The proposed Grid _Based Navigation Approach

Hachour Ouarda

Abstract**—** Navigation is a major challenge for autonomous, mobile robots. The problem can basically be divided into positioning and path planning. In this paper we present a scheme for path finding, we focus on positioning. Starting out from an initial position in the grid, the mobile robot can autonomously head for destination cells in the grid. On its way it determines the current location in the grid using a *connectivity_cell* principle by picking up line-Crossing cells. This principle will be clarified in detail. A key ability needed by an autonomous, mobile robot is the possibility to navigate through the space. The focus is on the ability to move and on being self sufficient. The robot navigates on a grid which regularly divides the ground into rectangular cells. To carry out tasks in various environments as in space applications, the robot succeeds to reach its target without collisions. The proposed approach can deal a wide number of environments. This navigation approach has an advantage of adaptivity such that the intelligent autonomous mobile robot approach works perfectly even if an environment is unknown. The results are promising for next future work of this domain

Keywords**—** Intelligent Autonomous Systems, grid workspace, path planning, Obstacle avoiding, intelligence.

I. Introduction

THE problem of the path planning has been studied extensively over the last decades [3]. Most great research application efforts have been spent on path planning in *static* environments. That is, a path has to be found between two configurations for a movable object in an environment containing stationary obstacles whose geometry and coordinates are given. Whereas less attention has been given for *dynamic* environments. Besides stationary obstacles, dynamic environments contain moving obstacles with which collisions must be avoided as well. As an example, a mobile robot operating at a factory floor will have to navigate among humans or other robots, which can be considered as moving obstacles. In general, we can clarify the path planning problem is in its most general form a geometric problem which is based on the following steps:

- A description of the geometry of the robot.
- A description of the geometry of the environment or *workspace* in which the robot moves or operates.
- A description of the degrees of freedom of the robot's motion.
- An initial and a target configuration in the environment, between which a path is to be planned for the robot.

Safe manoeuvring of Autonomous Ground Vehicles (AGVs)

In unstructured complex environments, densely cluttered with obstacles is still a major challenge in goal-directed robotic vehicle applications. Navigation through a forest, which attracts special interest from the military community due to the lack of stealth and concealment found in open environments, is typical of such challenges. This navigation problem is a multi objective control problem that seeks to ensure that the robot not only reaches its goal without hitting obstacles, but also does so at safe speeds that ensures stability. The problem is particularly difficult because some of the navigational objectives may be in opposition to one another.

It is important that algorithms for navigation control in cluttered environments not be too computationally expensive as this would result in a sluggish response. It has been acknowledged that the traditional Plan-Sense-Model-Act approaches are not effective in such environments; instead, local navigation strategies that tightly couple the sensor information to the control actions must be used for the robot to successfully achieve its mission.

With recent success coming from both planetary exploration rovers and ground robots, interest has increased in autonomous navigation. Autonomous ground robot navigation presents a number of difficult problems, which are unsolved as of this time. The largest and most difficult problem concerns how to create a model representing the area in-front of the robot. Currently this problem is unsolved except for very constrained environments. As more successful autonomous navigation systems are developed, more insight is gained into how to incorporate higher levels of knowledge [1,2,3,4,5] .

Many works on this topic have been carried out for the path planning of autonomous mobile robot. Because perfect information concerning the moving obstacles in the environment may not be available, it is important that partial information is adequately coped with. There are a number of existing methods for dealing with this scenario. In particular, we can estimate future trajectories of the obstacles based on current behavior, or we can assume worst-case trajectories. Whichever of these we choose; we end up with some trajectory or set of trajectories that we can represent as objects in the configuration-time space.

We can then avoid these objects as we plan a path for the robot. Planning a least-cost path through this space can be computationally expensive, and although there may be time to generate the robot's initial path, if the robot is continuously receiving new information as it moves, replanning least-cost paths over and over again from scratch may be infeasible. In order to do this, we need to plan in a backwards direction from the goal to the start, so that when the robot moves, the stored paths and path costs of all the states in the search space that have already been computed are not affected1. Since we don't know in advance at what time the goal will be reached, we seed the search with multiple goal states.

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[6,7,8,9] .

