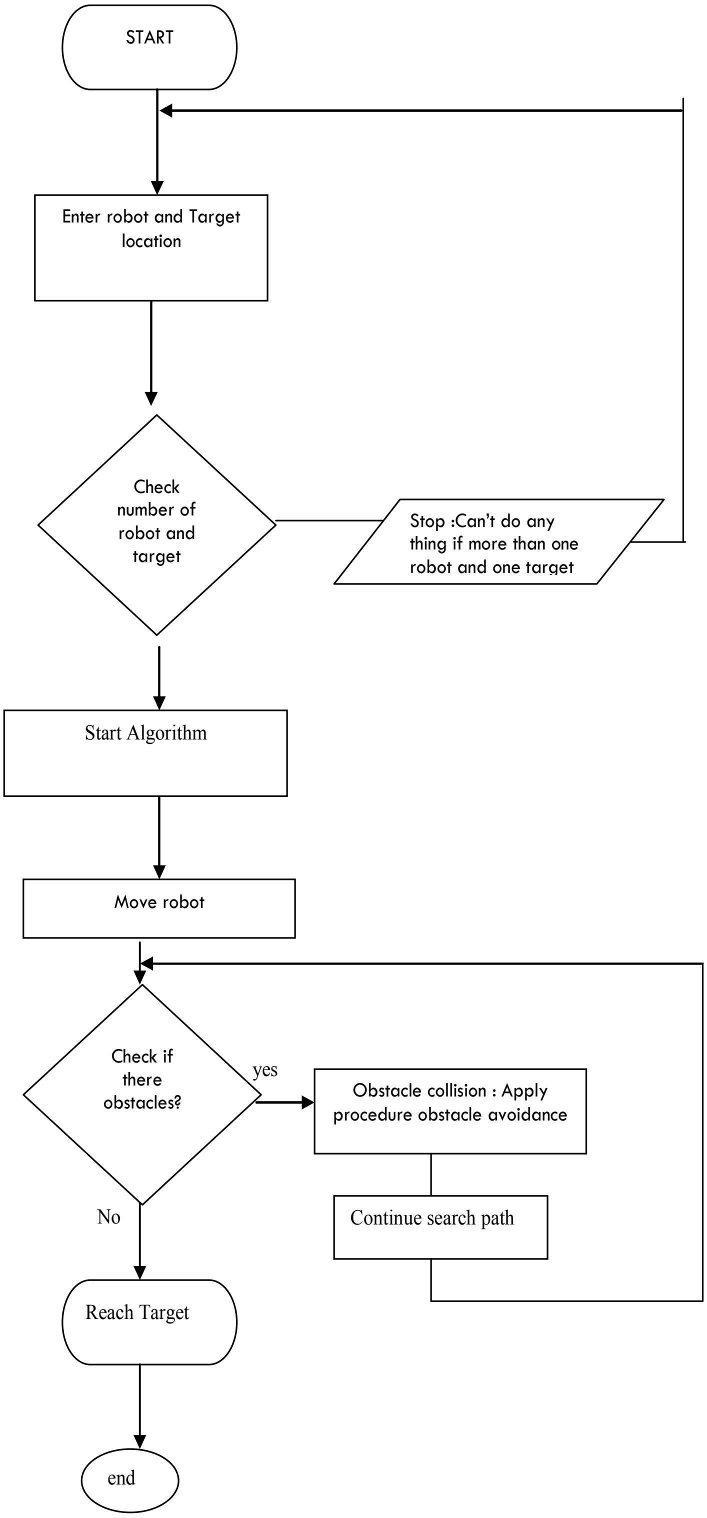

Fig. 7 the general flowchart

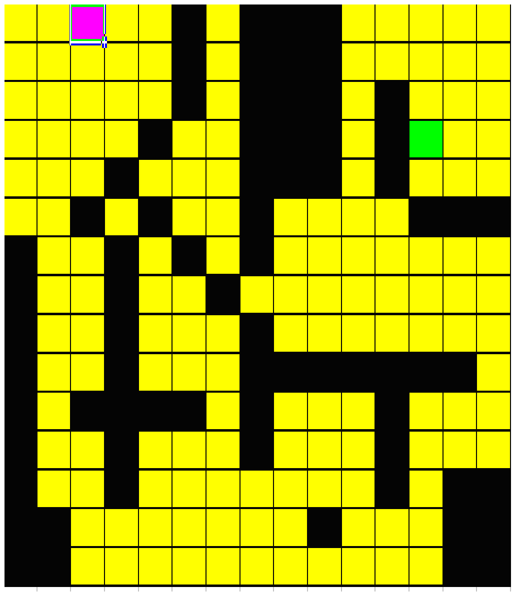

Fig. 8 Assumed initial environment set up

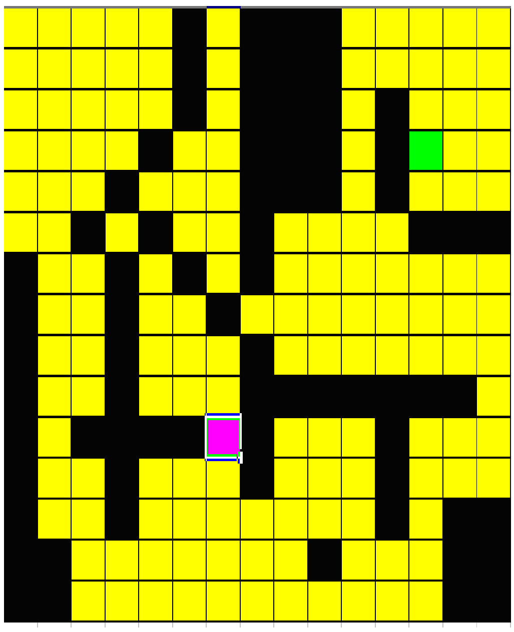

Fig. 9 Intermediate condition

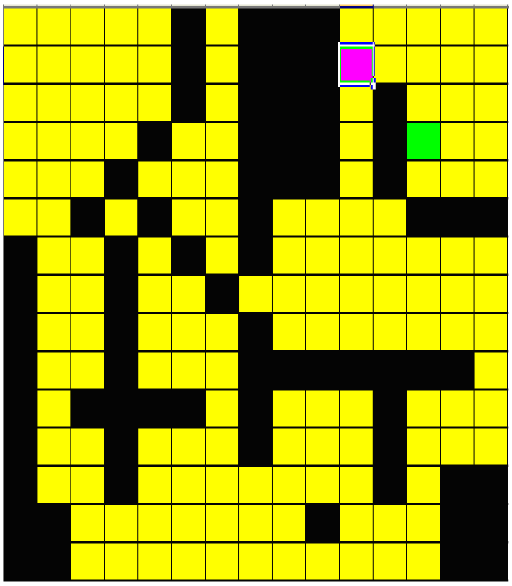

Fig. 10 The robot approaches the target

This navigation approach has an advantage of adaptivity such that the mobile robot algorithm works perfectly even if an environment is unknown. This proposed approach has made the robot able to achieve these tasks: avoid obstacles, deciding, perception, and recognition and to attend the target which are the main factors to be realized of autonomy requirements.

We have run our simulation using Visual Basic language: the robot reaches the target by avoiding obstacle regardless of the number of square it takes. The path is found by figuring out the least number of squares it can take to reach the target. : in the basic programming language the robot reaches the target by avoiding obstacles. Conceptually speaking, our algorithm is quite simple: it is an algorithm of finding a feasible path on the ground from a starting location to a target location, avoiding the obstacles and minimizing the cost. The concept is to find the low cost for each sub path (short path) taking into account the collision principle.

To provide a graphical programming environment and to deal with autonomy requirements: we have executed our algorithm in another environment set up where the robot succeeds to reach its target without collision intelligibly. As extension for the first environment set up execution, we note that this algorithm is flexible and generic, and can be changed by user demand. This is to prove the robustness and adaptivity of this algorithm. We 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 (see the figure 11)

