

## Robot Manipulator Path Planning Based on Intelligent Multi-resolution Potential Field

Babak Ranjbar, Javad Mahmoodi, Hasan Karbasi, Gholam Dashti and Ali Omidvar

*Department of Electrical Engineering, Delvar Branch, Islamic Azad University,*

*Delvar, Iran*

*babakranjbar2014@gmail.com*

### ***Abstract***

*Robot path planning is an important part of the development of autonomous systems. Numerous strategies have been proposed in the literature regarding mobile robots but trajectory planning for manipulators is considerably more difficult since the entire structure can move and therefore produce collisions with surrounding obstacles. This paper presents an original solution and analytical comparison to path planning for manipulator arms. Path planning is executed in two parts: first, a global path is found to guide the end-effector in the environment using artificial potential fields and multi-resolution occupancy grids, then, a local path is determined for the entire robot structure by considering the kinematics of the robot as well as the repulsive forces of nearby obstacles in a fuzzy logic controller. Results are shown from a simulator that has been built for this purpose. The contribution of this research is to develop a robust solution for path planning with collision avoidance: one that can be used for various manipulator arms and environment configurations.*

**Keywords:** Robot path planning, fuzzy logic control, artificial potential field, robot manipulator

### **1. Introduction**

Robotic systems are becoming an integral part of our society. In manufacturing, robots have completely changed the way goods are made. But this is not the end of robotic technology. For many reasons, such as human safety or economics, it is desirable to have robots able to complete complex tasks with minimal or no human intervention. However, this is still far from being accomplished. Moving a robot is a complex operation that requires a lot of knowledge about the environment. This information must be analyzed and interpreted to conduct the desired task. In most cases, the goal is to safely move the robot to a target position to perform a deed. The action of determining the sequence of steps to do this is called path planning. For a path planning algorithm to be effective, it must consider numerous parameters: reaching a target position and avoiding nearby obstacles. There has been a lot of research performed on path planning for mobile robots. Many researchers have proposed very efficient methods. However, in the case of manipulator arms, the structure of the robot adds to the complexity of the solution. Although many approaches have been put forth, there are none that can truly be called robust where the algorithm is independent of the robot configuration and works in an arbitrary environment situation.

Although path planning is an essential step in robot automation, many of the approaches used are simple heuristics: successively searching for face connected cells or configuration with or without the minimization of a criterion. These usually provide limited results depending on the complexity and configuration of the environment. These techniques are

applied to both mobile robots and manipulator arms. In an attempt to improve their performance and robustness, numerous alternatives to those methods have been proposed.

Lee and Kardaras [1-2] present a different path planning algorithm for mobile robots that makes use of “via points” (denoted VP) that connect to form the path of the robot. The number of points varies according to the complexity of the surrounding environment: the path along a straight line or in free space will generally require very little number of VP, while the paths closest to an object, which usually tends to “curl” the path, will introduce numerous VP. Simulated annealing is used to smooth out the resulting trajectory. The use of a neural network properly trained is also proposed to make the entire process more parallel.

Nam, Lee and Ko [3] present an approach for collision avoidance of moving objects. The proposed approach recalculates the potential field only for a small region known as the accessible sweep region, which reduces the complexity of rebuilding the entire potential field. The authors use the knowledge of the position, speed and acceleration of the objects to move the mobile robot.

However, this requires an enormous amount of calculation, as numerous measurements need to be taken at very short intervals. From the speed and acceleration data obtained, a form of regression is used to find the most adequate solution using heuristic approaches. The approach is demonstrated to work mainly on simplified environments where distances are large; therefore the sampling time is larger.

In [4] by Nishimura *et al.*, potential fields are used in conjunction with genetic algorithms for path planning for a manipulator arm. The potential fields are used to guide the end effector, while the genetic algorithm ensures collision avoidance for the rest of the structure of the robot. The path planning sequence is not very well defined but the paper presents a very good overview of generic algorithms in the context of robot path planning. The variations in joint angles are coded as genes. This gene is then passed through a series of genetic operations: genetic coding, fitness, crossover, mutation, natural selection and parameter tuning. These operations mutate the gene to a pseudo-random position that is evaluated. These operations are stopped when the robot has reached the desired configuration. The results reported confirm the effectiveness of the proposed solution. There are a few drawbacks to this approach that can be observed, most notably is that the algorithm finds a solution through pseudo-random means. Even though the randomness is controlled, the authors do not include any estimation of execution time, which leads to the deduction that this approach may be very lengthy.

Solteiro Pires and Tenreiro Machado [5] also present an approach to collision avoidance using a standard genetic algorithm. Their concept is to minimize a penalty function that represents the configuration of the robot manipulator (obstacles, robot position, angular speeds...) by using the fundamental functions of genetic algorithms (reproduction, crossover and mutation). The approach used is simplified and test environments are simplistic. However, the authors show the ability to avoid collisions in path planning.

In [6], Oriolo *et al.* present a heuristic-like approach to path planning where the end-effector path is given and must be followed in a tracking operation. The algorithm presented by the authors determines the position of the structure of the robot arm throughout the given path. This path is segmented into smaller steps where the robot configuration must be computed. Every possible configuration of the robot is analyzed until one is found that does not create collisions with the environment. Those solutions are found in a random order and do not guarantee the optimal solution in terms of joint displacement and computation time.

Lin and Chuang [7] offer a different perspective on manipulator path planning. The authors propose to use guide planes (GP) as intermediate goals in the 3D workspace. They provide a general direction for the manipulator to move forward towards the goal by specifying

polygons that represent regions between the obstacles. Using continuous repulsive fields, the algorithm finds the path with the lowest value for repulsion within the boundaries of the GP. This method yields good results; however, the authors assume “that the sequence of GPs is given in advance”. The generation of the GP is not precisely defined. By analysis, this approach seems to avoid obstacles when they are far from each other, but may lead to some problems when it is not the case.