Path planning plays an important role in various fields of application and research, among which are CAD-design, computer games and virtual environments, molecular biology, and robotics. In its most general form, we can say that the main work of this level is to plan a feasible path for some moving mass between a start position and a goal position in some environment. A more challenging path planning problem occurs when the set of all possible states is not discrete as in the case of a grid, but continuous. To clarify more the idea, an industrial manipulator robot that has to move in a three-dimensional environment while avoiding collisions with itself and obstacles in the environment [10, 11, 12, 13]. The challenge in these cases is to discretize the problem in a sensible way, such that it becomes tractable.

Using these informations, we can construct the *configuration space* of the robot, in terms of which the path planning problem is formulated generally.

A configuration *or the workspace o*f the robot is described using a number of parameters. a configuration of a robot Translating in a two-dimensional workspace can be described using two parameters, which are often denoted *x* and *y*. or In 3D x, y, z or the angle Θ can also viewed.

The simplest instance of the path planning problem is finding a path for a *point* robot in a two-dimensional static environment. In most cases, it is assumed that the geometry of the workspace obstacles is given using a polygonal representation. As the robot is considered as material point, the configuration space exactly resembles the workspace. Hence, the obstacles in configuration space are *explicitly* represented. This can be used to efficiently solve the planning problem.

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 [13].

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 [14, 15, 16, 17].

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.

Many researchers have addressed this problem. Many authors have considered a model with complete information, where the robot has perfect knowledge about the obstacles. The drawback of these approaches is that under many practical circumstances robot does not have access to complete information about the environment.

This paper deals with an algorithm for two dimensional (2D) path planning to a target for mobile robot in unknown environment. The objective is to find a collision free path from an unknown initial position to an unknown target point. A complete path planning algorithm should guarantee that the robot can reach the target if possible, or prove that the target can not be reached. A few path planning algorithms are described here followed by the aim work of research in detail.

Our autonomous mobile robot is able to achieve these tasks: avoiding obstacles, taking a suitable decision, and attending the target which are the main factors to be realized of autonomy requirements. The algorithm returns the best response of any entering map parameters. The key idea is around the main line from the source to the destination and the m^{th} obstacle causing the collision where they construct the feasible path (a set of non linear segments) . 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 2D 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.

The aim of this paper is to develop an IAS algorithm for the 2D 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). 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 just to trace a path from the source position to the target position looking the 2D obstacles causing the collision. The main work is clarified by the following sections.

II. The proposed approach

The theory and practice of Intelligent Autonomous Systems 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.

The environment in which the robot is operating is a 2-dimensional manifold. It has one starting point *S* and one target point *T*. It also has a finite number of static obstacles. We assume that the boundaries of the obstacles are smooth curves. The robot is considered to be a point. It knows the coordinates of its current position C and the target T. The robot can measure the distance to the closest obstacles which are within the sensor range. We assume that we can control the Robot's translational and rotational velocities are described by

$$\frac{dx}{dt} = v \cos\phi \qquad \frac{dy}{dt} = v \sin\phi \tag{1}$$

Where (x, y) is the position of the robot, and Ø is its Orientation. The equations describing the robot are thus the setup corresponds to a path planning problem with incomplete information. The robot doesn't know the location and shapes of the obstacles until they are within the sensor range. This model is attractive because many practical robots operate in unknown environments.

Robotic motion planning consists of two basic tasks. First, the robot must understand what sequences are acceptable (e.g. collision-free) in a given environment. This requires classifying the environment as obstructed or collision free. Second, the robot must understand how to transition through a series of sequences so as to transition from some starting position to some goal position. This requires understanding the Connectivity of the free sequences in the environment. Both of these tasks are most important components of the motion planning problem.

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.

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. The steps of the algorithm of movement are done as follow:

A. Step 1

Move along the free path the set of free sub positions *set_free* towards the target T: **IF** T is reached. Stop **Else** **Go** to Step 2

B. Step 2

Choose a boundary following direction. **Move** around the boundary of obstacle *d_boundary*. The minimal distance along the followed obstacle's boundary to T and the minimal distance within the environment to T, **UNTIL** one of the following occur:
a) T is reached. **Stop. Else**
b) The leaving condition *d_boundary* is not complete. **Go** to Step1.

C. Step 3

The robot completes a loop around the obstacle. T is unreachable. Stop.

D. Step 3

This step is done in order to find the fewest number of *set_free*.

