

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Soft-computing Techniques for the Trajectory Planning of Multi-robot Manipulator Systems

Emmanuel A. Merchán-Cruz¹, Alan S. Morris² and Javier Ramírez-Gordillo¹

¹*Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Azcapotzalco, IPN*

²*Automatic Control & Systems Engineering Dep., The University of Sheffield*

¹México, ²United Kingdom

1. Introduction

The planning of the motion of robots has become increasingly more complex as robots are used in a wide spectrum of applications, from extremely repetitive tasks in assembly lines to assistance in delicate surgical procedures (Tombropoulos et al., 1999; Coste-Maniere et al., 2003). Whichever scenario, the planning has to consider the fact that the robot will be interacting with other elements in its environment avoiding collision with other objects, whether these remain static or in motion, while executing a given task.

In planning the motion for robots, it is a common misassumption that path planning and trajectory planning are synonymous. Motion planning, as defined by Sugihara and Hwang in (Hwang & Ahuja, 1992; Sugihara & Smith, 1997), is subdivided into path planning and trajectory planning. (Fraichard & Laugier, 1992) state the following distinction: path planning is the search of a continuous sequence of collision-free configurations between a start and a goal configuration, whereas trajectory planning is also concerned with the time history or scheduling of this sequence of configurations as well.

Considering that the motion of the elements that form the kinematic chain of robot manipulators is described by a system of non-linear equations that relate the motion in the Cartesian space of the end-effector as a consequence of the individual variations of the links of the manipulator due to the angular displacements at each joint. When solving for a particular position of the end-effector in the Cartesian space, the inverse kinematics problem, a set of configurations has to be calculated to position the tip of the manipulator at that desired point. Due to the natural dexterity of robot manipulators, the space of solution is non-linear and multidimensional, where more than a single solution exists to solve a particular point in the Cartesian space and choosing the appropriate solution requires an optimisation approach.

Taking this into consideration, the solution of the motion planning problem of robot manipulators is an ideal candidate for the use of soft-computing techniques such as genetic algorithms and fuzzy logic, as both approaches are known to perform well under multidimensional non-linear spaces without the need for complex mathematic manipulation to find a suitable solution (Zadeh, 1965; Mamdani, 1974; Goldberg, 1983; Bessiere et al., 1993; Doyle, 1995; Doyle & Jones, 1996).

Following this line of thought, the application of soft-computing techniques is considered to improve the performance of trajectory planning. The approach followed to solve this problem considers that the manipulators have to reach a specified goal or target defined in coordinates of their workspace, rather than a goal or target configuration as widely presented in numerous papers. This is because when the manipulator is forced to reach a certain configuration, the trajectory planning problem is greatly simplified by constraining the system to satisfy those specific values. By specifying the target in coordinates in the workspace instead, the system can solve for this condition assuming a number of different solutions, thus, dealing now with an optimisation problem.

1.1 Motion planning

Motion planning for robot manipulators has been extensively studied during the last two decades. A comprehensive survey on the common techniques used to solve this problem can be found in (Latombe, 1991; Hwang & Ahuja, 1992; Ata, 2007). Depending on the nature of the problem, some authors classify the motion planning problem in two categories: global and local motion planning (Lozano-Peréz, 1983; Latombe, 1991; Althoefer, 1996).

Global motion planning requires a complete description of the workspace of the robot where all obstacles are clearly identified before searching a path. This takes place in systems where the manipulators operate in a highly structured and controlled environment in which all possible obstacles are static and known in advance; the planning here is performed off-line allowing for an optimal trajectory to be found.

On the other hand, local motion planning calculates iteratively the next best position for the robot that best reduces the error to the goal while avoiding collision with any obstacles in the workspace. Conversely to global approaches, a local approach does not require prior knowledge of the system in terms of possible obstacles in the workspace. Local motion planning approaches have to process information which describes the vicinity of the manipulator and modifies its trajectory in order to avoid collision with any obstacle nearby while minimising the error to the target. Local planners are commonly used in dynamic environments where limited information on moving obstacles is available (Brooks, 1986; Lee, 1996; Liu, 2003).

The algorithms for motion planning are often categorised as either complete or heuristic (Chen & Hwang, 1998; Masehian & Sedighizadeh, 2007). Complete algorithms can take a considerable amount of computation time before finding a solution if there is one, or fail to prove that there is no solution. Heuristic algorithms are fast but are not guaranteed to find a solution even if there is one (Eldershaw, 2001; Isto, 2003). Their attractiveness resides in the fact that they will either find a solution or fail in a space of seconds.

The development of the configuration space method or C-space by (Lozano-Peréz & Wesley, 1979) marked a standard which has been used in various path-planning algorithms, as can be seen in (Wise & Bowyer, 2000). In the C-space, each node represents a possible configuration of the robot, and a path is specified by following a line that connects feasible configurations from a start to a goal configuration. Prior to this, all path planning methods were dependent on the use of the Cartesian space to represent a manipulator and its environment. The configuration space approach reduces the problem to that of moving a point through an n -dimensional space, where n corresponds to the number of degrees of freedom (dof) of the manipulator. Obstacles in the manipulator's environment are mapped into the C-space and are represented as a forbidden region known as configuration space

obstacles or CS-obstacles where a collision with the manipulator occurs. A typical technique to navigate in the C-space consists of exploring a uniform grid in C-space using a best-first search algorithm guided by a goal-oriented potential field (Barraquand & Latombe, 1991).

1.2 The Multi-mover's problem

When there is more than one robot working in a common workspace, the planning of motion of such a system is referred to as the multi-movers problem. The multi-movers problem is that of finding paths for the robots between their initial and final configurations whilst avoiding one another as well as any obstacles.

Regarding the motion planning into systems where multiple robots share the same environment, three approaches can be identified: the centralized approach, the hierarchical or prioritized and decoupled planning.

The centralized approach consists of treating the various robots as if they were one single robot, considers the Cartesian product of their individual C-spaces called composite C-space. The forbidden region is the space in which one robot intersects an obstacle or two robots intersect each other. Algorithms based on this approach require a great amount of computational resources to store the resulting information of the representation of the composite C-space and for solving the path over this space (Schwartz & Sharir, 1983; Fortune et al., 1987).

The hierarchical or prioritized approach presented by Freund and Hoyer in (Freund & Hoyer, 1985), plans the motions of the robots accordingly to a certain priority assigned to each robot depending on their specified tasks. A common configuration under this approach in multi-robot arm systems is the Master-Slave configuration, where the motion of the slave is dependent on the motion of the master, either to avoid collision with the master in pick-and-place like tasks, or to assist in the manipulation of a common object.

In the decoupled planning, the paths for each robot are planned independently and are later synchronized by scheduling the robots' motion to ensure that no collision takes place while executing the desired task. Lee and Lee (Lee & Lee, 1987), use this technique for a system consisting of two manipulators, where the speed of one of them is fixed while the speed of the other is modified in order to synchronize the previously planned paths, obtaining a collision-free interaction. The collisions here are studied using a bidimensional graph called "Collision map" where the path of the second robot is represented against time and collision regions are identified. The manipulators are modelled by spheres and the motion of the robots is restricted to straight-line paths. An extension of this method is presented in (Chang et al., 1994), where the robots are represented as polyhedral and the minimal delay-time value, necessary for collision free coordination, is determined.

The use of evolutionary techniques in the motion coordination of two manipulators is explored by (Ridao et al., 2001). As with other decoupled planning based approaches, the problem is broken into path planning, where collision-free paths for the robots are found independently and considering only fixed obstacles in the workspace; and trajectory planning where the paths are synchronized to ensure a collision-free interaction. Here an evolutionary algorithm gives an initial approximate solution in the C-space and from this solution a heuristic local search algorithm consisting of a monotonous random walk is used to find an optimum solution.

1.3 Soft Computing Techniques

Soft computing is a keyword in information technologies, and refers to a synthesis of methodologies from fuzzy logic, neural networks and evolutionary algorithms used to solve non-linear systems where conventional methods fail to provide a feasible solution.

As defined by Zadeh (Zadeh, 1997), “Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty and partial truth. ... At this juncture, the principal constituents of soft computing (SC) are fuzzy logic (FL), neural network theory (NN) and probabilistic reasoning (PR), with the latter subsuming belief networks, genetic algorithms, chaos theory and parts of learning theory. What is important to note is that SC is not a melange of FL, NN and PR. Rather; it is a partnership in which each of the partners contributes a distinct methodology for addressing problems in its domain. In this perspective, the principal contributions of FL, NN and PR are complementary rather than competitive”.

1.3.1. Genetic Algorithms

Developed by Holland (Holland, 1975), and further implemented by Goldberg (Goldberg, 1983), a genetic algorithm (GA) is a search strategy that mimics the theory of evolution of life in nature, belonging to the class of stochastic search methods. The main difference between GAs and other search methods is that, while most stochastic search methods operate on a single solution to the problem at hand, genetic algorithms operate on a population of solutions.

For any given problem to be solved by a GA, an initial population of possible solutions has to be created. These solutions are coded into Chromosomes of a fixed or variable length. The coding can be done in any representation although binary representation is commonly used. Each of the chromosomes are evaluated and assigned a fitness value accordingly to a fitness function. The basic operations in a GA are: Reproduction, Crossover and Mutation.

