Proceedings of the IEEE
International Conference on Mechatronics & Automation
Niagara Falls, Canada • July 2005

Flexible Fuzzy Logic Control for Collision-Free Manipulator Operation

Martin Soucy and Pierre Payeur
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
[msoucy & ppayeur]@site.uottawa.ca

***Abstract* - Robot path planning is an important part in 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 produce collisions with surrounding obstacles. This paper presents an original and robust solution that can be used for safe path planning for various manipulator arms working in cluttered environments. Path planning is executed in two steps. First, a global path is determined to guide the end effector using classical artificial potential fields, probabilistic datasets and multi-resolution occupancy grids. Then, a fuzzy logic controller performs a local tracking for the entire robot structure by considering the kinematics of the robot as well as repulsive forces originating from nearby obstacles. Experimentation with robots of various architectures and different environments demonstrated the robustness and flexibility of the proposed planning and control approach which requires minimal tuning.**

Index Terms - path planning, collision avoidance, fuzzy logic, potential fields, manipulator robots.

I. INTRODUCTION

Robotic systems have significantly changed the way goods are manufactured. But for many reasons, such as safety and economics, it is desirable to develop robots able to perform more complex tasks with minimal or no human intervention. Numerous methods have been proposed to achieve collision-free path planning, mainly for mobile robots. However, in the case of manipulator arms, the structure of the robot adds to the complexity of the solution. Although many algorithms have been put forth, they do not provide an approach that is independent of the manipulator's configuration and cannot deal with arbitrary environments.

The goal of this research is to explore ways to develop a robust solution for collision-free path planning working on generic robot manipulators. An original technique is designed to work in a cluttered environment for an arbitrary manipulator configuration while keeping computation time short such that the advantages of automating the process are not compromised by a heavy computational workload.

The solution that is proposed and evaluated consists of having an initial path for the end effector computed with classical attractive potential fields. The use of probabilistic occupancy maps with multiple resolution levels makes this initial planning phase more efficient by reducing the computation of attractive forces and lowering the number of steps along the trajectory.

Once the end effector trajectory is determined, the whole robot structure must safely travel through free space at all time while keeping the end effector on its path. This phase presents an important challenge since when a manipulator arm moves the entire structure needs to overcome the complex interactions of the structure with the environment. Explicit control approaches determine the movement of each manipulator component via the inverse kinematic (IK) set of equations. Unfortunately, solving IK equations often leads to complex or even undefined closed-form solutions and differs greatly for various architectures. Many valid solutions are also possible that must be sorted and evaluated. In order to bypass the expensive computation of IK equations, an original fuzzy logic controller is proposed that combines the current robot configuration and the effect of repulsive forces generated from surrounding obstacles. In this manner, a safe trajectory for the entire robot structure can be determined that makes the robot follow the predefined end effector path while minimizing joints movement and avoiding collisions.

The following sections present a review of literature of the classical approaches that inspired this work and detail the proposed two-phase path planner including the design of a fuzzy logic control scheme that overcomes the inherent complexity of manipulators inverse kinematic's explicit solution. Experimental results are presented and analyzed for various robots and environments.

II. REVIEW OF LITERATURE

Potential fields have taken an important place over the last decade in robot path planning research as they offer relatively simple and efficient approaches for collision avoidance. Typically, potential fields are made up of an attractive field, which directs the robot towards its objective, and a repulsive field that pushes the robot away from obstacles. Many alternatives have been proposed to reduce the inherent limitation of potential fields, that is the occurrence of local minima [1, 3, 4, 5, 9]. Other original strategies propose to guide robots by emulating various physical phenomena [12, 13, 17]. However, in general these approaches do not offer a solution that can be used with generic robot architectures.

A large part of the literature also presents research performed on controlled environments which are typical of industrial settings. But the development of autonomous robotics requires that systems can operate in arbitrary workspaces that can be monitored via occupancy maps with progressive refinements [7, 8, 11]. This aspect is rarely considered.