During moving-towards-target, the robot's motions include both motion between obstacles and sliding along obstacle boundaries. Before an obstacle is detected, the robot moves along a straight line directly towards T. And on obstacle boundaries are detected o build the set of *d_boundary*.

In motion planning, efforts are made to ensure that the resulting configuration space has nice properties that reflect the true structure of the space of transformations.

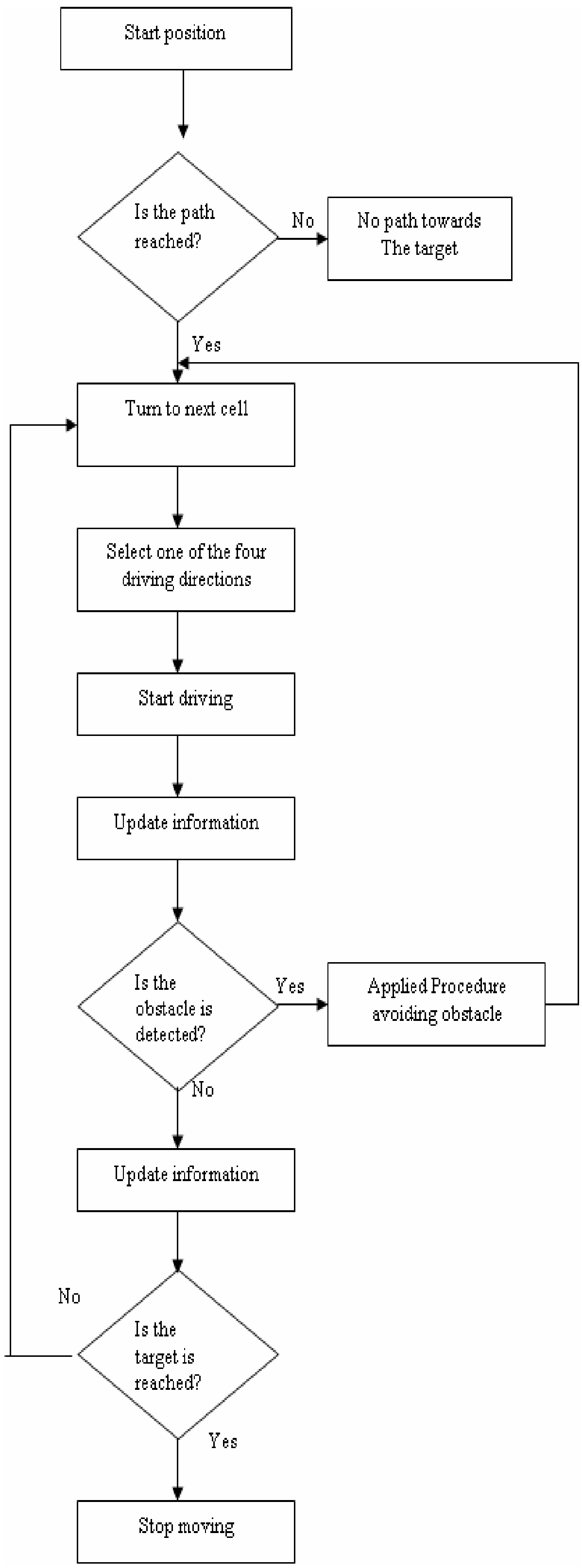

Fig. 1The proposed algorithm

III. The proposed Navigation approach

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

Navigation is a major challenge for autonomous, mobile robots. The problem can basically be divided into positioning and path planning. In this paper we present an approach which we call grid-based navigation. Though we also propose a scheme for path finding, we focus on positioning. Starting out from a position and orientation in the grid, the mobile robot can autonomously head for destination cell in the grid. On its way it determines the current location in the grid by updating the acquired knowledge from he environment.

It is sufficient that the full grid is rectangular and all cells are the same size. It is not required that the robot knows the dimension (n, m) of the grid. It is satisfactory to only request navigation to cell that really exists, see the figure3.

Heading from cell to cell, the robot distinguishes four (04) driving directions: right, left, diagonal upper, diagonal lower so on, see the figure1.In this figure we present one example of navigation approach using a square cellule grid for the movement. Depending on the size of the robot and the complexity of the environment: the robot estimates it's safely way from the source position to the target position.

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.

