A three dimensional collision- free-path planning

O. Hachour

Abstract— In this present work we present a 3D (three dimensional) collision free path planning of autonomous mobile robot. The planar is considered as 3D smoothed cubic B-spline surface. The 3D algorithm allows a mobile robot to navigate through static obstacles, and finding the path without collision. The proposed method can deal a wide number of environments and gives to our robot the autonomous decision of how to avoid 3D obstacles and how to attend the target. The path connecting the start point to the target point can be interpolated with a NURB-spline (*nonuniform rational B-splines*) curve to obtain a smooth curve which fits the surface perfectly. The NURB-spline curves are flexible and powerful mathematical tool to design and control the shapes of three-dimensional surfaces. The proposed algorithm using NURB-spline can deal with 3D environment complexity and finds the optimal path The results are satisfactory to see the great number of environments treated. The results are satisfactory and promising. The proposed method is computationally efficient and is suitable for more 3D applications.

Keywords— Autonomous Mobile robots, Navigation, 3D Path planning, Cubic B-splines .

I. Introduction

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.

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 [1,3,4,10,11]. 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 [2].

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, 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 .

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 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.

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 [2,7].

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.

In assistive robotics, a manipulator arm constitutes one possible solution for restoring some manipulation functions to victims of upper limb disabilities. The aim of research work is to present a global strategy of approach of an assistive mobile manipulator (manipulator arm mounted on a mobile base). A manipulability criterion is defined to deal with the redundancy of the system. The aim is to keep the arm manipulable, *i.e.* capable of moving by itself. The strategy is based on human-like behaviour to help the disabled operator to understand the action of the robot.

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 work, a simple and efficient three dimensional collision free-path planning using linear parametric curve 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 3 D obstacles and how to attend the target. More, the 3D 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 3D 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.

Our 3D path planning is based on the use of a linear parametric smoothed curve, where the trajectory is expressed by the parameters S and T that represent the coordinates of the starting and the target points. The directional property of the parametric curve enables us to express the intersection of two segments of a line in simple terms. This is to determine if the line ST collides an object (obstacles) or no. In the workspace, we have defined the position of the obstacles with respect to the line *ST*, and there exist three cases: the obstacles which cut the line *ST*, which are above the line *ST* and which are below the line *ST.* First, we consider the planar navigation in 2-dimenssion coordinates (The 2D simulation), when it is done, we translate the principle into 3D level, where we have to take into consideration 3D obstacles shape and the real robot dimensions**.**

To deal with the principle of the smoothness trajectory: the use of the NURB-spline technique is very useful to solve the problem of 3D environment. The 3D path planning problem can be reduced to a problem of two dimensions by projecting the objects on the plan containing the initial point, the target point and the control point. In this case, the planar is considered as 3D smoothed cubic B-spline surface. The path connecting the start point to the target point must be interpolated with a NURB-spline (*nonuniform rational B-splines*) curve to obtain a smooth curve which fits the surface perfectly. The NURB-spline curves are flexible and powerful mathematical tool to design and control the shapes of three-dimensional surfaces. Our 3D algorithm based on NURB-spline can deal with any 3D shape of obstacles. For any proposed environment, the robot succeeds to reach the optimal path without collisions. The proposed 3D path planning algorithm for the autonomous mobile robot stationary obstacle avoidance provides more autonomy and intelligence to see the wide number of tests and simulations . This approach can be realized in efficient manner and has proved to be superior to the combinatorial optimization techniques, due to the problem complexity. We clarify more the principle of work by the following sections

II. 3D Path Planning

In the workspace, we have to take into account the 3D obstacles and the dimensions of the robot. To get a smoothed trajectory, we have applied the NURB-spline which is very useful to solve the problem of 3D environment and surface. We have defined the position of the obstacles with respect to the line connecting Source S and Target T, where three cases arise: the obstacles which cut the line ST, which are above the line ST and which are below the line ST. For the case of 3D we have to take the width of the robot to consideration as it is shown in the figure 1.1. In this context, the path in 3D environment is considered as an object which has the width and the height of the robot. First we consider that it exits a virtual path that we will select it. Then we select all obstacles in the environment. Finally, we create an initially empty set: the algorithm checks the interference between solids in the first selection set against those in the second selection set.