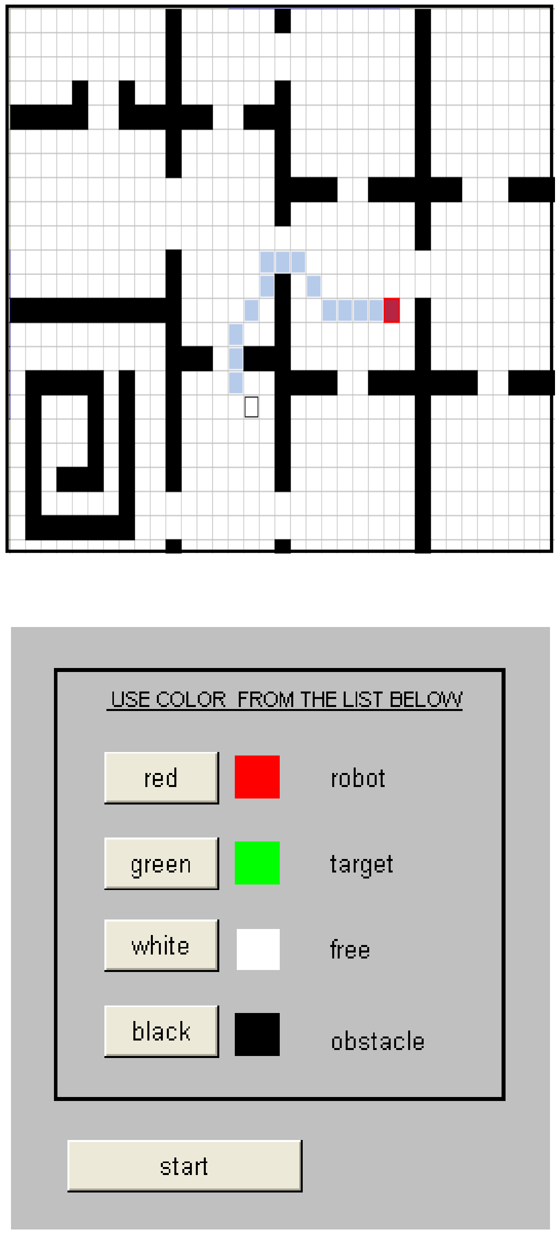

Fig. 11 The robot approaches the target

Another environment set up

VI. Conclusion

In this present work we have studied the problem of path planning in a 2-dimentional surface with obstacles avoidance. A complete path planning algorithm guarantees that the robot can reach the target if possible, or returns a message that indicates that there is no free path when the target cannot be reached. The goal of the navigation system of mobile robot is to move the robot to a specified destination in unknown static environment; in most practical situation the mobile robot can not take a direct path from the start to the goal point, therefore path finding techniques must be used . When the robot starts to travel along a planned path, it may find that there are unexpected obstacles along the path that were not on the map. When this happens, the robot must chart the obstacle and, if no local avoidance move is possible, it must plan a new bath with the modified map and its current position as the new starting location. The obtained path is the shortest path from all possible free trajectories. The proposed path finding strategy has the advantage of being generic and can be changed at the user demand. The obstacles can take any shape since the algorithm is general for any obstacle detection. This approach works perfectly even if an environment is unknown. Hence, the robot is able to understand the structure of the environment to find a way towards its target without collisions.

We have run our simulation in several environments where the robot succeeds to reach its target in each situation and avoids the obstacles capturing the behaviour of intelligent expert system. Our autonomous mobile robot is able to achieve these tasks: avoid obstacles, taking a decision, perception, and recognition and to attend the target which are the main factors to be realized of autonomy requirements. This navigation approach has an advantage of adaptivity such that the approach works perfectly even if an environment is unknown. Hence; the results are promising for next future work of this domain.

References

[1] 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.

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

[3] O. Hachour and N. Mastorakis, IAV : A VHDL methodology for FPGA implementation, *WSEAS transaction on circuits and systems,* Issue5, Vol 3,ISSN 1109-2734, 2004,pp.1091-1096.

[4] V. J. Lumelsky, "Effect of kinematics on motion planning for planar robots arms moving amidst unknown obstacles", *IEEE J. of Robotics and Automation*, RA-3(3), 1987,pp207-223.

[5] S. M. LaValle, "*Planning Algorithms*", Published by Cambridge University Press, 2006.

[6] O.Hachour, "The Proposed Genetic FPGA Implementation For Path Planning of Autonomous Mobile Robot", *International Journal of Circuits , Systems and Signal Processing,* Issue 2, vol2 ,2008,pp151-167.

[7] 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 ,vol 3 , Venice , Italy , , ISSN 1109-2777, November 2004, pp2827-2834.