Assume that the navigation process 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(x, y) from cell ″x ″to its adjacent cell ″y ″ is defined by the Euclid distance from the center cell ″x″ of one cell to the center cell ″y ″of another cell. Each cell in this grid is assigned two statements: *occupied* or *unoccupied*. A cell is unoccupied 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 unoccupied are (free area) or occupied area (hazardous containing obstacles).

After choosing a boundary following direction. The movement around the boundary of obstacle creates a set of *d_boundary*. The minimal distance along the followed obstacle's boundary contributes to build a connexion between free cells. This connection is belonging to *connectivity_cell set* which is the group off all connected free cells that are able to build the feasible path. A feasible path is the meaning of a possible free trajectory towards the target position In the grid space, the algorithm of *connectivity_cell* is done as follow:

```
Algorithm connectivity_cell
Begin
 Move along the free cell
 IF (Right Cell OR Left Cell OR
 Diagonal upper Cell OR
        Diagonal lower Cell ) is free
 THEN
 Begin
 Move along the free cell
      IF (Target Cell is detected) Then
      STOP
        Else
             Begin
              While free
              Move
             IF (Cell Obstacle is
                detected ) THEN
                Begin
                   Done Avoid_
                   hazardous_Cell
                   position  Else
                    Move until
                   Target_cell position
                End;
     End;
 End.
```

The robot requires only very basic sensing and actuation capabilities. 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.
- A void situations that are harmful to people, property, or it self unless those are of its design

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.The 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. In the figure 4 shows an example of a robot path from the starting to the target point with the presence of obstacle. As we can see the principle of work is clarified in this figure.