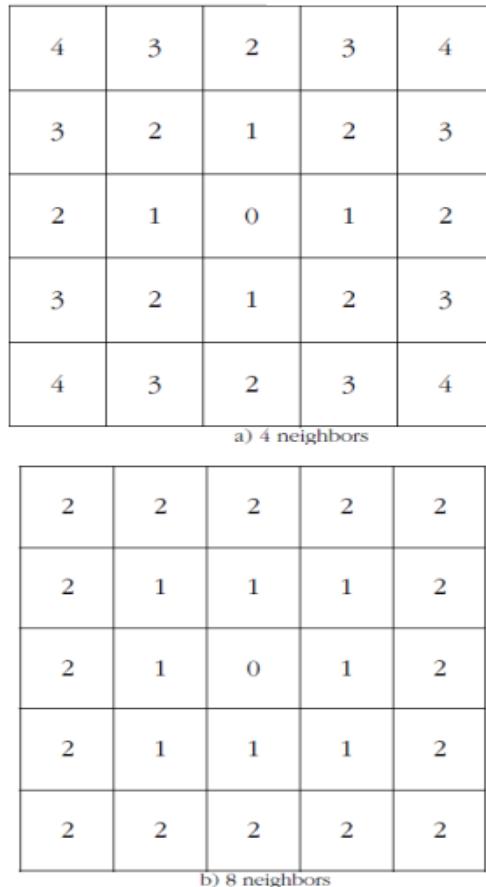
Hwang *et al.* [8] present an interesting approach for path planning with mobile robots that relies on adapted environment representations. The authors observe that the principles of quadtrees (and octrees) possess two intrinsic drawbacks: the quadtree representation needs to be of high resolution to model the obstacles correctly and the quadtree method cannot guarantee to find the shortest path due to large cells that tend to appear in free space. Instead of using square cells in the quadtree sequence, a 2D environment is converted into a triangular grid and blown into 3D where the slope of the triangle represents the proximity of an object. Free space will have a zero “z” component while obstacles will be nonzero. The resulting grid is then simplified by removing some vertices representing the edges of the triangles; this is done to reduce the size of the mesh by exploiting the redundancy of regions of similar occupancy. The path planning strategy is not extensively detailed, but it uses forces to push the robot and generate a path using a heuristic approach. Path planning results are however good, as the approach reduces the computation time by a factor of 10 (over standard quadtrees). On the other hand, the mesh computation is very lengthy (50 sec compared to 0.09 sec for path planning). No extension is proposed for an equivalent mapping for 3D environments [9-11].

This paper is organized as follows; section 2, is served as an introduction to the dynamic of robot arm, design linear PID controller and fuzzy inference system. Part 3, introduces and describes the methodology algorithm. Section 4 presents the simulation results and discussion of this algorithm applied to three degrees of freedom spherical motor and the final section describe the conclusion.

## 2. Potential Fields

Potential fields have taken an important place over the last decade in robot path planning research as they offer a very interesting approach that is relatively simple and efficient. The overall potential field is made up of an attractive field, which attracts the robot towards its objective, and a repulsive field that pushes it away from the obstacles. As discussed previously, many of the path planning approaches still makes use of simple heuristic algorithms, which may yield a quick result but are usually not optimized or require enormous computation. When not designed from a general enough perspective, an important drawback of potential fields is that the robot (whether a manipulator arm or a mobile robot) tends to get trapped in what are called local minima. These minima in the potential surface prevent the robot from reaching its destination. Originally proposed by Khatib [12], potential fields have been refined for a number of years and used for many applications including path planning for robots. They offer a simple yet efficient method to encode the location of obstacles in a given environment through a representation that can be directly interpreted by classical path planning techniques. A simple analogy to potential fields could be to picture a marble rolling down a hill and letting it reach the lowest point of a valley. Provided that this point in the valley corresponds to the target configuration to reach, a trajectory can be computed for the robot by following the negative gradient of the surface. This section summarizes the main techniques of potential fields [12-15].

The repulsive potential fields result from the presence of obstacles in the environment. Their purpose is to repel the robot away from these obstacles. These fields are generally computed as a function of the distance from objects boundaries. As the object most likely to create a collision is always the closest one, only that distance is usually taken into consideration. A safety margin may be introduced around the objects to account for the structure of the robot and the uncertainty of sensors to measure an obstacle's location. In a typical application, the distance to the nearest obstacle is first computed for every point in the environment. The resulting array is an amalgamation of distances. By design, all immediate neighbors of a point are of similar distance, also, the distance of a point to the closest obstacle can be, at most, one larger than the smallest of its neighbors; it can also be only one smaller. For example, the neighboring points to the obstacle edges are assigned a distance of 1, and then the neighbors to those points are assigned a distance of 2, *etc...* For a Cartesian representation, made of a discretization of space in a grid of cells of a given resolution as shown in Figure 1, every point can have either 4 or 8 neighbors, depending on whether diagonal neighbors are considered to be adjacent cells or not. When extended to 3D space representation, 4-neighbor mapping results in 6 neighbors, and 8-neighbor mapping leads to 26 neighbor cells. In this research, we have chosen to use the representation with 8 neighbors in 2D and 26 neighbors in 3D as it increases the flexibility of path planning in cluttered space [16-19].