1.3.2. Fuzzy Logic

Proposed in (Zadeh, 1965) and successfully implemented for the first time in (Mamdani, 1974), fuzzy logic is an extension of Boolean logic which allows the processing of “vague” or uncertain information. Classic logical systems based on Boolean logic classify elements as members or not of a particular set, while Fuzzy-based systems can establish a degree of pretence of an element to any given set within a membership range between $[0, 1]$, providing a way to identify and rank an intermediate value. A membership value of zero indicates that the element is entirely outside the set, whereas a one indicates that the element lies entirely inside a given set. Any value between the two extremes indicates a degree of partial membership to the set. Fuzzy logic employs the classical set operations of union (OR), intersection (AND), and complementation (NOT), but their meaning varies from those in classic logic.

1.4 The Potential Field Approach

In order to avoid collisions between manipulators and other objects present within their workspace it is necessary to characterize these possible obstacles in a way that the information from this representation is of use to prevent any possible collision. The potential field approach, introduced in (Khatib, 1986), combines “attractive” and “repulsive” potential field to model the workspace and the obstacles therein. Figure 1 illustrates a line obstacle

and a goal at (10,10). As it can be appreciated, the global minimum of this combined field corresponds to the specified goal.

The attractive potential, P_a , is given by reducing the Euclidean error to the desired goal, while the repulsive potential, P_o , is given by:

$$P_o = \sqrt{Po_x^2 + Po_y^2} \quad (1)$$

subject to the following conditions:

$$\begin{aligned} \text{if } DO > (s+r) & \begin{cases} Po_x = 0 \\ Po_y = 0 \end{cases} \\ \text{if } r \leq DO \leq (s+r) & \begin{cases} Po_x = \beta(s+r-OD)\cos\phi \\ Po_y = \beta(s+r-OD)\sin\phi \end{cases} \\ \text{if } DO < r & \begin{cases} Po_x = m\cos\phi \\ Po_y = m\sin\phi \end{cases} \end{aligned} \quad (2)$$

where:

DO = distance to obstacle

s = distance of influence of the potential field of the obstacle

r = distance considered as imminent contact

Po_x, Po_y = x, y components of the Potential Field vector

β = scaling factor of the Potential Field for the influence zone

m = scaling factor of the Potential Field for the contact zone

ϕ = potential field direction

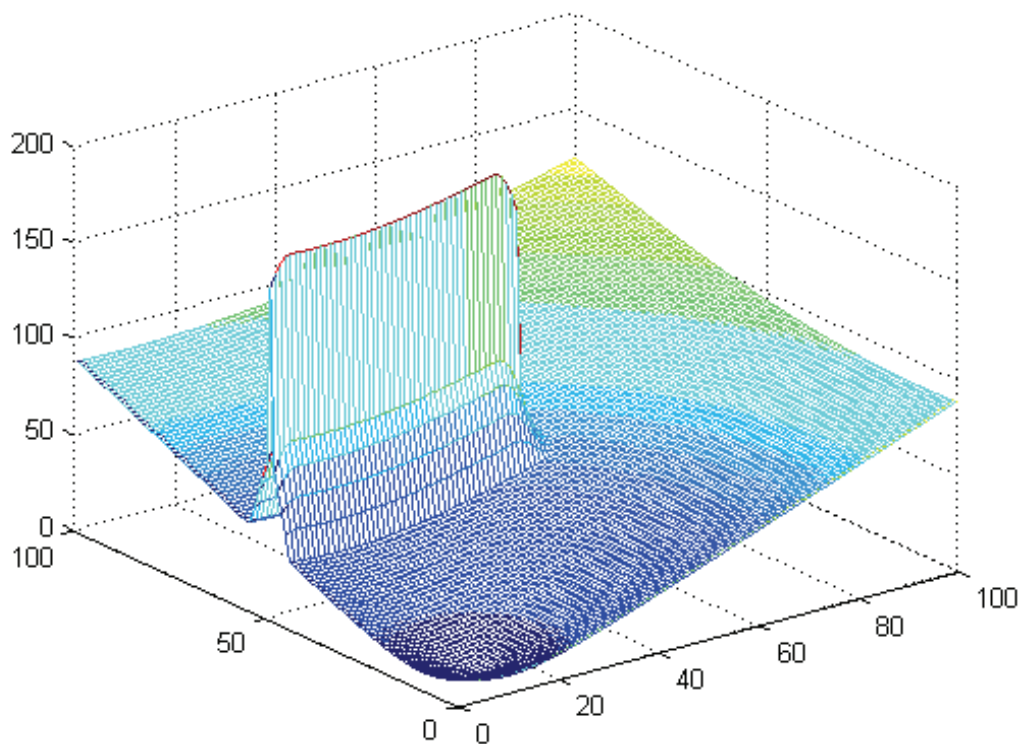


Figure 1. Combined potential field example

The resulting potential field P is obtained by the combination of these two potentials by:

$$P = P_a + P_o \quad (2)$$

This representation provides enough information to guide the manipulator, through a search algorithm, from a starting point to a desired final position. However, path planners based on this representation have to deal with the problem of local minima. To overcome the local minima problem different approaches have been suggested for both, the construction of the potential field and the search algorithm itself. An alternative in the construction of the potential field is the use of harmonic functions to build local minima free potential fields as proposed by Kim and Connolly (Connolly et al., 1990; Kim & Kolsha, 1992; Connolly & Grupen, 1993). The drawback of this approach for its practical implementation of path planners is that it requires the implementation of an iterative numerical algorithm for the solution of Laplace's equation, which results on a high computational cost.

The search algorithms used to navigate a potential field follow the gradient descent of the field until reaching the goal. A comprehensive summary of these methods can be found in (Latombe, 1991). A common characteristic of these methods is the addition of a procedure to escape local minima.

2. A GA based approach to path planning

Path planning of robots can either be carried out in the workspace or in the corresponding configuration space (C-space) of the manipulator. The C-space approach simplifies the problem to that of moving a point in a n -dimensional space where the point represents a particular configuration of the robot, n being equal to the number of degrees of freedom (dof's) of the robot. In order to avoid collisions when solving a path in the C-space, it is necessary to map the obstacles present in the workspace into C-space obstacles, and the actual mapping grows in complexity as the dimensionality of the C-space considered is higher. Before any mapping begins, it is necessary to determine the resolution of the C-space as its construction involves a sampling process. The amount of computational time and space required to process and store the C-space acts as a constraint on the implementation of algorithms based on this approach. Furthermore, if the obstacles change their position and/or orientation in the workspace, recalculation of the C-space is necessary.

On the other hand, path planning in the workspace requires appropriate obstacle representation and collision avoidance/detection strategies to allow the solution for a suitable path. Once a path has been determined, the computation of the inverse kinematics is required in the case of robot manipulators to find the suitable sets of configurations that are required to follow the specified path. At this point, it is necessary the implementation of a collision avoidance algorithm to ensure that the links of the manipulator will not collide with an obstacle while following the desired path.

In the path planning of robot manipulators, the C-space method requires the start and goal configuration of the manipulator to be defined. This reduces the complexity of the space of solution of the problem to that of a single possible solution, since the desired positions of the links of the manipulator are already known. Conversely, path planning in the workspace requires the initial configuration of the robot and of the goal, expressed in workspace coordinates, for the end-effector of the manipulator. The characteristic dexterity of robot

manipulators produces a solution space with multiple solutions depending of the number of dof's of the manipulator.

It follows that the motion planning of a robot manipulator can be classified as an optimisation problem, for which the path to be obtained is subject to different criteria of optimisation (such as path length, collision avoidance, etc.) while being subject to kinematic and/or dynamic constraints. This makes the path planning problem in robotics an ideal candidate for the implementation of genetic and/or evolutionary algorithms (Parker et al., 1989; Rana, 1996).

2.1 Problem specification

The C-space method previously introduced consists in solving the path planning problem in the configuration space of the manipulator. Being a configuration based approach, it is necessary to specify the goal as a desired configuration of the links of the manipulator thereby “reducing” the problem of that of moving a point through the C-space until the goal is reached. The major drawbacks of this method are the mapping process of the obstacles into the C-space which becomes a very time consuming process as the dimensionality of the C-space increases, the search for feasible paths in a high dimensional space and the large amount of computational resources to store the resulting C-spaces. Another disadvantage of C-space based algorithms is that they are only suitable if the obstacles are static or, in the case of moving obstacles, their trajectories are known in advance, since every time an obstacle changes its position or configuration it has to be re-mapped into the C-space.

Figure 2(a) illustrates a 2 dof manipulator and a static line obstacle. The corresponding C-space of this system is presented in Figure 2(b). The obstacle can either be modelled using the potential field approach in the workspace (Figure 3(a)) or it can be mapped into the C-space (Figure 3(b)).