Moreover, even though numerous approaches are dedicated to mobile robot path planning, manipulators have not been the focus of as much research. This fact results mainly from the difficulty of controlling the entire structure of a manipulator which moves and interacts with itself and the surrounding environment as well as from the inherent complexity of the inverse kinematic model.

In order to overcome the complex computation of an explicit inverse kinematic solution that significantly differs between manipulators, Nedungadi [10] proposes an original fuzzy logic robot controller that combines end-effector's velocity and required position changes as fuzzy inputs. A rule base infers the necessary joints displacement that result in the desired end-effector movement. The goal of the algorithm is to track a pre-defined end-effector path using the proposed fuzzy logic controller. However, the approach has been applied only on planar manipulators operating in empty environments where collision avoidance is not considered.

Beheshti and Tehrani [2] propose an adaptive fuzzy logic strategy to overcome solving the inverse kinematic problem while taking into account collision avoidance with surrounding obstacles. When the manipulator enters threshold regions which are previously defined around objects, the fuzzy rule base is dynamically modified to cause the manipulator to leave the region. This approach appears to be flexible enough to be adapted to various manipulators. However, since the obstacle information is not directly interpreted as a fuzzy input, the algorithm is less reliable to determine a safe and optimized path and tends to lead to lengthy computation times.

Tian and Mao [16] combine fuzzy logic and neural networks for the solution of the inverse kinematics of manipulator arms. The fuzzy controller provides feedback control signals for the manipulator while modeling the robot arm dynamics. Placed in a feed-forward configuration, a dynamic recurrent neural network models the inverse dynamics of the manipulator. The neural network includes a hidden layer made of non-linear neurons and an output layer consisting of linear neurons. The non-linear neurons capture the non-linearity of the input function, such as the inverse kinematics of the robot. The linear neurons approximate the input function within an arbitrary range. This setup offers an increase in performance and stability of the control system but requires training of the neural network each time the kinematic equations are changed.

The approach introduced by Nedungadi [10] offers a simple method to control a planar manipulator arm. Since the setting of the fuzzy logic controller only requires the forward kinematic equations, which are easy to determine, this approach is well suited for use with any robot. The present work proposes a generalization of Nedungadi's fuzzy logic control scheme to include the effects of repulsive forces originating from nearby obstacles when the end effector follows a path determined with potential fields. The approach is also extended to 3D space and results in a robust strategy that can be used with any configuration of a n degree-of-freedom manipulator while ensuring collision avoidance in a cluttered environment.

III. Path Planning and Robot Control

Following traditional path planning procedures to make the robot move safely across the workspace, a sequence of configurations is computed in two steps: *i)* a global planning phase is performed to determine a path for the end effector only using a multi-resolution probabilistic occupancy model [15], then *ii)* a tracking phase determines a proper sequence of robot arm configurations that makes the end effector follow the precomputed trajectory while avoiding contact with obstacles over the entire manipulator's structure.

A. End-Effector Path Planning

The first path computed corresponds to an approximation of the final trajectory of the end effector. This path is determined using a multi-resolution occupancy grid [15] of a cluttered environment in which regions of similar occupancy states are grouped to form wide uniform areas. A discrete attractive potential field value is computed for each cell having an occupancy level lower than a given threshold, starting from the target position for the end effector [9]. Following standard approaches used for mobile robots, the end effector is approximated as a single point to be moved following the negative gradient of the potential field between its current configuration and the target. To minimize the computational workload that results from this global path planning phase, large uniform cells that appear in wide empty corridors are favored as they provide larger displacements with a minimum search effort and tend to keep the end effector at a safe distance from obstacles.