When the starts the checking for interferences, a temporary interference objects are created and included to last created selection set explained before. In order to avoid the obstacle, the virtual path must be rotated around the starting point S, by a small angle. After this rotation, the temporary interference objects that are created during the interference checking are deleted (see the figure 2). Thus the interference selection set becomes empty. The process iterates again by another rotation $\Delta\Theta$ and checks for collision by testing the interference selection set. When the linear path is in collision with obstacles unless the selection set is not empty. It keeps rotating until the interference selection becomes empty. When the selection set is not empty, the created interference objects must be removed before the next iteration. The iteration is limited for obstacles which interfere with the linear curve.

A. Control point insertion

A control point is created between the starting point and the obstacle when the linear path collides an obstacle. The control point Q, belongs to the characteristic curve network describing the trajectory without collision projected in 2D workspace. The polar coordinates are used to define a control point with the parameters Θ and ρ. The arbitrary values of Θ and ρ are fixed between 0<Θ<360° (Θ≠0, 180°) and 0 <ρ≤1. For the case of non-polygonal obstacles (such as cylinders, cones etc.) it must be circumscribed by poly-solids with the maximum edges.

The checking for interference must be carried out for the obstacles above and below the line ST, hence the rotation is done to the left and the right. As each obstacle produced well defined ρ ranges, the union of these ranges gives us as a result a well defined ρ ranges for a given value of θ. The checking of interference must be made for all the obstacles according to the above procedure. The obstacles in the Euclidian space are represented by a complex form that show only the surface for the control point *Q*. Here we see clearly that the obstacles are represented as 3D objects where the path is traced taking into account the height of the robot. Consequently, we can find a free path throw objects.

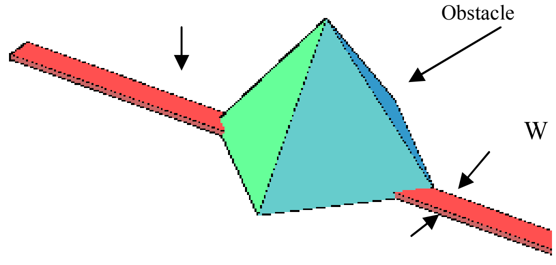

Fig. 1 the Interference between the linear trajectory and the obstacle in 3D environment - isometric view –

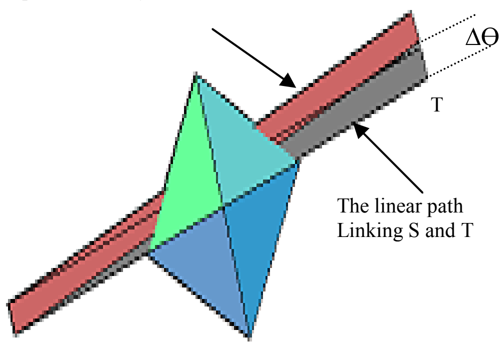

Fig.2 the Interference between the linear trajectory and the obstacle in 3D environment : linear path

B. Connecting smoothing NURBs

In this section we have used the quadratic and cubic parametric curves which generate Bezier curves to remedy the problem of the path gotten in form of segments which are discontinuous in the control point for 2D case.

In our algorithm, initially we add only one control point; then we test for collision between *QT* and the obstacle. In case of collision, we have to insert another control point, and it is for all cases which need more control points. To overcome the problem of non ensure local control points by Bezier method, we apply the NURBs which give more information and produce a smooth curve between control points. We have to keep only the new control points that are located and are in the ordered vector of control to . This is done in order to adjust some of the existing control points and to keep the shape of the curve unchanged.