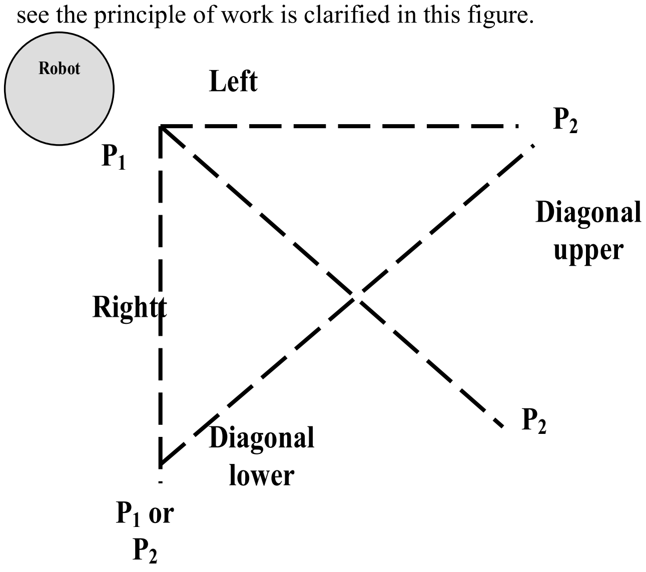

Fig. 2 The four driving direction

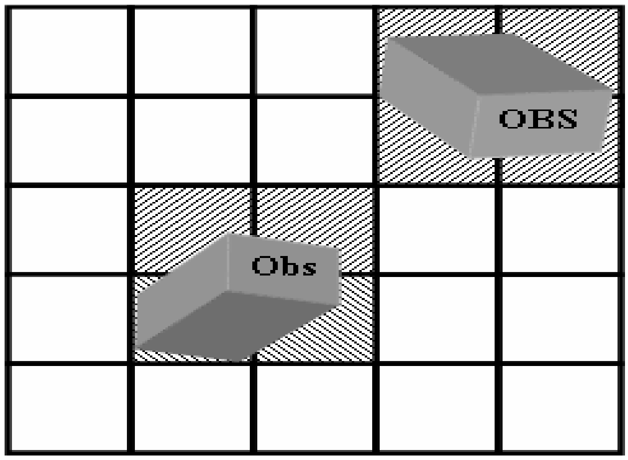

Fig. 3 an example of a rectangular cellule grid 3D navigation

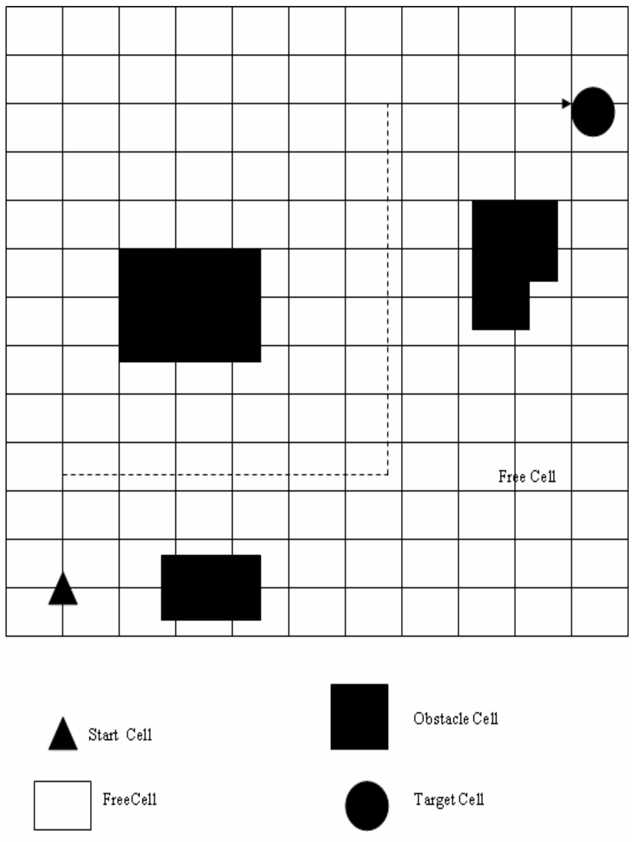

Fig. 4 an example of rectangular cell navigation

IV. Simulation Results

The robot navigates on a grid (see Figure 5) which regularly divides the ground into cells that are identified with Cartesian Coordinates. As mentioned earlier, the robot starts out from a Source position S and orientation, e.g, Left, Right, diagonal upper, diagonal lower (check the most near free cell between them randomly and starts the movement, the procedure *connectivity_cell* and the algorithm of navigation are applied after that). It is sufficient that the full grid is rectangular and all Cells are the same size. It is not required that the robot knows the dimension (n, m) of the grid. It is satisfactory to only request navigation to Cells that really exist.

Heading from Cell to Cell, the robot distinguishes the four direction driving the robot requires only very basic sensing and actuation capabilities. The process of driving to a destination Cell always follows the same step-algorithm seeing previously. Determining the next cell is done via a breadth-first search: starting from the current cell, the coordinates of all adjacent cells that have not been visited on the way to the current destination are stored into a vector together with information on its predecessor. If the destination cell is not among the Connectivity_cell set, again each entry's all adjacent cells and not yet considered cells terminates when the destination cell is in one of the *connectivity_cell* set . The path to cell that destination cell can then be derived using the chain of juxtaposed cells stored together with the coordinates. This rather naïve approach has potential to be optimized. Once a path has been derived, the robot turns towards the direction of the next cell (which can be calculated from the current and next coordinates) by turning both wheels in different directions.

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.

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 starts from any position then using fuzzy logic learning must move and attends its target. 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 targett at the same line. As an example: the environment set up is shown in the figure 6. The path is found by figuring all the cells. Once the path is found, the robot moves from one cell to the next until the target is reached, once we have simplified our search area into a convenient number of sub positions , 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 point, checking the adjacent cells, and training until we find our target. We start the search by the following steps: we have selected the starting position; it moves forward as shown above in figure 7.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 8.