4	3	2	3	4
3	2	1	2	3
2	1	0	1	2
3	2	1	2	3
4	3	2	3	4

a) 4 neighbors

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

b) 8 neighbors

**Figure 1. Neighbor Mapping in a Grid-based Representation**

The table of distances for a sample environment is given in Figure 4 where the black squares represent obstacles while the number inside the white cells represents their respective distance to the nearest object. In order to consider the lack of measurements taken beyond the edge of the environment, it is usually assumed that outside the workspace is an unknown space and therefore may contain obstacles. Therefore, the border of the workspace is treated as an extra obstacle. As a result, cells are assigned a distance to the nearest object or border as shown in Figure 2.

1		1	1	1	1	1	1		1	1	1	1				
1		1	2	2	2	2	2	1		1	2	2	1			
1	1	1	1	2	3	3	3	2	1	1	1	2	2	1		
1	2	2	2	2	3	4	3	2	1	1	1	2	2	1		
1	2	3	3	3	3	4	3	2	2	2	2	2	3	2	1	
1	2	3	4	4	4	4	3	3	3	3	3	3	3	2	1	
1	2	3	3	3	3	3	3	3	3	3	4	4	3	2	1	
1	2	2	2	2	2	2	2	2	2	3	4	4	3	2	1	
1	1	1	1	1	1	1	1	1	2	3	4	4	3	2	1	
									1	2	3	4	4	3	2	1
									1	2	3	4	4	3	2	1
									1	2	3	4	4	3	2	1
1	1	1	1	1	1	1	1	1	1	2	3	4	4	3	2	1
1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

**Figure 2. Sample Table of Distance**

Once the table of distances is obtained, the repulsive field can be directly computed as an inversely proportional function to the distance for each cell. That is, a large distance means that the position is far from an obstacle and the repulsive field is weak. Inversely, if the distance is small, then the presence of the obstacle will cause a large repulsive field. There is an important problem that plagues the computation of tables of distances and repulsive fields. As they rely on a computation of distance to the closest object, a large amount of computation is required to determine the successive neighbors of every point in the environment.

### 3. Detecting and Escaping Local Minima

Barraquand and Latombe [2] have proposed a classical approach for path planning. The authors use a potential-field based method that follows the gradient descent to approach the target position. If the robot becomes stuck in a local minimum, the algorithm tries to escape by the addition of random movement widely known as the Monte-Carlo approach. The process is repeated until the path reaches the target position. The proposed approach shows good results in dealing with manipulators with a large number of degrees of freedom, however, the approach is very lengthy when dealing with narrow corridors. The random search for a valid solution leads to non-uniform planning times and is not repetitive; this represents a major drawback for real-time applications as required in modern robotics. Caselli *et al.* [4, 5] present a method for escaping local minima. The approach determines if the robot is currently located in a local minimum and uses different methods to escape it. The first approach (Straight Line) is to move the robot in a random “up-hill” direction until a criterion

is met. The second algorithm (Straight Line Select) eliminates unwanted candidate directions therefore optimizing the escaping path and minimizing the occurrence of the robot falling in the same minimum. The results obtained are not very suitable for real-time applications; however, the simplicity of the approach is interesting. Park *et al.* [25] propose a similar approach where once the robot is trapped in a local minimum, a random solution is found using simulated annealing from the set of neighbors to the current solution. The simulated annealing method searches for random robot configurations and evaluates them based on an optimization criterion that attempts to free the robot. As the Monte-Carlo approach, this random search method will not lead to uniform results, which limits its use for real-time application. The authors present very good results for mobile robots, but do not mention an extension to manipulator arms. Most approaches assume that the robot has a fairly simple shape. However, in order to be more rigorous, a more precise representation can be used, but it introduces different problems. Chang [6] examines the fact that every point in an object will exert a different force to every point on the robot. Hence different forces are applied to different parts of the robot structure, creating many possible attractive and repulsive fields. The method proposed by the author searches through the fields until a valid path can be found. The trajectories found using the proposed algorithm are very interesting and demonstrate that the approach is able to fully model the robot and determine a safe global trajectory. However, the algorithm requires significant amount of memory to store the robot representation and environment information, therefore leading to lengthy computation times. Chengqing *et al.* [7] present an interesting method to detect local minima. They observe that minima are created most of the time by concave objects, or a series of convex objects forming together a concave object in the workspace. The proposed method is used when the robot is trapped in the local minima as it tries to determine the optimal direction to escape it by determining the largest opening of the concave object. The approach used is interesting, however, real-time application of the proposed approach is not demonstrated.

#### 4. Soft Computing and Path Planning

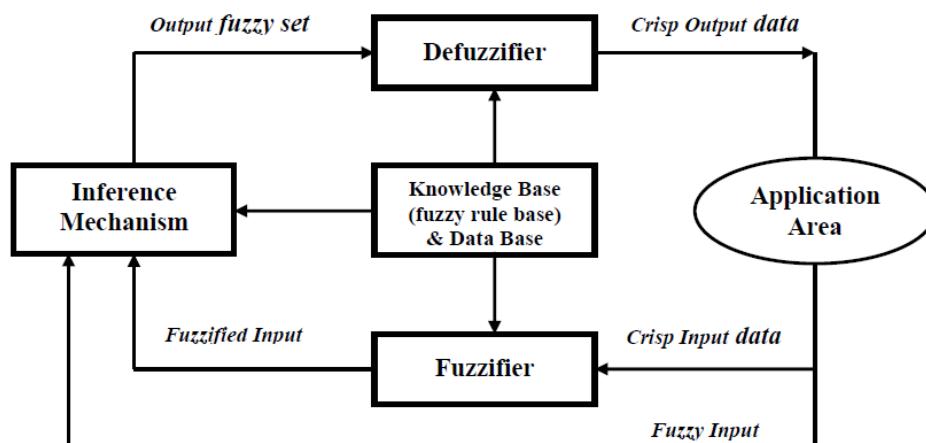
Soft computing is a consortium of methodologies centering in fuzzy logic (FL), artificial neural networks (ANN) and evolutionary computation (EC). What is important, to mention here, is that these methodologies are complementary and synergistic, rather than competitive. They provide in one form or another flexible information processing capability for handling real life ambiguous situations. Soft computing aims to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness and low-cost solutions. The guiding principle is to devise methods of computation that lead to an acceptable solution at low cost, by seeking for an approximate solution to an imprecisely/precisely formulated problem.