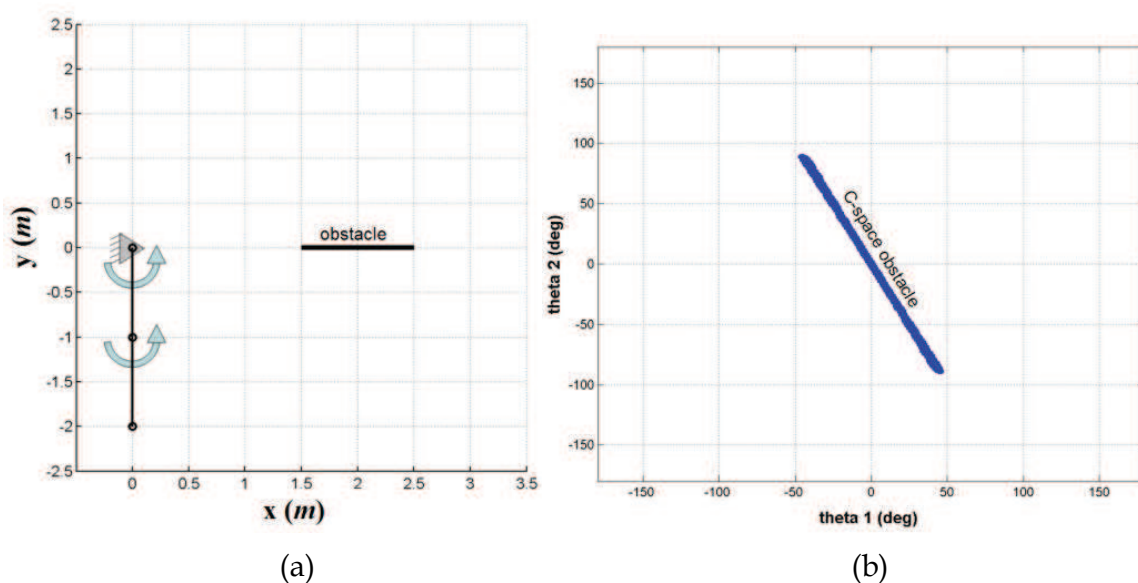


Figure 2. (a) 2 dof manipulator and static obstacle; (b) Mapped obstacle in the C-space

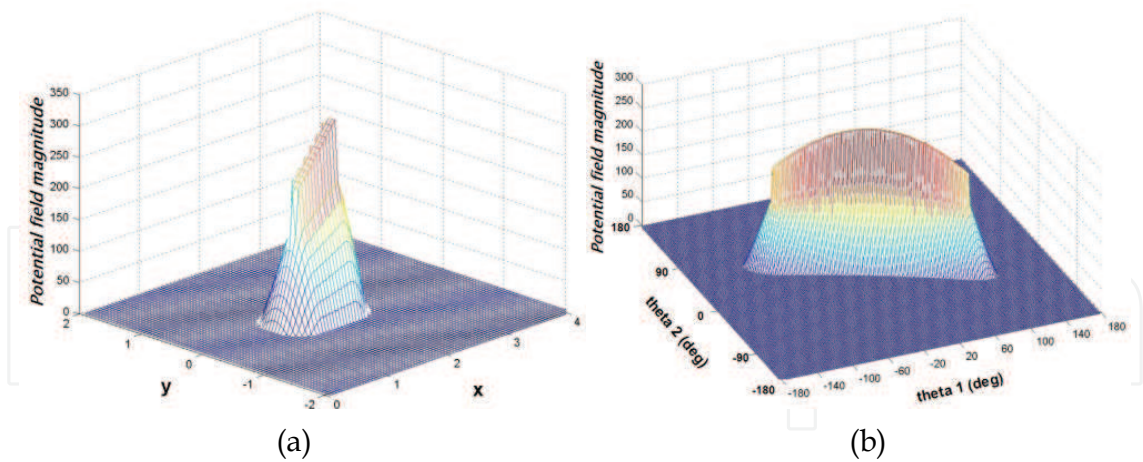


Figure 3. (a) Potential Field of the line obstacle; (b) Potential Field in the 2 dof C-Space

For example, if the manipulator in Figure 2(a) is desired to reach an arbitrarily chosen goal at (1.7, 0.5) in the workspace, the problem could be solved using a GA with a fitness function that reduces the error to the goal and is also affected by the magnitude of the potential field due to the presence of an obstacle. Equation 3 describes a possible fitness function where both requirements are met.

$$f_i = \frac{1}{Po + e^{error}} \tag{3}$$

where:
Po = Potential Field due to the Obstacle
error = Error to Goal

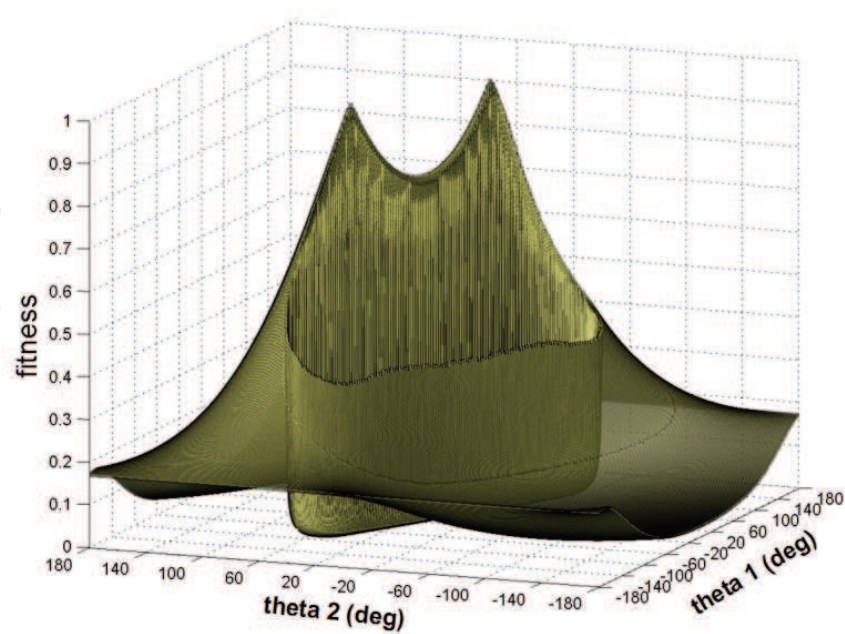


Figure 4. Fitness space for the 2 dof C-Space

The fitness space for the GA in the C-space is illustrated in Figure 4. It can be appreciated that, being a 2 dof system, two possible solutions to the problem exist which can be identified as the maxima of the fitness space. The fitness around the obstacle drops to zero. Figure 5(a) shows a 3dof planar manipulator and three static obstacles in its workspace. The C-space corresponding to this manipulator, Figure 5(b), is a three-dimensional space in which the obstacles are forbidden regions represented as 3d obstacles corresponding to those configurations where a collision with the obstacle occurs.

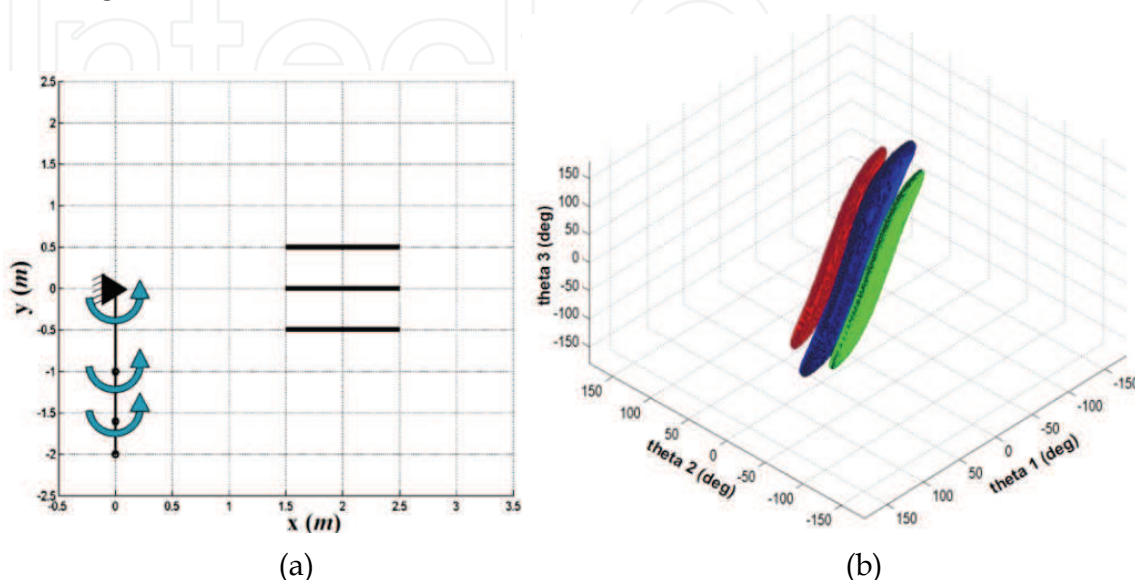


Figure 5. (a) 3dof Manipulator and static obstacles; (b) Obstacles in the 3dof C-space

The path planning using the potential field method in the C-space illustrated in Figure 5(b), requires tensor notation to represent the magnitude and direction of the potential field for each set of configurations that describe the obstacles in this three-dimensional space.