Before, we had mentioned that the Bezier curves do not ensure the local control. To overcome this drawback we use the NURBs, which are characterised by the following points:

- A NURBS curve produces a smooth curve between control points.
- We create splines by specifying coordinate points which are the starting point, the control points and the end point.
- We can change the spline-fitting tolerance. Fit tolerance refers to how closely the spline fits the set of fit points we specify.
- The lower the tolerance, the more closely the spline fits the points.
- At zero tolerance, the spline passes through the points.
- We can add a fit point (other control point) and refits the spline through the new set of points.

C. Simulation results

The path given in the figure 3 shows the path as a 2D modelling NURB spline[5,8]. To consider the width of the robot we should create a region from a set of entities. This method will create a region out of every closed loop formed by the input array of curves. The first curve is the spline created from the control points, the second curve is a spline shifted away from the obstacles by a distance equal to the width of the virtual robot, and the two ends of the two splines are linked by lines to create a closed region. The previous region presents our free path as 2D planar environment. To take the height of the robot into consideration we have to extrude the region by the height of the robot.

In the figure 6 and the figure 7 the obstacle is avoided by inserting one control point. To get the smoothed path we used the slip of the first portion of the linear path as the start tangent of the NURB spline, and the slip of the second portion as end tangent of the spline (see the figure 8). It is clearly shown that the number of control points and the degree of the NURBs are independent, i.e. the number of control points does not affect the shape of the smoothed curve .Whereas, in the case of the figure 9 the two obstacles are avoided by inserting two control points. Furthermore, the start tangent of the spline is defined by the slip of the first linear path, and the end tangent is defined by the slip of the third linear path. The smoothed path (Isomertic view) is shown in the figure 7.

To go around a vertex, more than one control point must be added when necessary. The start tangent of the spline for each case is defined by slip of the first portion of the linear path; however the end tangent is defined by the slip of the last portion of the linear path. The more is the number of control points the great is the computation time. The figure 9 is an illustration example.

The figure 9 ad the figure 10 illustrate also the path planning in congested environment Linear trajectory (Isomertic view). Where the figure 11 and the figure 12 are an example of navigation in complex environment passing behind obstacles. Another example is more clarified how to pass behind the obstacles in the figure 13 and the figure 14. Until now, although the obstacles are presented as 3D objects, we have dealt only with path planning in 2D planar environment. To deal with no planar path planning we have to start by defining the mesh, then we smooth this mesh to fit the B-spline surface. We avoid the existing obstacles on this surface by using the previous method.

III. 3D FREE COLLSION PATH PALNNING

B-spline are widely used to represent surfaces. They combine a low degree polynomial or rational representation of maximal smoothness with a geometrically intuitive variation of the surface in terms of the coefficients: by connecting the coefficients one obtains a mesh that roughly outlines the surface. Repeated refinement of this mesh by knot insertion results in a sequence of meshes whose points are averages of the preceding and whose limit is the surface itself. In addition to an elegant algebraic definition this yields an alternative geometric, procedural characterization of the splines useful for establishing many shape proprieties of spline surfaces. Each point in the interior of the B-spline mesh must be regular, that is surrounded by exactly four quadrilateral mesh cells.

A Polygon-Mesh object is an M x N mesh where M represents the number of vertices in a row of the mesh and N represents the number of vertices in a column of the mesh.

A mesh can be open or closed in either or both the M and N directions. A mesh that is closed in a given direction is considered to be continuous from the last row or column on to the first row or column. Vertices may be any distance from each other.

A Polygon-Mesh is always created as a simple mesh. A mesh can be smoothed after creation by using even: a quadratic B-spline surface fit, a cubic B-spline surface fit or a Bezier surface fit. The figures (15,16,17,18) show the different cases of smoothing a 4x4 mesh using quadratic and cubic B-spline as well as the Bezier surface concept [6].