The architecture of a fuzzy logic system shown in Figure 3 is composed of:

1. ***knowledge base or rule base***, expressed in the form of a set of fuzzy rules that express the relation between the input and output fuzzy variables;
2. ***data base*** containing the definitions regarding discretization and normalization of the universes, the definitions of the fuzzy domains of each fuzzy variable and the definition of the membership function for each fuzzy set;
3. ***fuzzifier*** to match numerical input with the fuzzy sets of the domain of the correspondent fuzzy variable;

4. *fuzzy inference* mechanism, that is, the algorithm that by firing each of the rules of the knowledge base in parallel obtains an inferred fuzzy set given a set of input values;
5. *defuzzifier* algorithm to calculate the crisp output from the inferred fuzzy set;
6. *Application area*, for example, software development effort prediction.

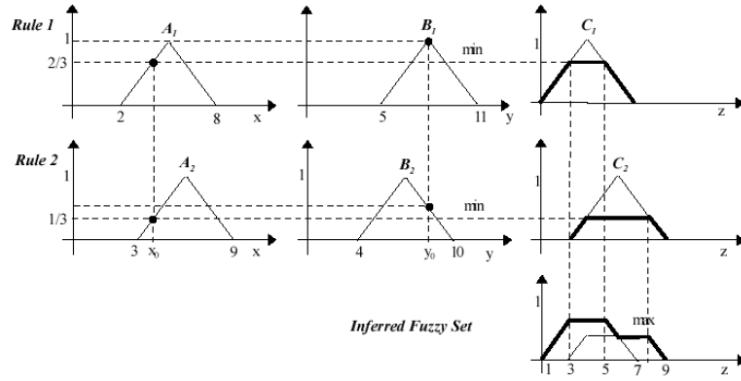
The basic fuzzy inference system can take either fuzzy inputs or crisp inputs, but the outputs it produces are usually fuzzy sets that can be defuzzified to extract a crisp output. Figure 3 below also illustrates the data flows between the constituents of a fuzzy based system. For a crisp input, for example, a set of input crisp values relative to variables in the antecedent of the fuzzy rules is introduced; values are fuzzified and the fuzzy inference system is activated. An output fuzzy set is obtained by matching the fuzzified values with the antecedent of all fuzzy rules in the knowledge base (rule base); and finally the output fuzzy set is defuzzified, that is, transformed to a crisp output value.



**Figure 3. Simplified Fuzzy System Architecture**

The process of matching the input value for a specific fuzzy variable with the fuzzy set that the variable assumes as its value is called *fuzzification*. For more details on fuzzification methods see.

The *fuzzy inference* mechanism is the process by which the input values for each of the fuzzy variables in the antecedent of the rules ( $x_0$  and  $y_0$  in Figure 4 below) are matched with all rules in the fuzzy rule base and an inferred fuzzy set is obtained. The fuzzy inference is a parallel inference in the sense that all rules contribute in a large or small extent to the inferred result. The weight that a specific rule has on the final output is determined by the degree of matching between the input (fuzzified) values and the rule's antecedent. Figure 4 illustrates a simple fuzzy inference mechanism using the max-min inference.



**Figure 4. Fuzzy Inference Using Max-min Method**

**Rule1: IF X is A1 and Y is B1 THEN Z is C1.**

**Rule2: IF X is A2 and Y is B2 THEN Z is C2**

The inferred fuzzy set can be defuzzified to give a non-fuzzy (crisp) output using any of the defuzzification methods.

The process by which a non-fuzzy (crisp) output is obtained from the fuzzy set,  $B^*$  resulting from the fuzzy inference process is called *defuzzification*. The two most commonly used defuzzification methods are [54]: Center of area (COA) method and Center of Gravity (COG) method.

In order for the robot to be able to complete the task it has been assigned to do, it is essential to have it move safely across the workspace. The goal of the path planning method is to determine a sequence of configurations for the robot to move around obstacles and avoid collisions while reaching a desired goal. This work follows an important trend in manipulator path planning which consists of finding a safe trajectory in two steps:

1. A planning phase where the path is determined for the end effector only using the multi-resolution occupancy model, which is usually done offline, and
2. An online tracking phase to follow from the initial path, in which a sequence of robot arm configurations is determined to satisfy the path of the end effector while avoiding contact with the obstacles for the entire structure. The robot moves towards the updated position while the controller finds the next position.