The attractive field is computed starting from the cell which covers the "start" position, corresponding to the initial end effector's configuration, using a neighbor search technique between adjacent cells [14]. New cells are selected for each step by taking the neighbor cell with the smallest distance to the target position. The resulting global path for the end effector is a sequence of cells linking the start and target positions. The path of the effector is then interpolated to pass through the center of the selected cells and supplementary intermediate positions are computed to link those cells, and reduce the size of the steps, either through common corners or smallest distance to the neighboring cell. Figure 1 illustrates the path of the end effector through a series of cells with different resolution levels.

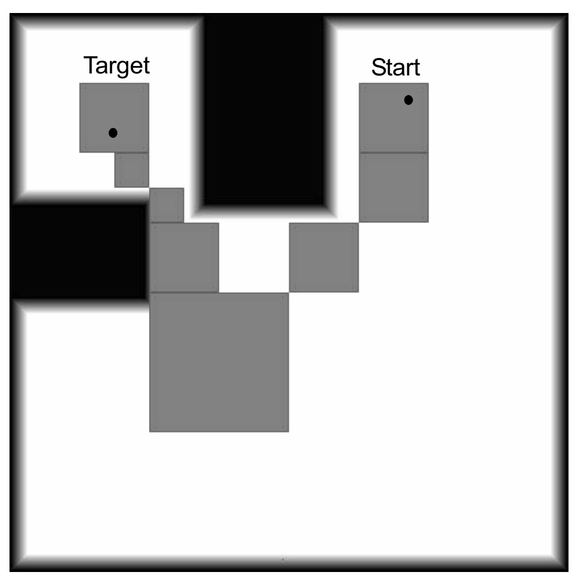

Fig. 1. Global path planning of the effector on a multi-resolution map.

B. Fuzzy Logic Control for Path Tracking

The fuzzy logic path tracker iteratively determines the joint parameters of the manipulator that satisfy a given position of the end effector to be reached. It is designed to achieve the following objectives:

- To separate the global end effector path into intermediate steps such that joint movements are minimized and the occurrence of singular movements is reduced;
- To determine joint configurations that minimize the angular changes required to reach a given position in order to achieve a smooth trajectory;
- To include the effect of repulsive forces exerted by nearby obstacles to ensure collision avoidance;
- To keep internal parameters of the fuzzy controller constant or auto-adapted for every scenario.

The algorithm seeks to update the joint angle values based on the target position considering the current state of the system. Instead of using absolute values for the position, the algorithm considers the required displacement, that is how far does the end effector need to move to reach the next desired position. This makes the control scheme independent of the magnitude of the global displacements and environment configuration.

The state of the robot is also characterized by the end effector velocity, which corresponds to the magnitude of movement of the end effector for a unit change in a joint angle. This is computed by partial derivatives of the forward kinematic equations of the manipulator. Hence, instead of explicitly solving the inverse kinematics, this approach only requires the explicit forward kinematics representation, which is easily obtained for any manipulator following the Denavit-Hartenberg convention [6], and its first derivative. Figure 2 shows the input and output membership functions of the proposed fuzzy logic controller. Their distribution is auto-adaptive based on the size of the environment.

The fuzzy rule base proposed by Nedungadi [10] provides a good basis for robot manipulator control, but important limitations were observed through experimentation. Most notably, from the fuzzy representation of the end effector speed, only a change in the sign generates a variation in the joints' movement. Such a control scheme cannot be used for collision avoidance. Therefore, the rule base has been refined such that a change in the magnitude of the effector speed can also influence joints' movement. Refinement was also made to address weaknesses observed on the respective magnitude of displacement of the joints with respect to the magnitude of the required end effector displacement and current velocity.

As a result, a new fuzzy rule base is proposed as shown in Table 1. Experimental comparison with Nedungadi's control scheme demonstrated faster convergence of the manipulator toward the target position in the presence of obstacles when the new rules are used and similar performance in empty space. The new rules also improve the representation of the kinematics of the robot as displacement and velocity magnitudes are now taken into account. Finally rules are defined to ensure that joint movement is limited when the speed of the end effector is high in order to minimize the risk of collisions.