The problem of path planning of robot manipulators is an ideal candidate for the implementation of a GA based approach due the multidimensionality and inherited non-linearity of the system. A GA based approach for the solution of the problem directly in the workspace of the manipulator eliminates the mapping process of obstacles and the solution is open to any configuration that, free of collision, takes the manipulator to its desired goal rather than restricting the solution to a single configuration.

2.2 Genetic Algorithms and Fuzzy Logic in path planning

2.2.1. Genetic Algorithms based approaches

The application of GA's in path planning has previously been studied by some authors, from where two approaches can be clearly identified, those based on the definition and optimisation of paths throughout via points in the C-space, and those based on the "fitting" of a desired path in the workspace through GA optimisation of the manipulator's variables until the resulting path is closely fitted to the desired one. Methods based on either approach are considered global and the motion planning takes place off-line as a result of the number of computations involved.

A GA based path planner for a bi-dimensional space considering static obstacles is presented in (Doyle, 1995; Doyle & Jones, 1996). Real coded chromosomes are used to represent the coordinates of via points that define the segments of the possible paths with

the following structure: $[(x_1, y_1)(x_2, y_2) \dots (x_n, y_n)]$ where n is the total number of defined via points.

The population is evaluated by considering if the path segments are free from collision and the total length of the path. This algorithm is later adapted to the path planning on the C-space of a 2 dof manipulator with static obstacles by changing the chromosome structure to $[(q_{11}, q_{21})(q_{12}, q_{22}) \dots (q_{1n}, q_{2n})]$ being (q) the manipulator articulations and n the maximum number of configurations. This approach uses a single point crossover operator at the gene boundaries.

A similar approach for the path planning in the C-space of 2 and 3 dof manipulators is presented in (Rana, 1996; Rana & Zalzala, 1996). The problem is solved by generating an initial population of trajectories through randomly distributed via points. The GA then optimises the path taking into consideration the total length of the path, the uniform distribution of via points to ensure a “smooth” path and a penalty function when a via point or path segment lies within a C-obstacle.

A variation of this approach is presented in (Eldershaw, 2001), where the planning in the C-space is obtained by optimising the location of via points for a fixed number of path segments. As with the previous approaches, the solutions are evaluated taking into consideration whether a path is free from collisions in the C-space and the total length of the evaluated path. These algorithms use the same chromosome representation.

In a different approach for the path planning in the C-space, (Bessiere, Ahuactzin et al., 1993) propose a path planner that uses a local GA to define Manhattan motions to a series of sub-goals or landmarks. A second GA planner then attempts to connect the landmarks with the goal. In a system with n dof's a Manhattan motion, M , consists of moving each degree of freedom q_i successively once by a defined Δq_i . This method finds feasible paths, but due to the nature of the stepped Manhattan motions, the paths obtained tend not to be very fine in their execution.

On the other hand, the implementation of GA's in trajectory planning of robots has been previously investigated by several authors (Davidor, 1991; Doyle & Jones, 1996; Rana & Zalzala, 1996; Wang & Zalzala, 1996; Chen & Zalzala, 1997; Kubota et al., 1997; Nearchou & Aspragathos, 1997; Gill & Zomaya, 1998).

In (Parker, Khoogar et al., 1989; Davidor, 1991; Chen & Zalzala, 1997; Gill & Zomaya, 1998) GA's are used to optimise the search for a solution of the inverse kinematics using the Jacobian matrix to relate the differential variations in the joint space to the workspace.

Chen and Zalzala focused their study on the position and configuration of a mobile robot. Like other conventional approaches based on the solution of the inverse kinematics, a singularity in the solution takes place when the Jacobian equals zero and therefore cannot be inverted.

In Davidor's approach, a desired path for the end-effector of the manipulator is pre-defined in the workspace. A number of trajectories in the C-space are randomly generated and are evaluated with respect to the trajectory they describe in the workspace against the required path. New paths are evolved by selecting analogous crossover points that will fit the evaluated trajectories closer to the desired one. The process of generating a new population starts by copying the joint displacements of the selected trajectory segments into a new string, producing chromosomes of different length in the same population. This process is repeated until an acceptable fit to the desired path is obtained. In general, the GA is applied to solve the inverse kinematics to pre-defined end-effector paths of the manipulator.

In (Gill & Zomaya, 1998), either the potential field or the cell decomposition approach are used to obtain the next best position from a starting configuration to a desired goal. This algorithm considers an alternative temporary goal if it gets stuck in a local minimum. Once the next position has been calculated, a GA is used to search for a solution to the inverse kinematics that best reduces the error between the current and desired end-effector position calculated in the previous step. To avoid the problem of singularities, the search scope of the GA for the solution of the inverse kinematics is restricted to a closed range of possible values for the increments of the joint positions of the manipulator.

The methods presented by Davidor and Gill & Zomaya can only be applied in non-redundant manipulators. In (Nearchou & Aspragathos, 1997), a similar path fitting method using GAs for redundant manipulators is presented. This approach was the first to solve the problem of path planning in the workspace of the manipulator rather than the C-space. A set of via points is established in the workspace and the GA is used to solve the inverse kinematics of the manipulator for the required positions. From the reported results, it can be appreciated that, when applying a path fitting algorithm to a redundant manipulator, the set of configurations for the joints of the manipulator to follow the desired path are not optimal with respect to the displacement to which the links are subjected. This is a result of the inherited dexterity of a redundant manipulator, where a single point in the workspace of the manipulator can be reached by a number of different configurations of the robot.

To avoid the problem of singularities, some authors like (Doyle & Jones, 1996; Rana & Zalzal, 1996; Wang & Zalzal, 1996; Kubota, Arakawa *et al.*, 1997; Gill & Zomaya, 1998) base their approaches on the direct kinematics. (Doyle & Jones, 1996) use a GA to search an optimum path for the robot in the C-space. (Rana & Zalzal, 1996) also use GA's to find an optimum path in the C-space of non-redundant manipulators and also take into consideration the required time to perform a task to find a near-optimal solution; the path is represented as a string of via points connected through cubic splines.

In (Merchán-Cruz & Morris, 2004) the application of GA's is extended to obtain collision free trajectories of a system of two manipulators, either non-redundant or redundant, using potential field approach. In this approach each manipulators is considered as a moving obstacle by the other and collision is avoided. The GA carries parallel optimization to find the best configuration for collision free as well as minimizing the error to their respective goals. Later, in (Merchán-Cruz *et al.*, 2007) it is explained how the inherited monotony of the trajectories described by the robot manipulators, either when moving along a collision free path, or when avoiding an obstacle, can be exploited to enhance the performance of a GA based trajectory planner.

2.2.2 Fuzzy Logic based approaches

The use of an artificial potential field (APF) to describe the workspace of a manipulator is naturally appealing for the implementation of a fuzzy collision avoidance strategy since the value of the potential field increases or decreases accordingly to the nearness of obstacles or to the goal. In contrast to many methods, robot motion planning based on an artificial potential field (APF) is able to consider simultaneously the problems of collision avoidance and trajectory planning as this approach can drive the robot to its desired goal while keeping it from colliding with static or dynamic obstacles present in the workspace.

However, the use of this method is often associated with local minima problems which cause the planning algorithms to get stuck in a sub-optimal solution. While some previous

research has been done in constructing potential fields free from local minima, other research has focused on the development of strategies to identify and move away from local minima within the planning algorithm.

Following on from the original approach proposed in (Khatib, 1986), (Volpe & Khosla, 1990) developed elliptical potential functions called superquadric artificial potential field functions which, from the results, do not generate local minima when used to represent the workspace in the cartesian space, but the problem of local minima is still present when used with the C-space representation. Later on, (Koditschek, 1991; Koditschek, 1991; Rimon & Koditschek, 1992) introduced analytical potential fields in the C-space for ball or star shaped obstacles which limited its application to other obstacle arrangements. The use of harmonic functions to construct local-minima free potential fields is explored in (Kim & Kolsha, 1992; Connolly, 1994), where the potential field is constructed for the C-space of the manipulator. In a more recent approach, another alternative in the construction of the potential field presented in (Vadakkepat et al., 2000) is the use of an evolutionary algorithm for the optimisation in the construction of the potential field. This approach starts by building a potential field using simple expressions, from where a local minima-free potential field is evolved. The size and resolution of the potential field reflects on the necessary number of generations needed to obtain an optimal field, and its application is restricted to static workspaces due to the computations involved if a dynamic obstacle is considered.

With regards to the algorithms for motion planning of robot manipulators using fuzzy logic, very little work can be referred to in comparison to the research done for mobile robots. The first reported approach of a fuzzy navigator for robot manipulators is that reported in (Althoefer & Fraser, 1996) and further discussed in (Althoefer, 1997; Althoefer et al., 1998; Althoefer et al., 2001). In this approach, the goal is specified as a particular arm configuration to which the manipulator is required to reach, that is, the goal is specified in the joint space of the robot. A fuzzy navigator is presented consisting of individual fuzzy units that govern the motions of each link individually and determine the distance to an obstacle by constantly scanning a predetermined rectangular surrounding area around each link. These areas correspond to the model of the space that could be obtained by ultrasonic-sensors attached to both sides of each link. Each fuzzy unit produces an output relative to the joint error and the closeness to an obstacle. The application of this approach is presented for 2dof and 3dof planar manipulators in static environments.