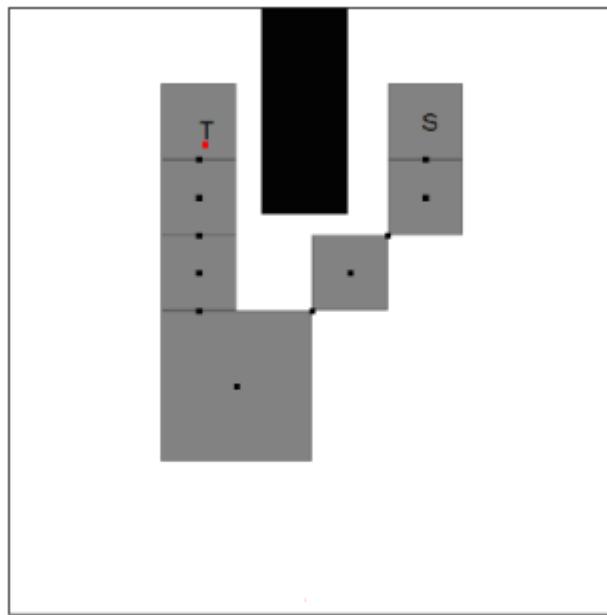
The first path found corresponds to an approximation of the trajectory of the end effector and is based on a “global path planning” method. It is generally preferred for this path to be at conservative distance to obstacles. This trajectory is found by following the down slope of the discrete attractive potential field. This approach uses the advantages of path planning algorithms for mobile robots, that is, in both cases; only a fixed point or structure is considered for movement. For the “path tracking” phase, that extends the trajectory definition to the whole structure, the rest of the robot arm is configured to follow the path of the end effector. Since the direct computation of manipulator parameters through inverse kinematics is a difficult and inflexible process that varies for each robot architecture, an alternative method is proposed to combine the model of the robot arm with the local environment mapping and output the required sequence of joint values.

The purpose of this step is to provide the local robot controller with a generic path to follow during the more refined path tracking phase. We wish to generate a guideline path for the end effector to travel. This path allows leeway to reduce computation times and the

occurrence where the structure of the manipulator may lead to unreachable positions along this initial path. The global path serves as a general indication and is obtained with methods very similar to those found for path planning in mobile robotics. The discrete attractive field computed from the quad tree or octree mapped as a sequence, is the main guide in finding this path. From the starting position and configuration of the robot, the algorithm follows the down slope of the attractive field until the goal or target position is reached. However, as this is a method mainly directed towards mobile robot, or the movement of a single entity, the extension to the control of a manipulator implies that the following assumptions are made:

- Given the current position of the end effector, there exists a configurable position of the robot, *i.e.* the position must be physically reachable by the arm without any collisions.
- The target position is in free space and is reachable with the end effector, *i.e.*, it is not inside a closed region.

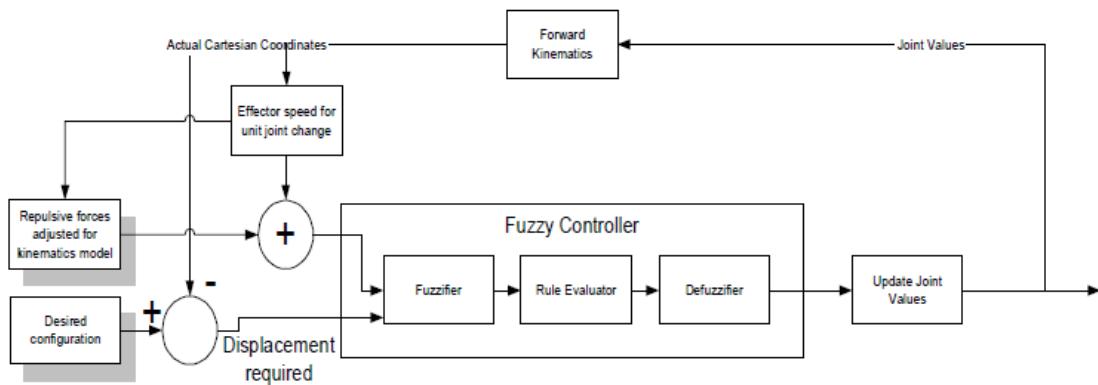
As the result of this first planning phase is considered a coarse path, one that gives a general indication of the movement of the end effector, the output path does not need to be entirely in its highest resolution. Therefore, this global approach can be computed using the multi-resolution cells to minimize the number of neighbor searches and improve performance. This path is consequently a sequence of multi-resolution cells from the starting cell position to the goal cell position used to guide the end effector in the environment. An example of a 2D environment is shown in Figure 5 with a planned coarse path. The global path of the end effector is defined by the sequence of 7 multi-resolution cells shown in grey, starting with “S” and reaching the goal denoted “T”. To minimize the occurrence of a collision, this general path suggests that the end effector successively reaches the center of these cells and the joint border with the next, as indicated with the small squares in Figure 5. The path is to be divided into smaller steps in the path tracking phase.



**Figure 5. End-effector Path**

Safe path planning implies to move the robot up to a desired target position while avoiding collisions between the robot and the environment. The approach proposed here also considers