The figure 19-left is an illustration example of the smoothed spline curve creation, it is shown as a 3D wireframe. Whereas the figure 19-right shows the 3D path generation with consideration of the robot width. As there are no abrupt changes on the surface, the width of the robot is considered by offsetting the first smoothed curve by a distance equal to the width of the robot. The figures 20 and the figure 21 illustrate the use of our algorithm to avoid the 3D obstacles existing on this surface.

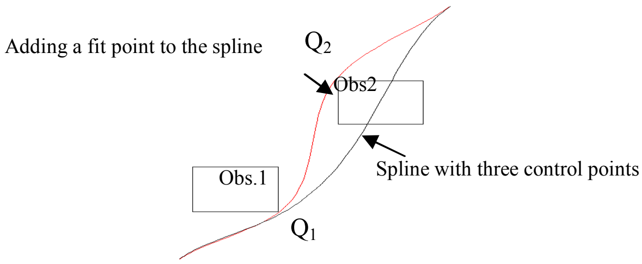

Fig. 3 Addition of control point to a spline curve

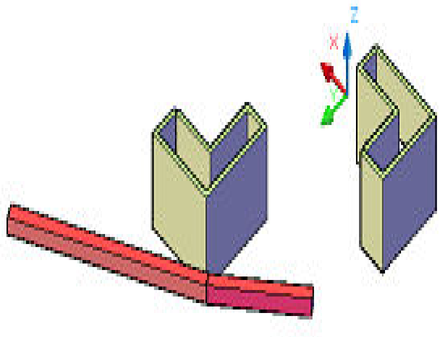

Fig. 4 Free smoothed path realization the insertion of one control point Linear path (Isomertic view)

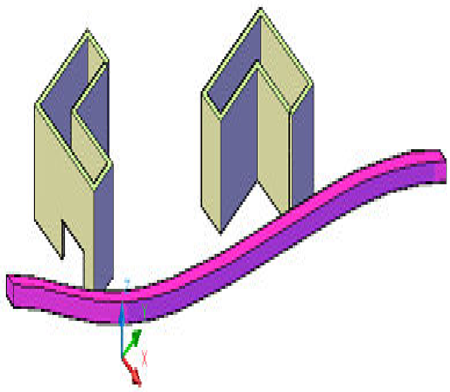

Fig. 5 Free smoothed path realization the insertion of one control point: Smoothed path (Isomertic view)

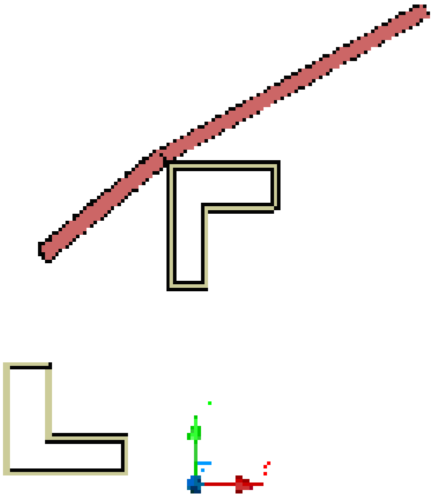

Fig. 6 Free smoothed path realization the insertion of one control point- Linear path (Top view)

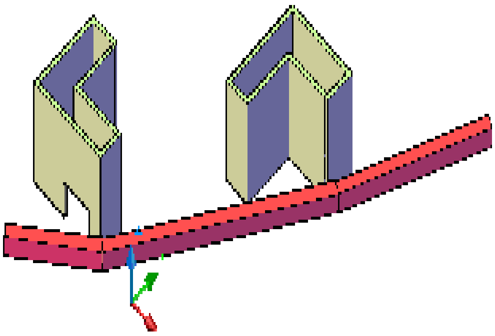

Fig. 7 Free smoothed path realization the insertion of two control points : linear trajectory (Isomertic view)