The work by (Althoefer, 1997) presents a briefly discussion of the application of this approach where a 3dof planar manipulator deals with a point obstacle whose trajectory is known.

In (Zavlangas & Tzafestas, 2000), a modified version of this approach is presented to consider the specific case of the implementation to an Adept 1 industrial manipulator in a 3D space, the Adept 1 architecture corresponds to that of a SCARA manipulator. The modification consists of the way the nearness to an obstacle is determined. In this case, the area surrounding the links is said to be a cylindrical volume surrounding each link. Again, as in the original work presented by Althoefer, the start and goal of the manipulator are both specified in the joint space of the robot as a set of given configurations. This implementation, due the nature of the SCARA configuration, can be simplified to just analysing a 2dof planar manipulator.

In (Mbede et al., 2000), a similar approach is presented where an harmonic potential field is used to drive a 2dof planar manipulator to its desired goal, which has been described in the

joint space. The cases of static obstacles and that of a trajectory-known point obstacle are presented and successful results are reported. Later, in (Mbede & Wei, 2001), a neuro-fuzzy implementation of this approach is used to solve the same proposed cases.

3. Fuzzy-GA trajectory planner

The fuzzy-GA trajectory planner (FuGATP) approach presented in this section employs a simple-GA based trajectory planner (simple-GABTP) to initially drive the manipulators towards their goals. Individual fuzzy units provide a correction in the displacement of each articulation in the event that a link is approaching an obstacle. The goals are specified in workspace coordinates to take full advantage of the dexterity of the redundant manipulators.

The manipulators are modelled as obstacles in the workspace using the artificial potential field (APF) method. Since this is a local approach, the APF is only calculated for the near vicinity of the links of the manipulator to save time in the modelling process. Because each manipulator is considered by the other as a mobile obstacle where the position and orientation on their links are constantly changing, there would be a major computational problem if the APF for the whole of the shared workspace were to be calculated each time one of the manipulators has moved as would be necessary in a global approach.

The simple-GABTP algorithm, illustrated in figure 6, carries out an initial estimation of the motion of the manipulators without considering the presence of any obstacles. This calculation is performed by finding the best set of joint displacements within a pre-established range of $\Delta\theta$'s per unit of time until the goal is reached, without solving the inverse kinematics of the manipulator or dealing with singularities in the case of redundant manipulators.

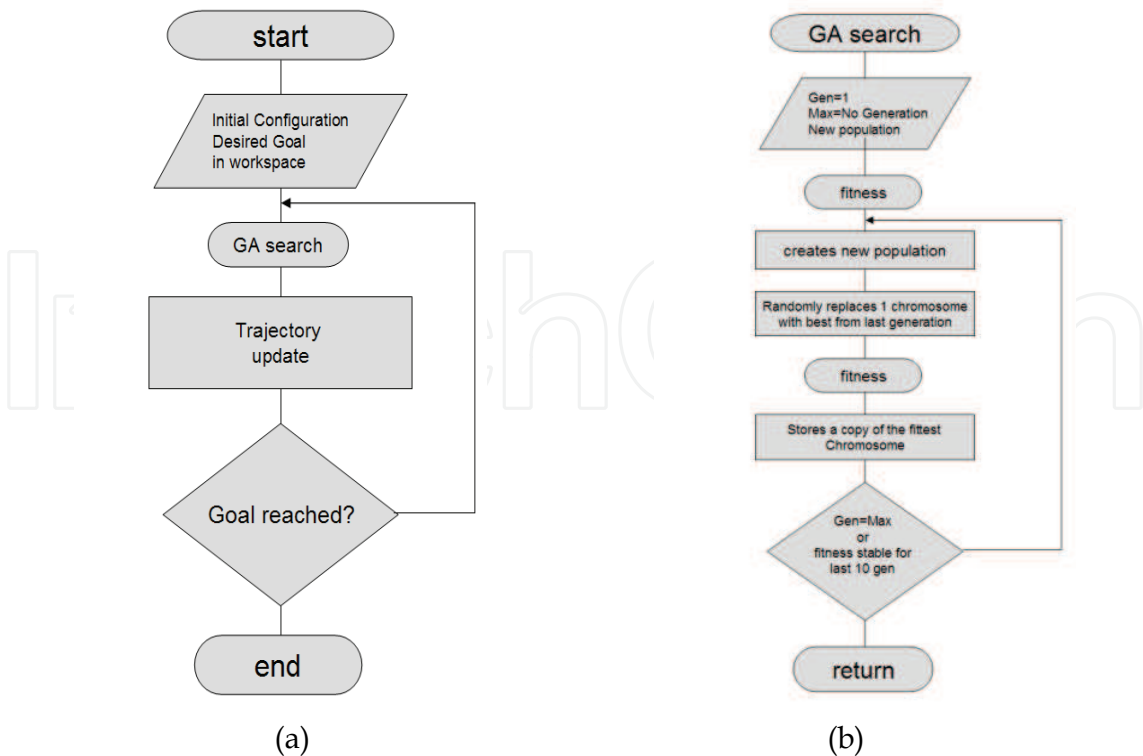


Figure 6. Simple GABT algorithm. (a) General outline of the planner; (b) GA search process

In order to favour fully-extended motion of the manipulator to minimise unnecessary displacements of the links when moving in a space free from obstacles, the fitness function for the GA search considers the degree of extension of the manipulator, equation 4.

$$f = \frac{1}{e^{\frac{error}{R}}} \quad (4)$$

Where:

$$error = \|(x_g, y_g) - (x, y)\|$$

$$R = \|(x_o, y_o) - (x, y)\|$$

(x_o, y_o) = Manipulator's base coordinates

(x_g, y_g) = Goal coordinates

$$(x, y) = DKM((\theta_1 + \Delta\theta_1), (\theta_2 + \Delta\theta_2), \dots, (\theta_n + \Delta\theta_n))$$

DKM = Direct Kinematic Model as a function of θ 's

Each chromosome of the population considered by the GA is composed of substrings of equal length corresponding to a particular joint variation. Figure 7 shows how these strings are decoded from the binary representation into joint increments/decrements which are then applied to the DKM of the manipulators to calculate the new error due to these variations. In this representation, m determines the minimum increment and n is the number of joints in the manipulator. The algorithm returns the set of $\Delta\theta$'s that best reduces the error, updates the configuration of the manipulator and continues until the goals are reached.

$$\begin{array}{ccccccc} [& x & x & x & x & x & \dots & x & | & x & x & x & x & x & \dots & x & | & \dots & | & x & x & x & x & x & \dots & x &] \\ & \pm & 2^1 & 2^0 & 2^{-1} & 2^{-2} & \dots & 2^{-m} & \pm & 2^1 & 2^0 & 2^{-1} & 2^{-2} & \dots & 2^{-m} & \dots & \pm & 2^1 & 2^0 & 2^{-1} & 2^{-2} & \dots & 2^{-m} \\ & \pm & \Delta\theta_1 & & & & & & \pm & \Delta\theta_2 & & & & & & & \pm & \Delta\theta_n \end{array}$$

Figure 7. Chromosome structure

Once the simple-GABTP algorithm has determined an initial set of configurations that best reduce the Cartesian error between the current end-effector positions of the manipulator, the fuzzy units provide a correction based on the magnitude of the AFP that modifies the original displacement of the articulations given by the GABTP. This correction ranges from slowing the articulation to completely change the direction of the original motion given by the GABTP. The fuzzy units determine the best direction to correct the original motion of an articulation by determining the direction of the motion that implies an increment on the magnitude of the APF at the particular instant consider by the planner. This results in a signed correction to affect the initial estimation given by the GABTP.

The direct output from the FuGABTP algorithm is given by:

$$[\Delta\theta_1 \quad \Delta\theta_2 \dots \Delta\theta_n]_{FuGABTP} = [\Delta\theta_1 \quad \Delta\theta_2 \dots \Delta\theta_n]_{GABTP} + [\Delta\theta_1 \quad \Delta\theta_2 \dots \Delta\theta_n]_{FuzzyCorrection} \quad (5)$$

The diagram shown in figure 8 illustrates the general outline of the trajectory planner.

The fuzzy correction is achieved by individual Mamdani-type fuzzy units that provide a correction value per articulation in order to avoid collision. Each of the units has three

inputs: obstacle direction, APF magnitude and error; and a single output named correction, as illustrated in Figure 9.