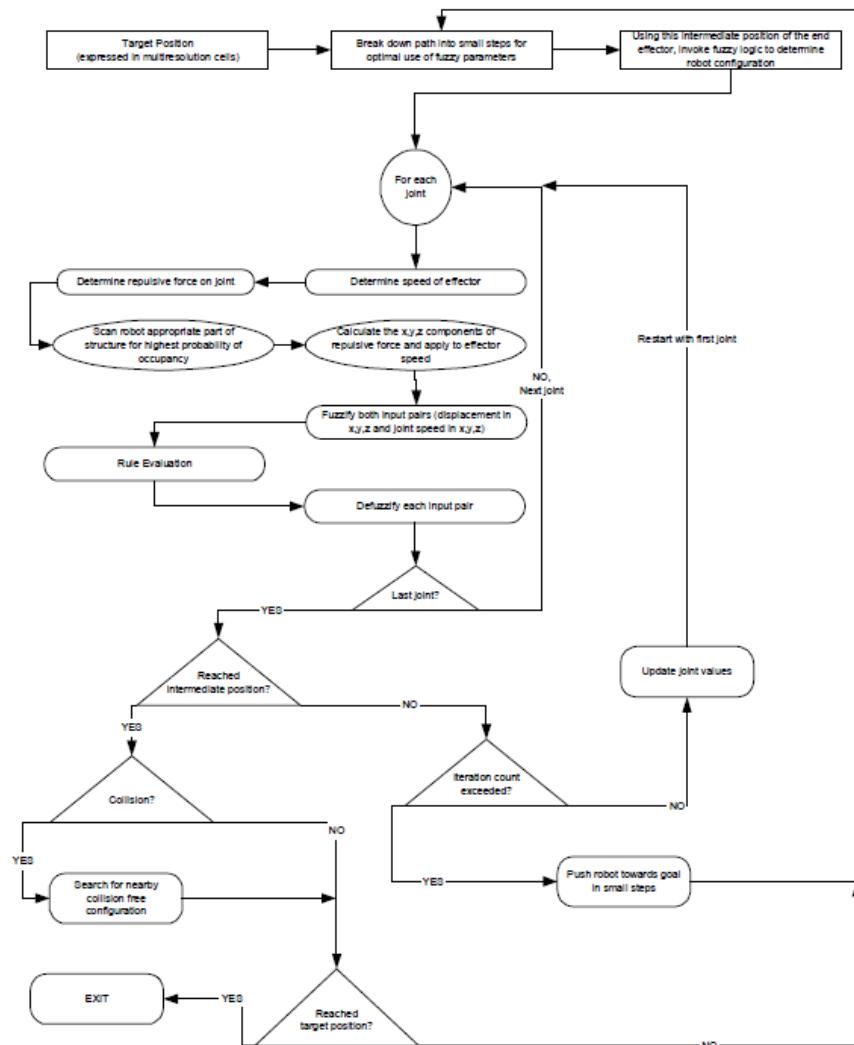
the effects of the environment on the entire robot structure while moving the joints. To achieve this, a supplementary component is added to the fuzzy logic model. One of the inputs to the system that is considered is the probabilistic encoding of the occupancy of the environment. This data provides a relationship with the distance to a given point of a nearby obstacle as they increase for points that are closer to one. These occupancy levels can also be interpreted to be forces that are exerted away from obstacles. Using them in combination with the regular “mechanical” forces implied by the robot kinematic constraints, an approach is developed to guide the robot in the environment. The main problem that exists in any kind of approach using repulsive forces is that it may lead the robot to be trapped between contradictory forces of the same magnitude but working in opposite direction, this is known as the classical “local minimum” problem. Through the proposed algorithm and correction methods, some of these classical problems are overcome. A system diagram of the proposed procedure including collision avoidance is shown in Figure 6.



**Figure 6. Fuzzy Controller**

Repulsive forces are proportional to the probability of occupancy determined by the probabilistic model. The strongest repulsive force is found where the probability of occupancy is the highest at any point on the robot structure. In order to avoid collisions and to reduce the joint displacement when a collision is near, the repulsive forces and the joint velocities are combined. Once the repulsive force reaches a threshold, defined in the following section, the value of the velocity is slowly reduced by the force to provoke a sign change, therefore causing the joint to move in the opposite direction.

The probability of occupation is analyzed for the entire link of the robot at every step of the fuzzy process; this includes checking for collisions during every iteration of the fuzzy logic controller. While this increases computation times, is it a crucial step to avoid collisions during robot movement, and determining the regions in the robotic structure where the probability of the presence of an obstacle is the largest (therefore is at the most risk of colliding with objects) is essential. Figure 7 displays the block diagram of the proposed algorithm.



**Figure 7. Block Diagram of Fuzzy Logic/Collision Avoidance Procedure**

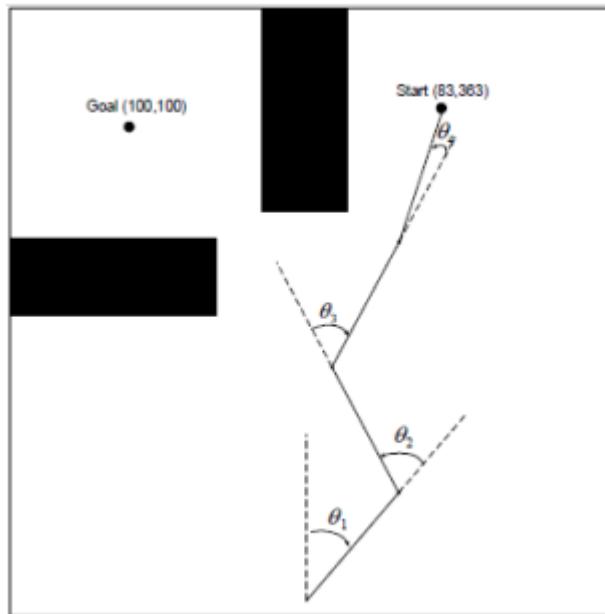
The proposed path planning method has been tested with robots having various numbers of degrees of freedom and with different architectures to evaluate its flexibility, performance and robustness. In this section, the simulation results shown come from two different robot types: generic 3 and 4 degrees-of freedom planar robots and a 3D PUMA-like robot both containing only revolute joints. The Denavit-Hartenberg parameters for the planar robots are shown in Table 1.