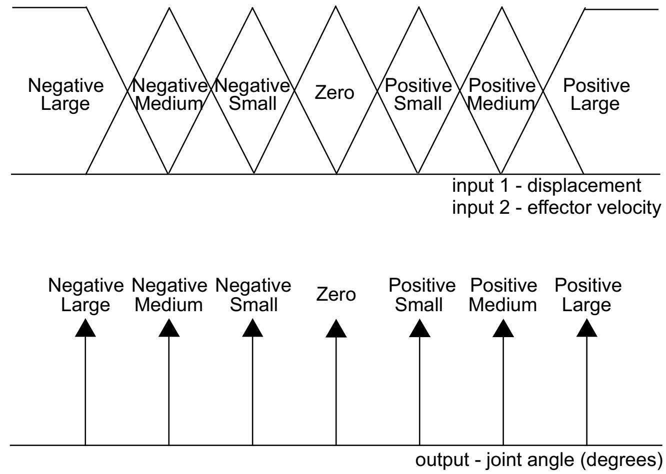

Fig. 2. Input/output fuzzy membership functions of the path tracker.

		Effector's displacement required						
		NL	**NM**	**NS**	**Z**	**PS**	**PM**	**PL**
Velocity of effector for positive joint change	**NL**	PM	PS	PS	Z	NS	NS	NM
	NM	PL	PM	PS	Z	NS	NM	NL
	NS	PL	PM	PM	Z	NM	NM	NL
	Z	Z	Z	Z	Z	Z	Z	Z
	PS	NL	NM	NM	Z	PM	PM	PL
	PM	NL	NM	NS	Z	PS	PM	PL
	PL	NM	NS	NS	Z	PS	PS	PM

Table 1. Refined fuzzy rule base for the path tracker.

When applied on a serial manipulator, the fuzzy logic inference procedure iterates on every joint along the structure starting from the robot base for a given displacement of the end effector. The robot configuration is progressively refined by supplementary iterations until the target position for the end effector is reached within a given error margin or until a maximum number of iterations is reached indicating that a solution for the desired end effector configuration is not possible given the robot architecture and the current environment configuration.

The internal structure of the proposed fuzzy controller with collision avoidance is detailed in Figure 3. The first input represents the displacement required for the end effector to reach the next target position along the global trajectory. It is encoded as dx and dy in Cartesian coordinates. In 3D workspaces, a third displacement along the Z axis can be added and computed similarly. The second input provides the end-effector velocity for a unit change in each of the joint angles as determined by a partial differentiation of the forward kinematic equations around the current robot configuration for each joint (V_{Xi}, V_{Yi}). For example, considering a planar robot with 4 degrees of freedom (DOF), there are 8 pairs of inputs ([dx] with [V_{X1}, V_{X2}, V_{X3}, V_{X4}] and [dy] with [V_{Y1}, V_{Y2}, V_{Y3}, V_{Y4}]) to compute. The joint angular correction for each DOF is iteratively calculated for each combination until the desired configuration is reached.

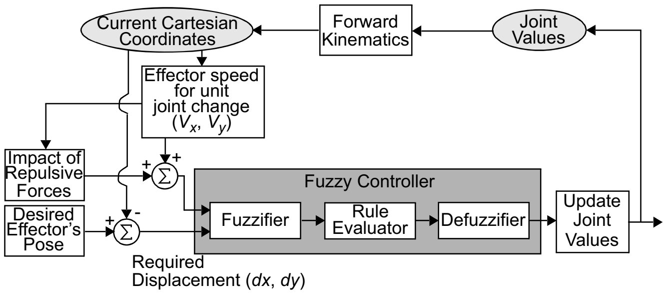

Fig. 3. Fuzzy path tracker structure with repulsive force feedback.