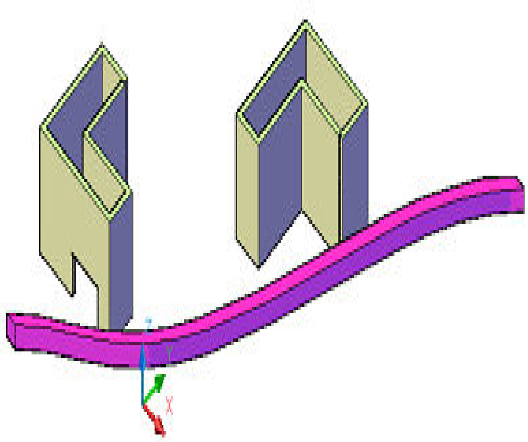

Fig. 8 Free smoothed path realization the insertion of two control points : smoothed path (Isomertic view)

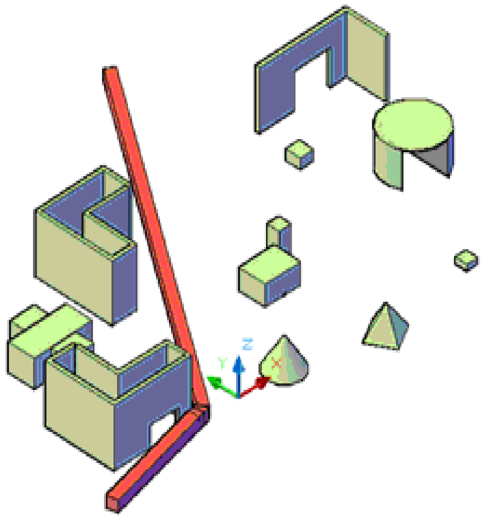

Fig. 9 path planning in congested environment : Linear trajectory (Isomertic view)

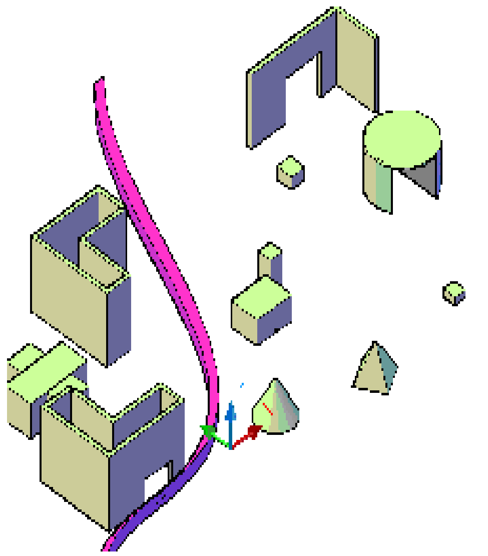

Fig. 10 path planning in congested environment :Smoothed path (Isomertic view)

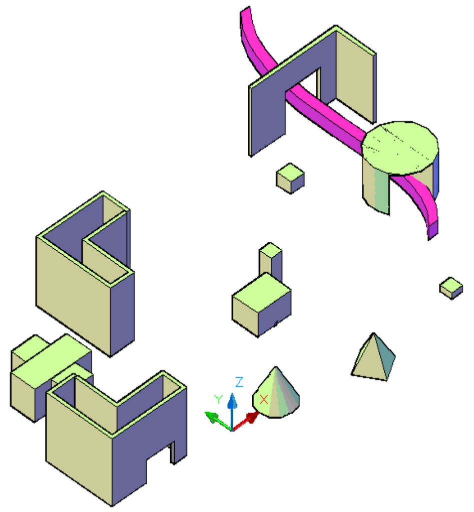

Fig. 12 path planning in complex environment 1:Passing behind obstacles : smoothed path (Isomertic view)