**Table 1. DH Parameter for 2D Planner Robot**

Joint	$L_i$	$D_i$	$\alpha_i$	$\theta_i$
1	$L_1$	0	0	$\theta_1$
2	$L_2$	0	0	$\theta_2$
3	$L_3$	0	0	$\theta_3$
4	$L_4$	0	0	$\theta_4$

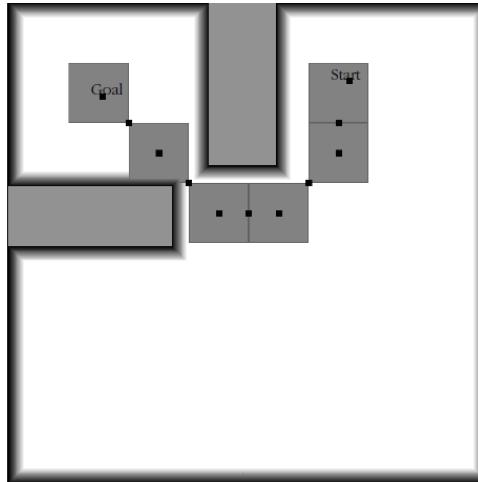
The link lengths,  $L_i$ , are left as variables to provide greater flexibility when changing robot architecture and are encoded as a ratio to the size of the environment.

In the first test case, the base of the robot is fixed in the middle of the workspace near the bottom; [500 250] for an image of size  $512 \times 512$ . The start position is chosen arbitrarily at [83 363] and the target position is also chosen arbitrarily as long as it is contained in the enclave seen in the top left corner, in our case, we have used [100 100]. Figure 8 displays this environment with the robot in its starting position. The obstacles are depicted in black while the free space is in white.



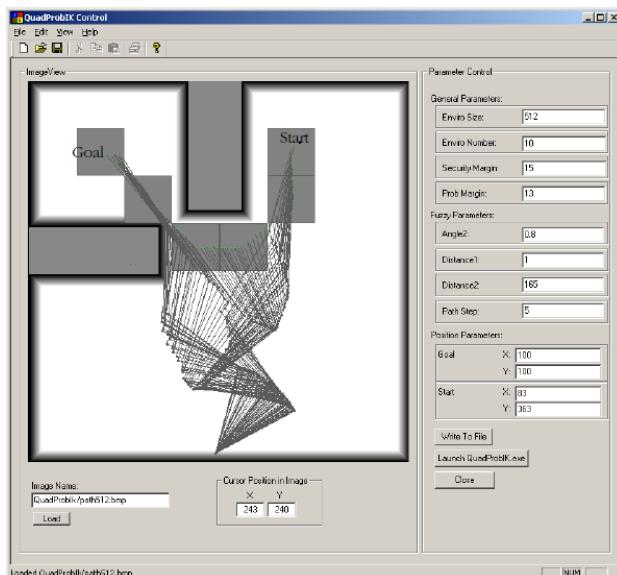
**Figure 8. Initial Robot Configuration**

From the starting position, the free space is encoded into a sequence of multi resolution cells. The attractive field is then computed on the quadtree sequence following the neighbor search techniques and the global path planning method is applied. To guide the path planner, a sequence of neighbor cells are selected by taking for each step the neighbor cell with the smallest distance to the target position. The resulting path is a sequence of multi-resolution cells linking the start and target positions, within the security margin but which may overlap the probabilistic data. The path of the effector is then interpolated to reach the middle of the multi-resolution cells which are depicted in grey in Figure 9. Supplementary intermediate positions are finally computed to link the cells, either through common corners or smallest distance to the neighboring cell. Intermediate points are displayed with smaller squares in Figure 9. This path yields an approximate sequence for the end effector to follow and represents the global path of the end effector.



**Figure 9. End Effector Path**

Next, the global path of the end effector is refined using the fuzzy logic-based path tracking approach to obtain the overall path of the manipulator arm as shown in Figure 10. From the starting robot configuration, the local path tracking method is executed to determine the robot configurations from the global path that has been divided into smaller intermediate segments for which the fuzzy controller is optimized; in this system 5 pixels or about 1% of the workspace size is used. A series of joint angles is computed to reach those segments and displayed in Figure 10. The global path is also shown for analysis.



**Figure 10. Complete Robot Path**

For the preceding figure, the joint values at every step are shown in Figure 11. One of the initial objectives was to propose an algorithm that minimizes the joints' movement between consecutive positions to maximize stability and minimize the risk of collisions that often results from abrupt displacements. By consequence, the occurrence of singularities can be

greatly reduced. This is achieved, as the largest joint displacement performed between consecutive positions is approximately 4.5 degrees in this study.

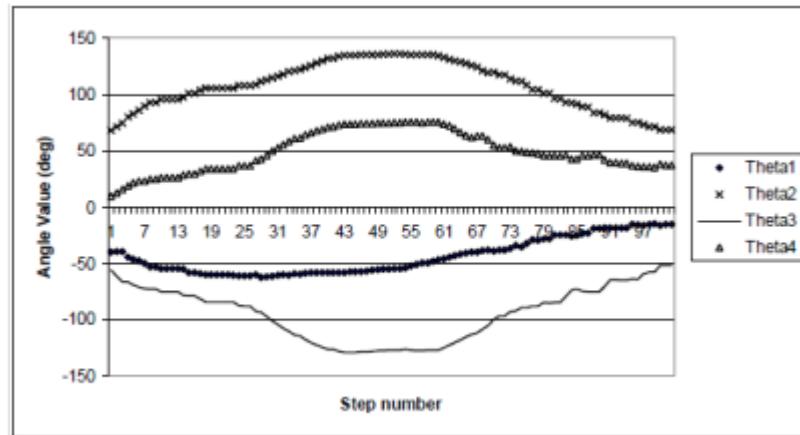


Figure 11. Angles Progression

## 5. Conclusion

This work presents an approach for a path planning strategy for robot manipulators. In opposition to most techniques dedicated to mobile robots that are found in the literature, the proposed path planning approach considers the complexities of modeling a robot arm. An original solution using potential fields and fuzzy logic is presented and analyzed. The aim of the approach is to achieve a robust and adaptable algorithm that can work with various manipulator architectures and environment configurations.