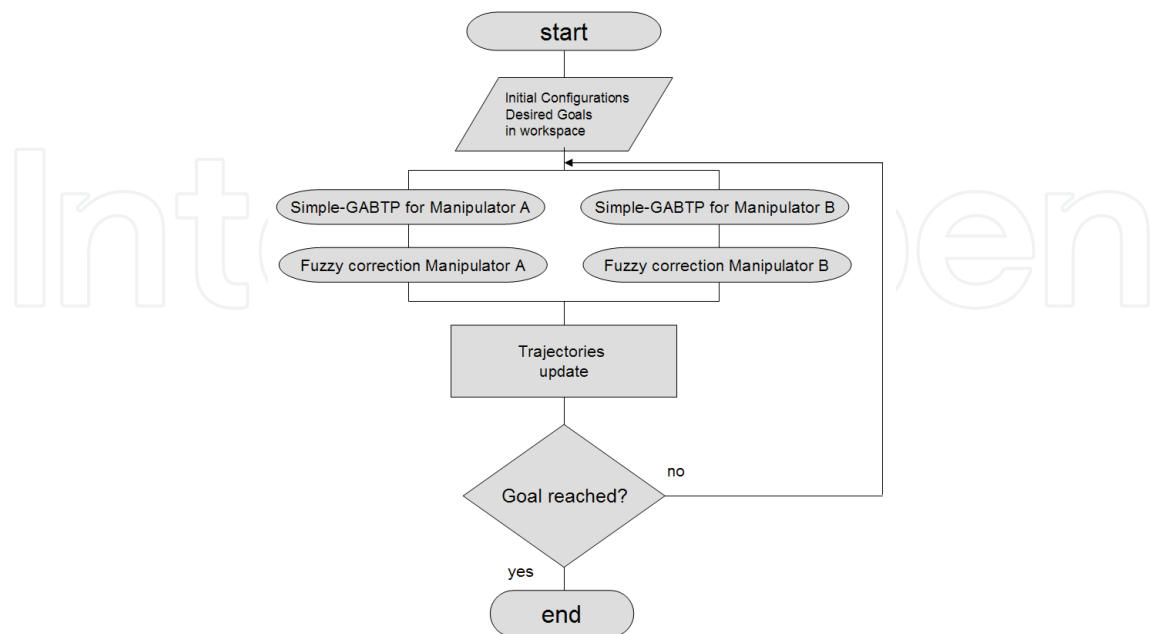


Figure 8. General outline of Fuzzy-GA trajectory planner

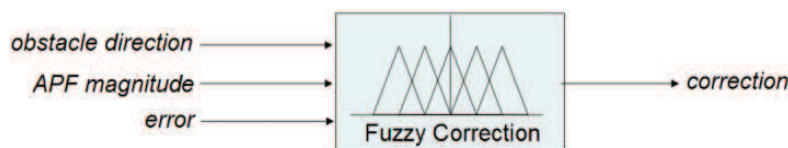


Figure 9. Fuzzy correction unit

The obstacle direction can either be right if the magnitude of the APF increases by moving the considered link around its axis of movement in a positive direction, or left if the magnitude of the APF increases in the opposite direction.

The magnitude of the APF indicates how close a link is from a nearby obstacle, and the error is the Cartesian error between the actual position of the end-effector and the goal. Finally, the correction output indicates the positive or negative correction needed to ensure that no collision takes place. The range of each variable, or universe of disclosure D , is partitioned into fuzzy sets $\mu_1 \dots \mu_p$. Each of the sets $\mu_{\bar{p}}$, $\bar{p} = 1, \dots, p$ represents a mapping $\mu_{\bar{p}}(d) : D \rightarrow [0,1]$ by which the degree of membership of d within a fuzzy set is associated with a number in the interval $[0,1]$.

The functions employed to represent the fuzzy sets have been chosen to be asymmetrical triangular functions as they are easily computed. The degree of membership of each input is calculated by :

$$\mu_{\bar{p}}(d) = \begin{cases} \min\left(\frac{(d - ml_{\bar{p}})}{mc_{\bar{p}} - ml_{\bar{p}}}, 0\right) & \text{if } d \leq mc_{\bar{p}} \\ \min\left(\frac{(d - mr_{\bar{p}})}{mc_{\bar{p}} - mr_{\bar{p}}}, 0\right) & \text{if } d > mc_{\bar{p}} \end{cases} \quad (6)$$

where:

$ml_{\bar{p}} \quad mr_{\bar{p}} \quad =$ x-coordinates of the left and right zero crossing, and

$mc_{\bar{p}} \quad =$ x-coordinate where the fuzzy set becomes 1

For the values that lie either at the left or right of each input range, the triangular functions are continued as constant values of magnitude 1 as:

$$\mu_1(d) = \begin{cases} 1 & \text{if } d \leq mc_1 \\ \min\left(\frac{(d - mr_1)}{mc_1 - mr_1}, 0\right) & \text{if } d > mc_1 \end{cases} \quad (7)$$

and

$$\mu_p(d) = \begin{cases} \min\left(\frac{(d - ml_p)}{mc_p - ml_p}, 0\right) & \text{if } d \leq mc_p \\ 1 & \text{if } d > mc_p \end{cases} \quad (8)$$

Figures 10 and 11 illustrate the partition of the fuzzy sets that describe each of the fuzzy variables.

The defuzzification of the data into a crisp output is accomplished by combining the results of the inference process and then computing the "fuzzy centroid" of the area. The weighted strengths of each output member function are multiplied by their respective output membership function centre points and summed. Finally, this area is divided by the sum of the weighted member function strengths and the result is taken as the crisp output.

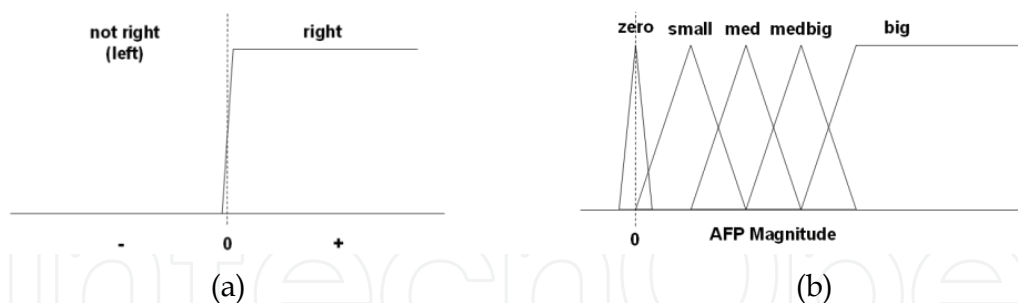


Figure 10. Fuzzy sets (a) Obstacle direction, (b) APF Magnitude

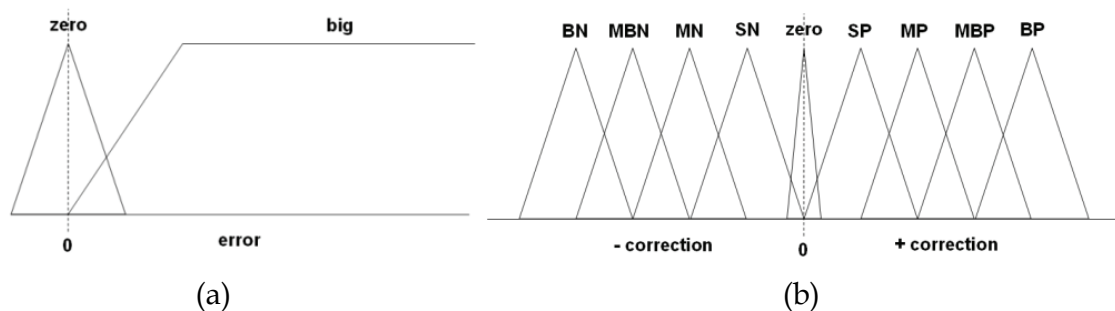


Figure 11. Fuzzy sets (a) Error, (b) Output correction

The correction output of each fuzzy unit is given as the product of the intersection of the following fuzzy rules:

If (obstacle is left) and (APF Magnitude is small) then (Correction is SN)
If (obstacle is left) and (APF Magnitude is med) then (Correction is MN)
If (obstacle is left) and (APF Magnitude is medbig) then (Correction is MBN)
If (obstacle is left) and (APF Magnitude is big) then (Correction is BN)
If (obstacle is left) and (APF Magnitude is zero) then (Correction is zero)
If (obstacle is right) and (APF Magnitude is small) then (Correction is SP)
If (obstacle is right) and (APF Magnitude is med) then (Correction is MP)
If (obstacle is right) and (APF Magnitude is medbig) then (Correction is MBP)
If (obstacle is right) and (APF Magnitude is big) then (Correction is BP)
If (obstacle is right) and (APF Magnitude is zero) then (Correction is zero)
If (obstacle is left) and (error is zero) then (Correction is zero)
If (obstacle is right) and (error is zero) then (Correction is zero)