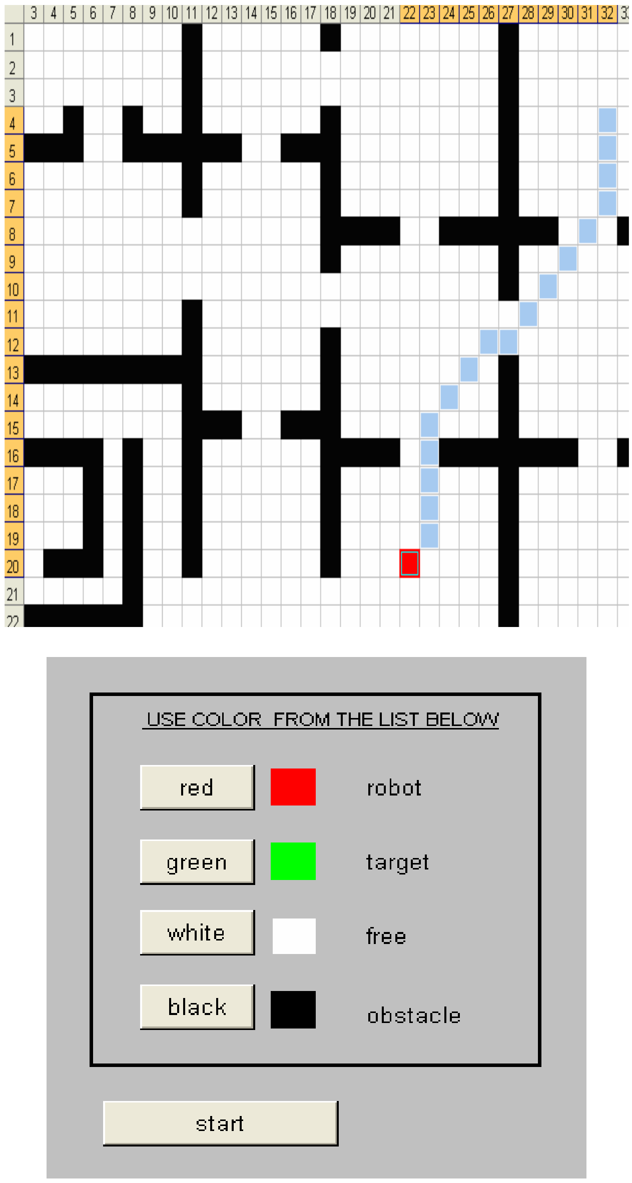

Fig. 5 an example of environment1 set up

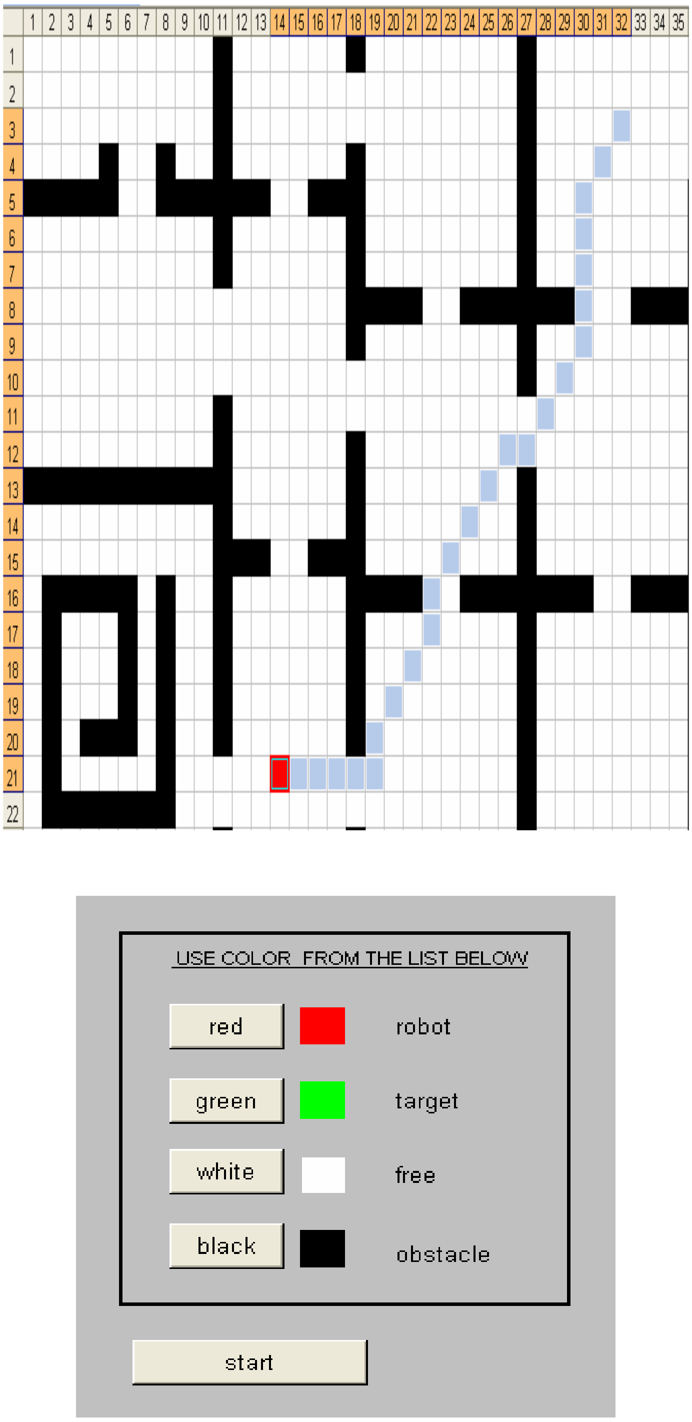

Fig. 6 an example of environment2 set up

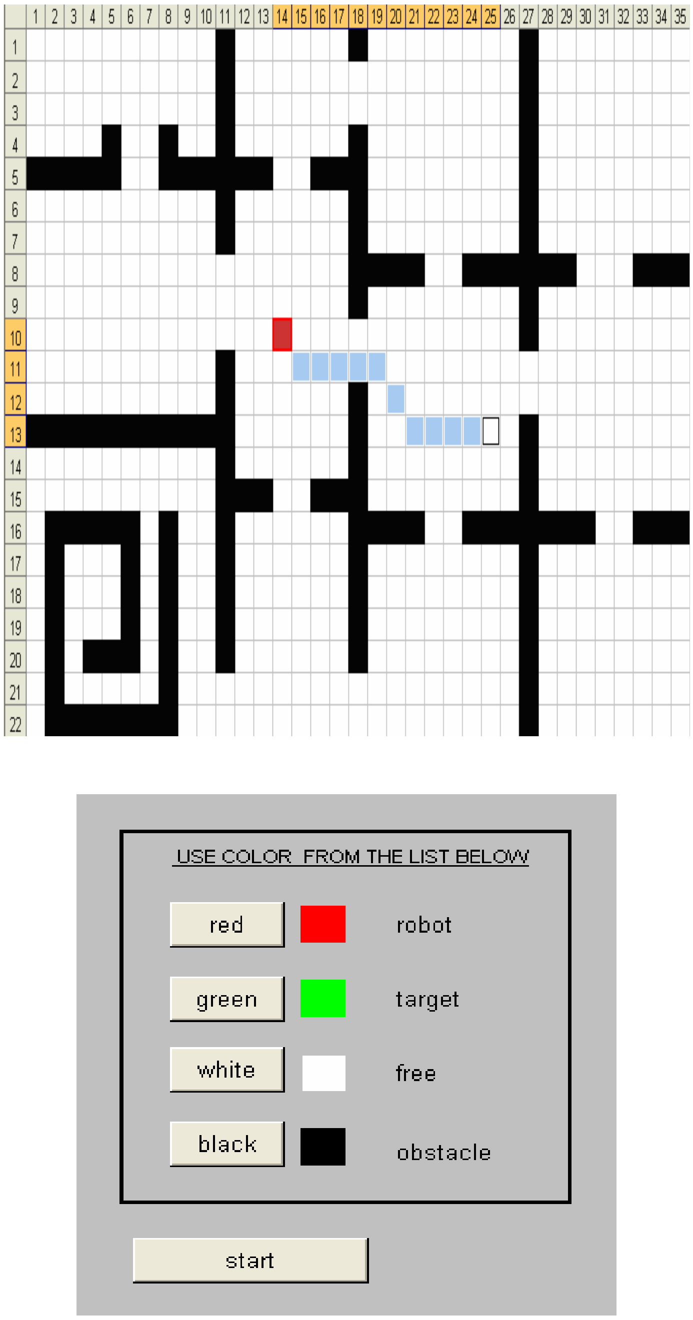

Fig. 7 an example of environment3 set up

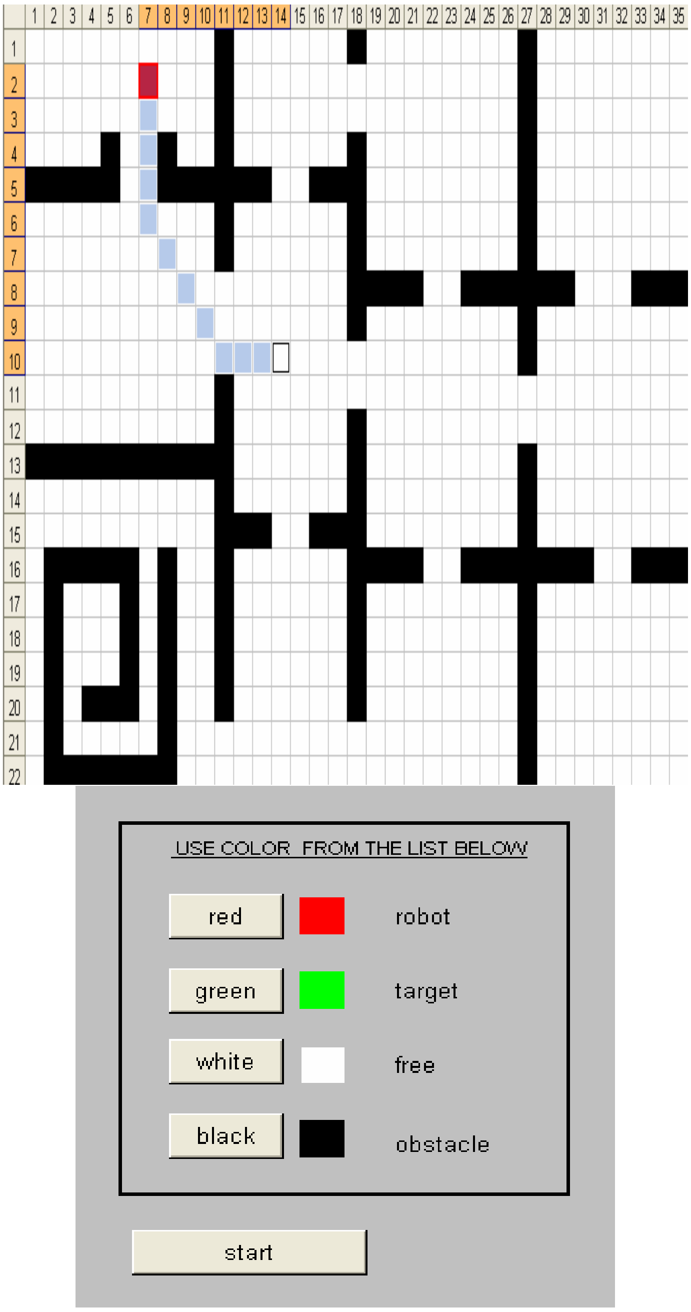

Fig. 8 an example of environment4 set up

V. Conclusion

In this paper we presented an approach for mobile robot positioning and navigation which we call grid-based navigation. Starting out from a start location and orientation in the grid, the mobile robot can autonomously head for destination Cells. On the way it determines its location in the grid using the principle of *connectivity_cell* set and the boundary of obstacle *d_boundary*. We demonstrated how we implemented the underlying algorithm in software. A possible approach is to calculate the next cell to go to whenever the robot reaches a new Cell. A possible metric is the line of sight distance to the destination Cell.