C. Collision Avoidance

A significant addition in the proposed control scheme is to directly consider the effects of the environment on the entire robot structure while moving the joints. To achieve this, a supplementary component monitoring repulsive forces generated by close obstacles is introduced in the fuzzy logic control loop under the form of an extra input to the fuzzy inference engine, as shown in Figure 3. In order to include the influence of repulsive forces, the end-effector velocity value is modified if a part of the manipulator is close to an obstacle such that its movement is changed accordingly. As the robot comes closer to an obstacle, the repulsive force progressively makes one joint move in another direction to avoid an imminent collision. In turn, given the mechanical constraints of the robot structure, the other joints overcome this change by readjusting in the following iteration or by exploiting the redundancy of the manipulator, if available.

The repulsive force, (F_x, F_y), for any point in space is computed to be proportional to: *i)* the probability of occupancy, $p[occ]$, which is discretized on 16 levels (0-15) as provided by the probabilistic occupancy map that is used for path planning [7], *ii)* the orientation, θ, of the repulsive force with respect to the x-axis which determines the direction of the force applied on the robot, and *iii)* a force coefficient, $coeff_{rep\_force}$:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} \frac{p[occ]}{3+p[occ]} \cdot coeff_{rep\_force} \cdot \sin(\theta) \\ \frac{p[occ]}{3+p[occ]} \cdot coeff_{rep\_force} \cdot \cos(\theta) \end{bmatrix} \quad (1)$$

where $coeff_{rep\_force}$ provides a threshold at which a collision is imminent and for which the joint should move in the opposite direction. It is experimentally calibrated to a value equal to 2.0 to ensure that the robot will be pushed away sufficiently if close to an obstacle, while not creating large oscillations or significantly reducing the work space of the robot.

The strongest repulsive force occurs where the probability of occupancy of the space covered by any point on the robot structure is the highest. In order to avoid collisions and to minimize joints displacement when a collision is imminent, the repulsive forces and the joint velocities are combined to provide a proper input to the controller. When the repulsive force indicates an imminent collision, the value of the velocity is progressively reduced until it provokes a sign change, therefore causing the joint to move in the opposite direction to prevent the collision. To achieve this, the local probability of occupancy is verified from the environment map for the entire structure of the robot at every step of the fuzzy iterative process. As a result, collisions are continuously monitored.

The components of the corrected fuzzy input for joint i with end effector velocities of V_x and V_y when the components of the repulsive force are F_x and F_y and the highest probability of occupancy occurs on joint j is defined by:

$$\begin{bmatrix} V_{Xi\_corrected} \\ V_{Yi\_corrected} \end{bmatrix} = \begin{bmatrix} V_{Xi}\left(1 - \frac{i}{j}F_x\right) \\ V_{Yi}\left(1 - \frac{i}{j}F_y\right) \end{bmatrix} \quad (2)$$

Experimentation demonstrated that making correction on joint displacement only for the part of the structure where $i \le j$ rather than on all joints was the most appropriate approach to avoid collisions while preserving smooth trajectories. This results in applying repulsive forces only on joints located between the base of the robot and the joint which receives the strongest repulsive force as illustrated in Figure 4.

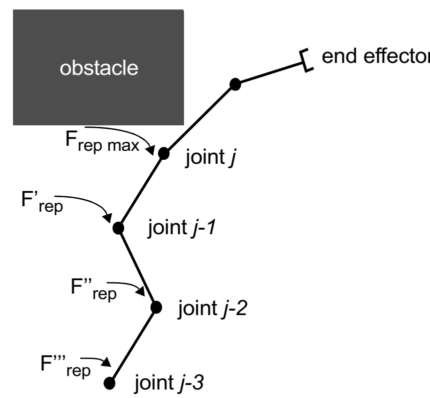

Fig. 4. Application of repulsive forces on robot joints.

IV. Experimental Results

In order to evaluate the generality of the proposed solution, the algorithm has been implemented and tested on numerous manipulator architectures. Given the flexibility of the controller, only the forward kinematic parameters of the robot and their first partial derivatives are encoded as a series of equations providing the necessary relationships.