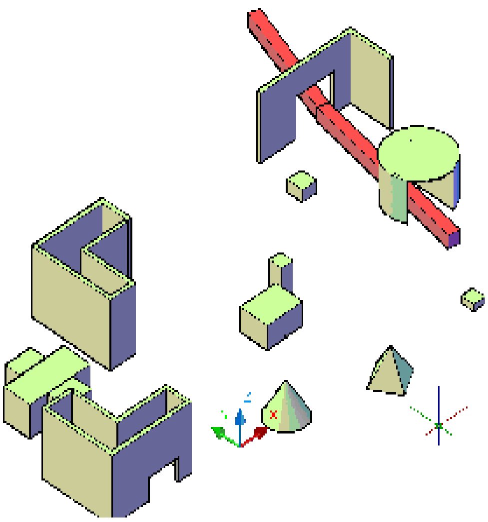

Fig. 11 path planning in complex environment 1:Passing behind obstacles : linear trajectory (Isomertic view)

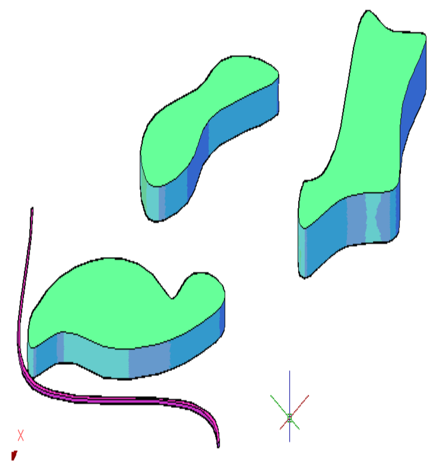

Fig. 13 path planning in complex environment 2:Passing behind obstacles : linear trajectory (Isomertic view)

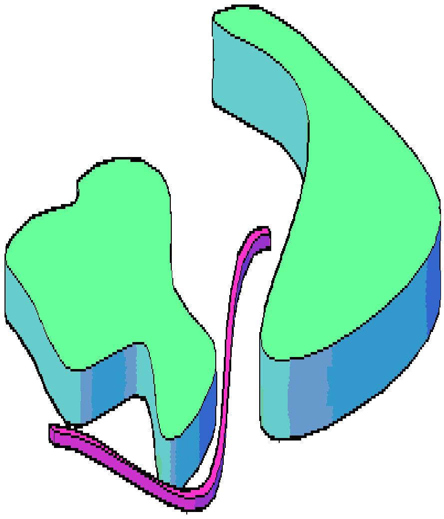

Fig. 14 path planning in complex environment3: Passing behind obstacles smoothed path (Isomertic view)

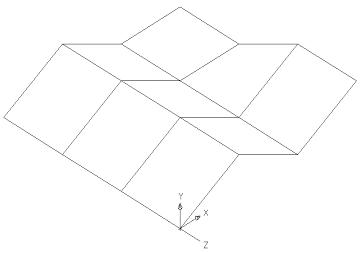

Fig. 15 Mesh creation and 3D surface smoothing: case of 4x4 simple mesh

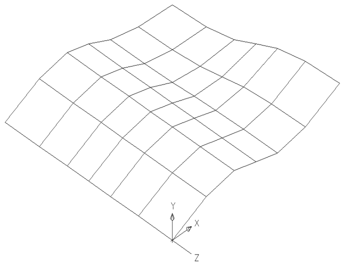

Fig. 16 Mesh creation and 3D surface smoothing: case of Quadratic smoothed B-spline surface