To carry out tasks in various environments as in space applications, the robot succeeds to reach its target without collisions. The environment of navigation can be changed by user demand, that means that the robot can move in another environments where a given shape is designed by the user : a square and rectangle. The Robot come only move to 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.

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. The proposed approach can deal a wide number of environments. This navigation approach has an advantage of adaptivity such that the intelligent autonomous mobile robot 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, and 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. 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.

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] S. M. LaValle, PLANNING ALGORITHMS, *University of Illinois*, Cambridge University Press,2006.

[4] 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.

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

[6] P.Steinhaus, M.Strand, and R.Dillmann**,** Autonomous Robot Navigation in Human-Centered Environments Based on 3D Data Fusion, Hindawi Publishing Corporation EURASIP Journal on Advances in Signal Processing, Volume 2007, Article ID 86831, 10 pages.
[7] A.KRUPA and F.CHAUMETTE, Guidance of an ultrasound probe by visual servoing, *Advanced Robotics*,2006, Vol. 20, No. 11, pp. 1203–1218.
[8] 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.
[9] A. Howard, M.J MatariĆ and G.S.Sukhatme, An Incremental Self-Deployment Algorithm for mobile Sensor Networks, autonomous robots, special Issue on Intelligent Embedded Systems, 13(2), September 202, pp. 113-126.
[10] 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.
[11] K. Schilling and C. Jungius : Mobile robots for planetary explorations, *in Proceeding second international conference IFAC*, Intelligent Autonomous
[12] S. Florczyk, *Robot Vision Video-based Indoor Exploration with Autonomous and Mobile Robots*, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2005.
[13] 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.
[14] 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.
[15] 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 .
[16] 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.
[17] 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.