The global path planning phase uses a combination of probabilistic datasets and multi-resolution grids. The use of a probabilistic representation increases the reliability of the mapping and provides a more direct relationship to compute the repulsive potential field. An original approach to encode multi-resolution cells is proposed. This new method greatly reduces the storage requirements for storing environment information. A neighbor searching method is developed on the basis of this encoding and is applied for efficient computation of the attractive field that guides the end effector. The new approach is compared to other algorithms found in the literature and it is shown that its performance meets or exceeds that of the other techniques.

## References

- [1] S. Ando, "A fast collision-free path planning method for general robot manipulator", in Proceedings of 2003 IEEE International Conference on Robotics and Automation, pp. 2871-2877, Taipei, Taiwan, (2003).
- [2] J. Barraquand and J. C. Latombe, "A Monte-Carlo algorithm for path planning with many degrees of freedom", in Proceedings of IEEE International Conference on Robotics and Automation, pp. 1712-1717, Cincinnati, OH, (1990).
- [3] M. T. H. Beheshti and A.K. Tehrani, "Obstacle Avoidance for Kinematically Redundant Robots using an Adaptive Fuzzy Logic algorithm", in Proceedings of 1999 American Control Conference, pp. 1371-1375, San Diego, CA, (1999).
- [4] S. Caselli, M. Reggiani and R. Rocchi, "Heuristic Methods for Randomized Path Planning in Potential Fields", in Proceedings of 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 426-431, Banff, AB, 2001.S. Caselli, M. Reggiani and R. Sbravati, "Parallel Path

- Planning with Multiple Evasion Strategies”, in Proceedings of 2002 IEEE International Conference on Robotics and Automation, pp. 260-266, Washington, DC, (2002).
- [5] H. Chang, “A New Technique to Handle Local Minimum For Imperfect Potential Field Based Motion Planning”, in Proceedings of 1996 IEEE International Conference on Robotics and Automation, pp. 108-112, Minneapolis, MN, (1996) April.
- [6] L. Chengqing, M. H. Ang Jr., H. Krishnan and L. S. Yong, “Virtual Obstacle Concept for Local-minimum-recovery in Potential-field Based Navigation”, in Proceedings of 2000 IEEE International Conference on Robotics and Automation, pp. 983-988, San Francisco, CA, (2000) April.
- [7] E. S. Conkur and R. Buckingham, “Increasing the maneuvering ability of highly redundant manipulators”, in Proceedings of 1997 IEEE International Conference on Robotics and Automation, pp. 155-160, Albuquerque, NM, (1997) April.
- [8] A. Elfes, “Using Occupancy Grids for Mobile Robots Perception and Navigation”, in IEEE Computer, vol. 22, no. 6, (1989) pp. 46-57.
- [9] A. Ham, Z. Qu and R. Johnson, “Robust fuzzy control for robot manipulators”, in IEEE Proceedings of Control Theory and Applications, vol. 147, no. 2, (2000), pp. 212-216.
- [10] J. Y. Hwang, J. S. Kim, S. S. Lim and K. H. Park, “A Fast Path Planning by Path Graph Optimization”, in IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 33, no. 1, (2003) January, pp. 121-128.
- [11] O. Khatib, “Real-time Obstacle Avoidance for Manipulator and Mobile Robots”, in Proceedings of IEEE International Conference on Robotics and Automation, (1985), pp. 500-505.
- [12] Farzin Piltan, Sara Emamzadeh, Zahra Hivand, Fatemeh Shahriyari & Mina Mirzaei, ”PUMA-560 Robot Manipulator Position Sliding Mode Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate/Undergraduate Nonlinear Control, Robotics and MATLAB Courses”, International Journal of Robotics and Automation, vol. 3, no. 3, (2012), pp. 106-150.
- [13] F. Piltan, N. Sulaiman, A. Zargari, M. Keshavarz and A. Badri, “Design PID-Like Fuzzy Controller with Minimum Rule Base and Mathematical Proposed On-line Tunable Gain: Applied to Robot Manipulator”, International Journal of Artificial Intelligence and Expert System, vol. 2, no. 4, (2011), pp. 184-195.
- [14] G. S. Chirikjian and D. Stein, “Kinematic Design and Commutation of a Spherical Stepper Motor”, IEEEIASME Transactions on Mechatronics, Piscataway, New Jersey, vol. 4, no. 4, (1999), December pp. 342-353.
- [15] K. Kahlen and R. W. De Doncker, “CWTent Regulators for Multi-phase Permanent Magnet Spherical Machines”, Industry Applications Conference Record of the 2000 IEEE, vol. 3, (2000), pp. 2011-2016.
- [16] K. M. Lee, I. Pei and U. Gilboa, “On the Development of a Spherical Wrist Actuator”, Proceedings of the 16th NSF Conference on Manufacturing Systems Research, Tempe AZ, (1990), January 8-12.
- [17] F. Piltan, M. H. Yarmahmoudi, M. Shamsodini, E. Mazlomian and A. Hosainpour, ”PUMA-560 Robot Manipulator Position Computed Torque Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate Nonlinear Control and MATLAB Courses”, International Journal of Robotics and Automation, vol. 3, no. 3, (2012), pp. 167-191.
- [18] F. Piltan, M. Eram, M. Taghavi, O. R. Sadrnia and M. Jafari, “Nonlinear Fuzzy Model-base Technique to Compensate Highly Nonlinear Continuum Robot Manipulator”, IJISA, vol. 5, no. 12, (2013), pp. 135-148. DOI: 10.5815/ijisa.2013.12.12.