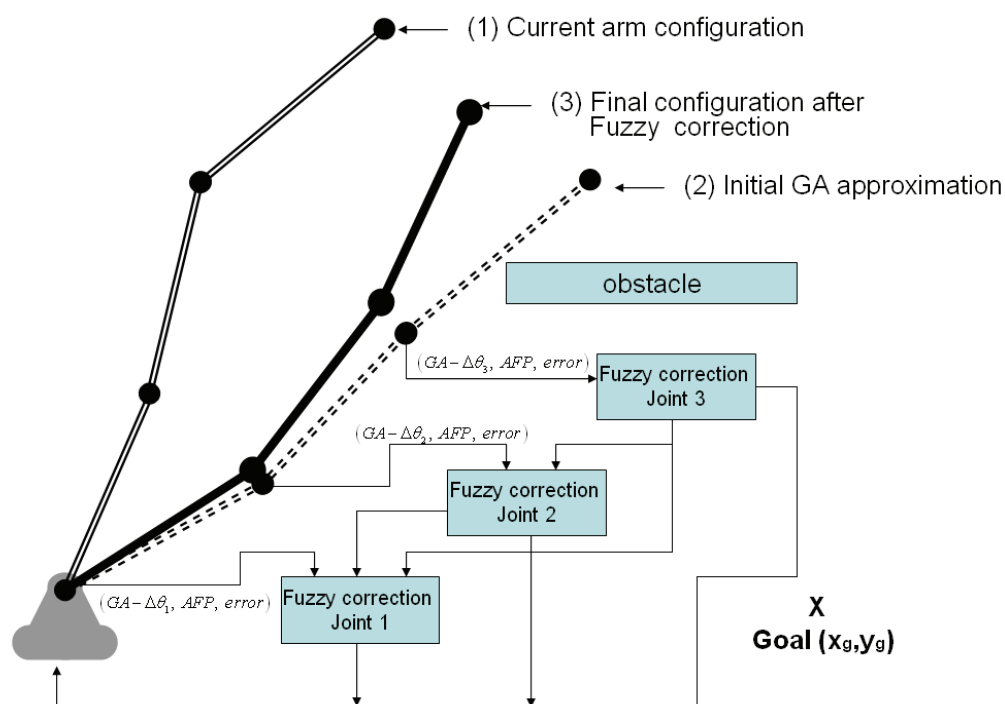


Figure 12. Fuzzy-GA trajectory planner (FuGATP) approach, schematic representation (Merchán-Cruz & Morris, 2006)

The final fuzzy correction for each joint is also affected by the correction of the distal links since the motion of the distal links is not only given by the action of a single articulation. The overall approach is illustrated for a 3dof manipulator in Figure 12 where (1) shows the manipulator at the current configuration, (2) indicates the initial GA estimation that simply reduces the Cartesian error between the current position of the end-effector and its goal, and finally, (3) shows the final position of the manipulator after the fuzzy units have corrected the final $\Delta\theta$'s for the manipulator. Since the $\Delta\theta$'s are obtained by the planner, the velocities and accelerations necessary to calculate the necessary torque of each articulation are easily calculated as a function of time.

3.1 Simulations for two manipulators sharing a common workspace

In this section the described approach is applied to solve the motion planning problem for the 3dof & 7dof two-manipulators systems. Initially the problem of static obstacles is dealt with by considering the manipulator B as a stationary obstacle. Finally, the approach is tested when both manipulators are required to move to reach their independent goals while considering the other manipulator as a mobile obstacle with unknown trajectory.

3.1.1 Static case

For the study of the static obstacle case, the manipulator A is required to reach a goal in the workspace from a given starting configuration while the manipulator B remains in a stationary position where it behaves as a static obstacle. The following cases are presented, as they test the performance of the algorithm in different circumstances and are considered to be representative of the common task that the manipulator performs in its workspace. Tables 1 and 2 summarise these cases for the 3dof and 7dof manipulators.

The example cases for the 3dof system corroborate the application of the proposed approach for a planar manipulator with low redundancy, while the example cases solved for the 7dof manipulator probe that the algorithm is also applicable for highly redundant manipulators. Table 3 summarises the results for the application of the Fuzzy-GA trajectory planner considering a static obstacle.

Case	Manipulator A			Goal (x,y)		Manipulator B			Goal (x,y)	
	θ_{1i}	θ_{2i}	θ_{3i}	x	y	θ_{1i}	θ_{2i}	θ_{3i}	x	y
I	90	0	0	0	-2	0	0	0	1.5	0
II	90	0	0	1.9	-0.5	0	0	0	1.5	0
III	-35	20	15	1.95	0.5	0	0	0	1.5	0

Table 1. Representative cases for the two-3dof manipulator system

Case	Manipulator A							Goal (x,y)	
	θ_{1i}	θ_{2i}	θ_{3i}	θ_{4i}	θ_{5i}	θ_{6i}	θ_{7i}	x	y
I	90	0	0	0	0	0	0	0	-4
II	90	0	0	0	0	0	0	2.75	-1.705
III	90	0	0	0	0	0	0	3.5	-1.1
IV	90	0	0	0	0	0	0	3.95	-0.45
V	85	-25	-25	-35	0	0	0	2.75	-1.705

Case	Manipulator B							Goal (x,y)	
	θ_{1i}	θ_{2i}	θ_{3i}	θ_{4i}	θ_{5i}	θ_{6i}	θ_{7i}	x	y
I	0	0	0	0	0	0	0	1.75	0
II	0	0	0	0	0	0	0	1.75	0
III	0	0	0	0	0	0	0	1.75	0
IV	0	0	0	0	0	0	0	1.75	0
V	0	0	0	0	0	0	0	1.75	0

Table 2. Representative cases for the two-7dof manipulator system

Figure 13 illustrates the described path and trajectory profiles for case III of the 7dof system. The figure shows how the manipulators move from the given start configuration to the end-effector goal in the workspace. Collision with the static manipulator is successfully avoided.

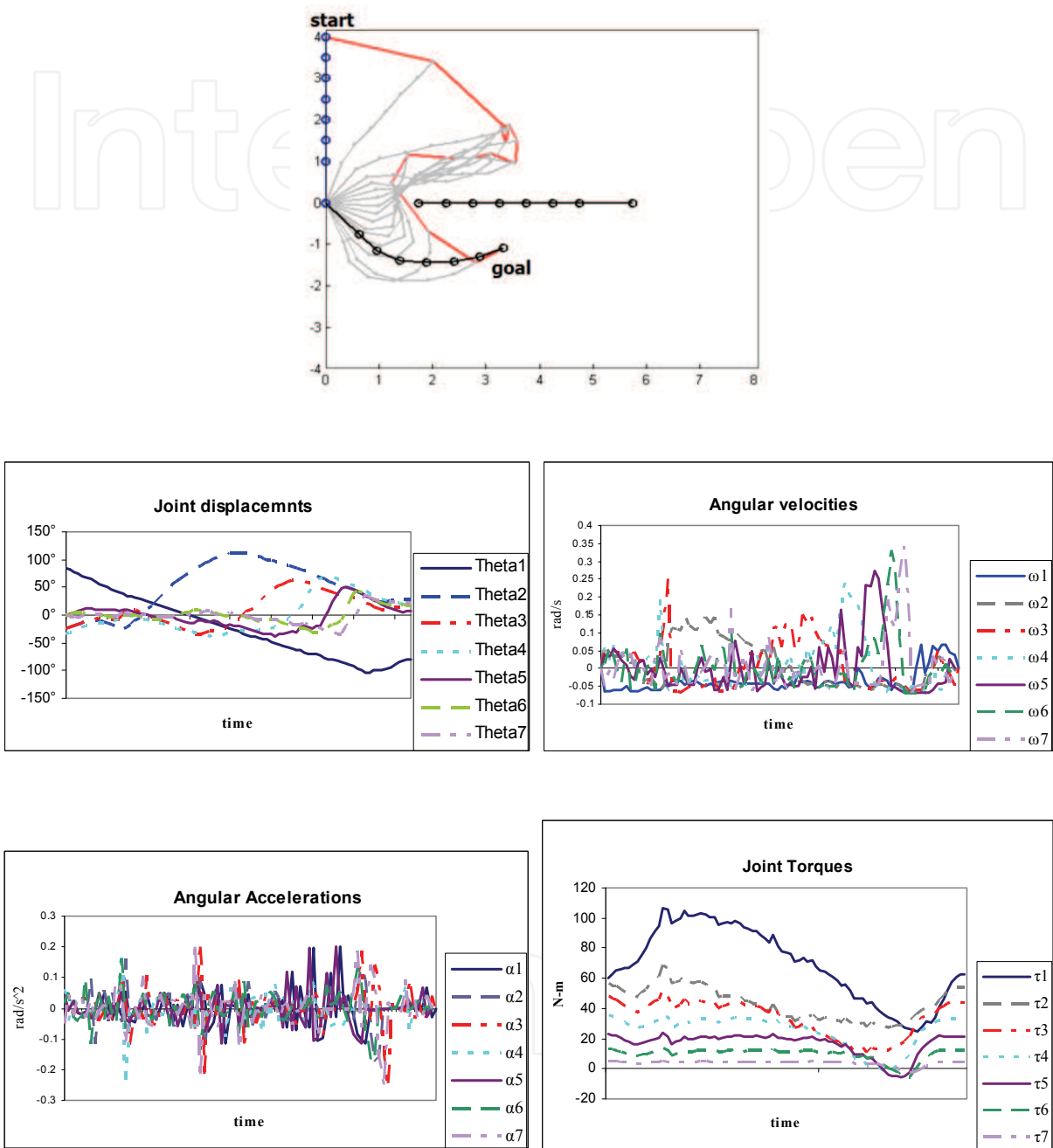


Figure 13. Described path and trajectory profiles for the 7dof manipulator case III, considering a static obstacle obtained with the FuGATP algorithm

The combination of the simple-GABTP algorithm with the fuzzy correction drives the manipulator towards the desired goal in the workspace while keeping the manipulator away from the obstacle by moving the link away from it as a result of the correction action determined by the fuzzy units.