The workspaces used for experimentation consist of environments usually containing a "safe" zone around the robot base and a number of obstacles elsewhere. The obstacles are either placed randomly or in specific positions to evaluate the effects of particular situations. Various tests were conducted to explore the behavior of the algorithm when facing situations such as obstacles in close proximity of robot, narrow corridors and moving the manipulator around an obstacle. Environments are mapped as probabilistic datasets encoded as greyscale images (black being 100% occupied space and white being 0% occupied). The typical size of the environments considered is 512x512 cells for 2-D maps.

Figure 5 presents the resulting trajectory for a 4-DOF planar manipulator having the end effector move from the right side of an obstacle up to a location contained in an enclaved zone between two obstacles. Here, the use of multi-resolution cells is put in evidence as the final part of the end effector trajectory is made of smaller steps. The trajectory of the end effector is first computed as detailed in section III.A and results in a sequence of intermediate points to be reached. The global path of the end effector is then refined with the fuzzy logic-based path tracking approach presented in section III.B to obtain the final displacement of the entire manipulator. From the starting robot configuration, the local path tracking method is executed to determine a series of intermediate robot configurations by subdividing the global path into small intermediate segments for which the fuzzy controller is optimized. In this example, each segment represents 5 cells of the occupancy map or about 1% of the total workspace. A series of joint angles is then computed iteratively to successively traverse those segments and avoid collisions.

The angular evolution for each of the 4 joints is shown in Figure 6. These curves demonstrate that the fuzzy controller minimizes the joints' movement between consecutive positions. This way, stability is preserved and the risk of collisions that often results from abrupt displacements is minimized. Also, the occurrence of singular behavior is greatly reduced. In this example, the largest joint displacement performed between two consecutive positions is approximately 4.5 degrees.

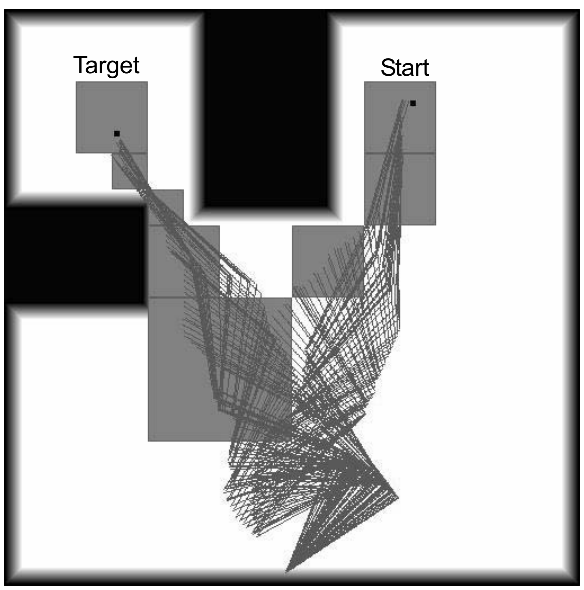

Fig. 5. Complete collision-free path for a 4-DOF planar manipulator.

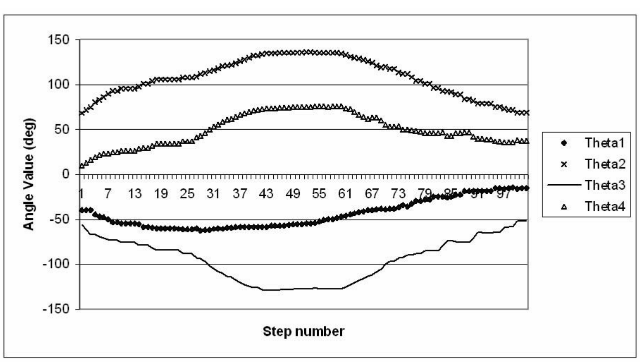

Fig. 6. Angular evolution for each joint along the trajectory.