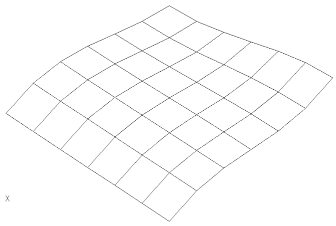

Fig. 17 Mesh creation and 3D surface smoothing: Smoothed Bezier surface

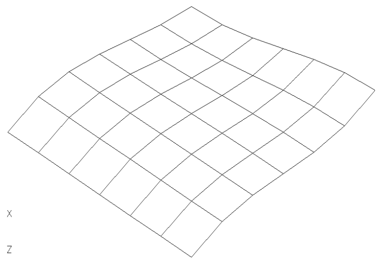

Fig. 18 Mesh creation and 3D surface smoothing: Smoothed Bezier surface

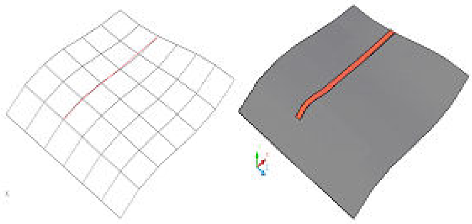

Fig. 19 Smoothed 3D path generation :the smoothed spline curve

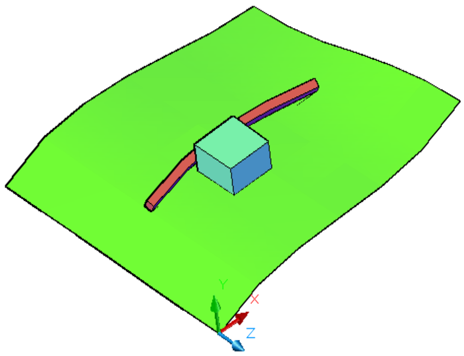

Fig. 20 3D path planning with obstacles avoidance: environment1

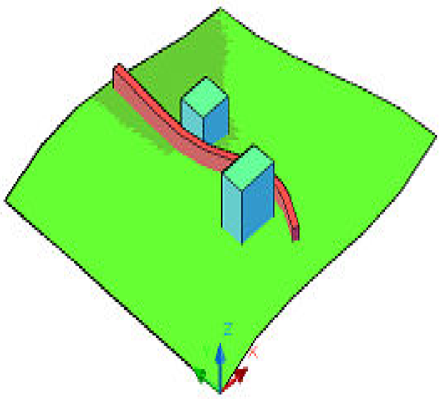

Fig.21 3D path planning with obstacles avoidance: environment2

IV. Conclusion

In this present work we have studied the problem of path planning in a 3-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. We first supposed that the robot navigates in planar environment. In this case, the robot can avoid any obstacles shape even in congested environment. However, for the case of non-polygonal obstacles it is better to be surrounded by poly-solid objects, otherwise the algorithm will spend much time to return a result. And some complex cases the algorithm cannot achieve the target even a free path may exist. The robot moves within the unknown environment by sensing and avoiding the obstacles coming across its way towards the target.

The navigation is done in 3D environment where the planar is considered as 3D smoothed cubic B-spline surface. The obtained path is the shortest path from all possible free trajectories (the smoothness of the trajectory is done around the control point). In this case, the start point and the target point must belong to the control points constructing the smoothed surface. And the obstacles are avoided in the same manner as in the case of planar navigation. The proposed algorithm 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.

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. For the main idea we propose to use simple projection sensors to measure the robot position and orientation. 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. However in the future, it is necessary to use a robot in hostile environment and space exploration or other applications by using advanced micro-product control systems that can be dealt in 3D dimensions surfaces.

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] Ihn Namgung and Joseph Duffy, "Two dimensional collision-free path planning using linear parametric curve", *Journal of Robotic Systems*, 1998.

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

[4] 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.

[5] D. F. Rogers and J. A. Adams, « Mathematical Elements for Computer Graphics », 2nd ed., *McGraw-Hill*, New York, 1990.
[6] Bezier, P., 1972, "Numerical Control-Mathematics and Application, A. R. Forrest and A. F. Pankhurst, Trans., John Wiley & Sons, New York, 1972.
[7] 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,.
[8] I. D. Faux et M. J. Pratt, *« Computational Geometry for Design and Manufacture"*, Jhon Wiley & Sons, New York, 1979.
[9] S. M. LaValle, "*Planning Algorithms*" , Published by Cambridge University Press, 2006.
[10] 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.
[11] 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.