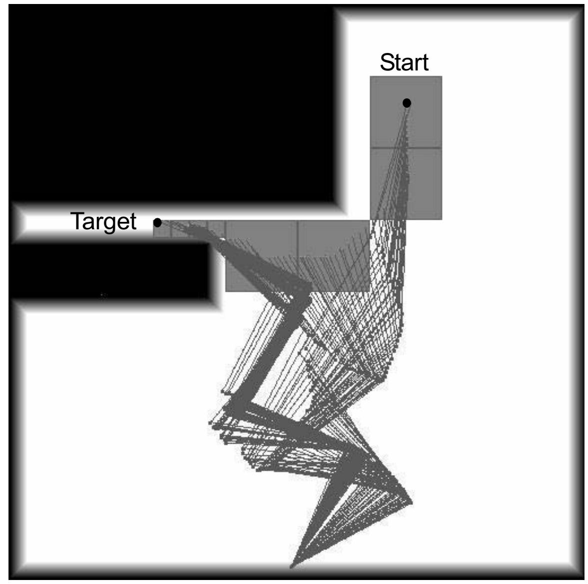

Fig. 7. Collision-free path for a manipulator entering a narrow corridor.

Figure 7 presents another example where a 4-DOF planar manipulator starts from an upright position and must reach a point inside the narrow corridor on the left. The same global path planner and fuzzy logic path tracker are used in this case without any adjustment of their parameters. The resulting trajectory shows that the controller succeeds to re-orient the end effector when it approaches the target position such that a safe entry in the narrow corridor is achieved. In the final part of the trajectory the computation and convergence rate of the controller slows down but keeps safely pushing the robot toward its goal. This example demonstrates the robustness of the proposed algorithm.

Experimentation has been conducted on numerous other scenarios and demonstrated the flexibility of the algorithm to deal with different manipulator arm architectures, being redundant or not. The results also demonstrated the adaptability of the algorithm to find a solution in cluttered environments at various levels of complexity. As demonstrated in the previous examples, concavities created between objects do not deteriorate performance since potential fields automatically adapt to space configuration and adjust repulsive forces accordingly.

The main benefits of the proposed fuzzy logic controller have been put in evidence as the approach does not need to be tuned for each manipulator. Parameters of the fuzzy logic path tracker are defined to automatically scale membership functions following the dimensionality of the environment. The fuzzy rule base remains the same despite changes in the forward kinematic models, which is the only set of parameters to be provided for each robot.

Using the current C implementation of the proposed path planner and fuzzy logic control scheme, execution times appear to be fast enough for on-line operation of robots that perform relatively slow movements as found in numerous manufacturing applications. For the two cases presented above, computation times are respectively 17 and 20 seconds when run on a SGI 320 workstation equipped with a PIII-450 MHz and 218 MB of RAM. This includes all steps from the input/output of data, the computation of potential fields, the neighbor search operations, the global path planning and the final path tracking.

The proposed approach has also been fully extended to operate in 3D space with n-DOF manipulators and demonstrated excellent performance with these more complex robots and workspaces. In terms of computation time, an average increase of 50% has been observed when dealing with 3D occupancy models and non-planar robots in comparison with the results presented above. However collisions were still successfully avoided while reaching the desired goal with the end effector.

V. Conclusion

This work presents an approach for collision-free path planning for robot manipulators. An original solution combining discrete potential fields and fuzzy logic is presented and analyzed. A flexible algorithm able to work with various manipulator architectures and environment configurations is introduced. Two main phases are used: a global path planning phase determines an optimal path for the end effector only and a local path tracking phase configures the rest of the robot arm to make the end effector follow the pre-computed sequence of positions while avoiding collisions over the entire manipulator structure.

The complex problem of manipulator's inverse kinematics is overcome through the use of a fuzzy logic controller which eliminates the need for determining and computing the complex equations of inverse kinematic models or the use of heuristic methods. The proposed algorithm is well suited for a large range of manipulator architectures and requires minimal tuning of the controller. Only the forward kinematic model and its first derivatives that differ between robot architectures need to be provided to the path planner.

The fuzzy logic approach proposed by Nedungadi has been extended to consider the presence of obstacles that generate repulsive forces acting on the manipulator structure, and generalized to include 3D robot architectures and workspaces. Experimentation demonstrated that the use of fuzzy logic allows to keep joint movements small between consecutive positions when traveling in a cluttered environment while preventing collisions and keeping control on singular configurations. The approach can be extended to operate in dynamic environments provided that a multi-resolution occupancy grid mapping moving objects is made available.

Acknowledgments

The authors wish to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) that made this research work possible.

References

[1] J. Barraquand and J.C. Latombe, "A Monte-Carlo Algorithm for Path Planning with Many Degrees of Freedom", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1712-1717, Cincinnati, OH, 1990.

[2] M.T.H. Beheshti and A.K. Tehrani, "Obstacle Avoidance for Kinematically Redundant Robots Using an Adaptive Fuzzy Logic Algorithm", in *Proceedings of the American Control Conference*, pp. 1371-1375, San Diego, CA, 1999.

[3] S. Caselli, M. Reggiani and R. Sbravati. "Parallel Path Planning with Multiple Evasion Strategies", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 260-266, Washington, DC, 2002.

[4] H. Chang, "A New Technique To Handle Local Minimum For Imperfect Potential Field Based Motion Planning", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 108-112, Minneapolis, MN, 1996.

[5] 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 the IEEE International Conference on Robotics and Automation*, pp. 983-988, San Francisco, CA, 2000.

[6] J.J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd edition, Pearson Prentice Hall, NJ, 2005.

[7] A. Elfes, "Using Occupancy Grids for Mobile Robots Perception and Navigation", in *IEEE Computer*, vol. 22, no. 6, pp. 46-57, 1989.

[8] Y. Kitamura, T. Tanaka, F. Kishino and M. Yachida, "3-D Planning in a Dynamic Environment Using an Octree and an Artificial Potential Field", in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, vol. 2, pp. 474-479, 1995.

[9] T. Laliberté and C.M. Gosselin, "Efficient Algorithms for the Trajectory Planning of Redundant Manipulators with Obstacle Avoidance", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2044-2049, San Diego, CA, 1994.

[10] A. Nedungadi, "A Fuzzy Robot Controller – Hardware Implementation", in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1325-1331, San Diego, USA, 1992.

[11] P. Payeur, D. Laurendeau and C.M. Gosselin, "Range Data Merging for Probabilistic Octree Modeling of 3-D Workspaces", in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3071-3078, Leuven, Belgium, 1998.

[12] M. Piaggio and A. Sgorbissa, "AI-CART: an Algorithm to Incrementally Calculate Artificial Potential Fields in Real-Time", in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 238-243, Monterey, CA, 1999.

[13] L. Sing, H. Stephanou and J. Wen, "Real-time Robot Motion Control with Circulatory Fields", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2737-2742, Minneapolis, MN, 1996.

[14] M. Soucy, *Manipulator Path Planning with Multi-Resolution Potential Fields and Fuzzy Logic Control*, Master's thesis, University of Ottawa, March 2005.

[15] M. Soucy and P. Payeur, "Robot Path Planning with Multi-resolution Probabilistic Representation: A Comparative Study", in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1127-1130, Niagara Falls, ON, 2004.

[16] L. Tian and Z. Mao, "Fuzzy Neuro Controller for a Two-Link Rigid-Flexible Manipulator System", in *Proceedings of the International Conference on Neural Information Processing*, pp. 1867-1871, Singapore, 2002.

[17] X. Yun and K.-C. Tan, "A Wall-Following Method for Escaping Local Minima in Potential Field Based Motion Planning", in *Proceedings of the IEEE International Conference on Automation and Robotics*, pp. 421-426, Monterey, CA, 1997.