3dof			
Case	Trajectory Segments	Time (s)	Avg per segment (s)
I	63	9	0.14
II	61	17	0.28
III	50	7	0.14

7dof			
Case	Trajectory Segments	Time (s)	Avg per segment (s)
I	182	36	0.20
II	177	34	0.19
III	165	31	0.19
IV	199	40	0.20
V	170	35	0.20

Table 3. Summary of results for the two-manipulator system considering one as a static obstacle

3.1.2 Dynamic case

In this section, the FuGATP algorithm is applied to solve the trajectories for two manipulators sharing a common workspace where both manipulators have the same priority when moving towards individual goals. As in the previous section, the examples presented here are just a selection to illustrate the applicability of the algorithm when one manipulator has to consider the other as a mobile obstacle of unknown trajectory. Table 4 summarises these representative cases where each one tests the algorithm under different circumstances.

Manipulator A								Goal (x,y)	
Case	θ_{1i}	θ_{2i}	θ_{3i}	θ_{4i}	θ_{5i}	θ_{6i}	θ_{7i}	x	y
I	-15	15	0	0	0	0	0	3.8	1
II	-90	0	0	0	0	0	0	0.5	3.5
III	90	0	0	0	0	0	0	3.8	-0.5

Manipulator B								Goal (x,y)	
Case	θ_{1i}	θ_{2i}	θ_{3i}	θ_{4i}	θ_{5i}	θ_{6i}	θ_{7i}	x	y
I	-15	15	0	0	0	0	0	1.95	-1
II	-90	0	0	0	0	0	0	5.25	-3.5
III	90	0	0	0	0	0	0	2	0.5

Table 4. Representative cases for the two-7dof manipulator system with individual goals
When individual goals have been specified and these require that both manipulators move, the algorithm starts solving in parallel the trajectory for both manipulators as described in

the algorithm outline (Figure 8) by considering the current individual configuration of each manipulator. The simple-GABTP built in the algorithm provides an initial approximation which is corrected by the fuzzy units in the event that this proposed configuration lays within the vicinity of an obstacle, in this case the links of the other manipulator. The components of the algorithm remain the same is with the static obstacle case, with the only difference that the manipulator B is now required to move.

The case of two 7dof planar manipulators is shown in Figure 14, where the manipulators are required to reach the goals from the starting configuration according to case I defined in Table 4. In this example, the applicability of the present approach for a highly redundant system is proven.

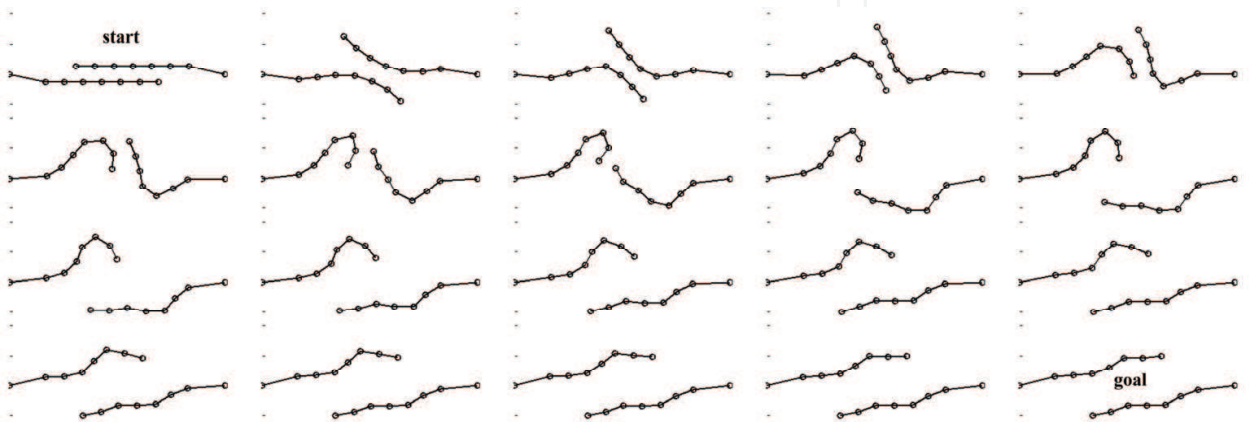


Figure 14. Detailed sequence for the two 7dof manipulators with individual goals Case I

As seen in Figure 14, the FuGATP maintains the manipulators moving towards their goals as a result of the simple-GATP which optimises the configuration of the manipulators to reduce the error to the desired goals. Solving in parallel for both manipulators, as described in Figure 8, produces individual trajectories that do not require from further scheduling as required by traditional decoupled methods. Given that both manipulators have the same priority, both adapt their trajectories when necessary to avoid collision. This avoids over compensating the motion of a single given manipulator as happens with conventional hierarchical approaches. Table 5 summarises the results for the representative cases where each one tests the algorithm under different circumstances.

Case	7dof		
	Trajectory Segments	Time (s)	Avg per segment (s)
I	71	14	0.20
III	128	26	0.20
III	138	27	0.20

Table 5. Summary of results for dynamic cases (two-7dof manipulator system)

4. Conclusion

This chapter has presented a discussion on the application of soft-computing techniques for the trajectory planning of multi-robot manipulator systems. The application of such techniques derived on an approach based on a combination of fuzzy logic and genetic algorithms to solve in parallel the trajectory planning of two manipulators sharing a common workspace where both manipulators have to adjust their trajectories in the event of a possible collision.

The planner combines a simple-GABTP with fuzzy correction units. The simple-GABTP solves the trajectory planning problem as an initial estimation without considering the influence of any obstacles. Once this stage is reached, fuzzy units assigned to each articulation verify that the manipulator is not facing a possible collision based on the magnitude of the APF exerted by the manipulators when considered as obstacles. If the initial output by the GABTP takes a link or links of the manipulator near the influence of the modelled APF, the fuzzy units evaluate a correction value for the $\Delta\theta$ that corresponds to the link and articulation involved in the possible collision and modifies not only the $\Delta\theta$ for that articulation, but also modifies the $\Delta\theta$'s for the more anterior articulations in case where a distal articulation is involved.

The approach presented here has shown a robust and stable performance throughout the different simulated cases. It produced suitable trajectories that are easily obtained since the direct output from the FuGABTP algorithm is a set of $\Delta\theta$'s after each iteration. When a time dimension is added, these provide not only positional information but also the angular velocities and accelerations necessary to describe the trajectory profiles of each manipulator and allow the calculation of the necessary torques to be applied at each joint to produce the obtained trajectory.

Since the planning is done in parallel for both manipulators, no further planning or scheduling of their motions are necessary as is required with decoupled approaches since the trajectories are free from any collision. The implementation of the simple-GABTP in the algorithm solves the problem of over compensation associated with the fuzzy units by maintaining the direction of the manipulators towards the desired goal at all times, keeping the links of the manipulator from over-reacting when approaching an obstacle.

Finally, the proposed algorithm was applied to solve the trajectory planning problem of two manipulators sharing a common workspace with individual goals. In this case each manipulator considers the other as a mobile obstacle of unknown trajectory. The considered cases were successfully solved, obtaining suitable trajectories for both manipulators, obtaining an average of 0.20 seconds per iteration for the 7dof systems in the dynamic cases.

In conclusion, the approach discussed in this chapter could be applicable for real-time application since the compilation of the algorithms that conform the suggested approach into executable files could reduce even more the iteration time.

5. Acknowledgments

The authors would like to thank the National Council for Science and Technology (CONACYT) Mexico, and the Instituto Politécnico Nacional, for the means and support to carry this investigation through the projects SEP-CONACYT 2005/49701 and the corresponding SIP projects.

6. References

- Althoefer, K. (1996). *Neuro-fuzzy motion planning for robotic manipulators*. Department of Electronic and Electrical Engineering. London, UK, King's College London, University of London: 178.
- Althoefer, K. (1997). *Neuro-fuzzy motion planning for robotic manipulators*. Department of Electronic and Electrical Engineering. London, UK, King's College London, University of London: 178.
- Althoefer, K. & D. A. Fraser (1996). Fuzzy obstacle avoidance for robotic manipulators. *Neural Network World* 6(2): 133-142.
- Althoefer, K., B. Krekelberg, et al. (2001). Reinforcement learning in a rule-based navigator for robotic manipulators. *Neurocomputing* 37(1): 51-70.
- Althoefer, K., L. D. Seneviratne, et al. (1998). Fuzzy navigation for robotic manipulators. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems* 6(2): 179-188.
- Ata, A. A. (2007). Optimal Trajectory Planning of Manipulators: A Review. *Journal of Engineering Science and Technology* 2(1): 32-54.
- Barraquand, J. & J. C. Latombe (1991). Robot motion planning: a distributed representation approach. *International Journal of Robotics Research* 10(6): 628-649.
- Bessiere, P., J. Ahuactzin, et al. (1993). The 'Adriane's Clew' algorithm: Global planning with local methods. *Proceedings of the International Conference on Intelligent Robots and System*, IEEE: 1373-1380.
- Brooks, R. A. (1986). A robust layered control system for a mobile robot *IEEE Journal of Robotics and Automation* 2(1): 14-23.
- Connolly, C. I. (1994). *Harmonic Functions as a basis for motor control and planning*. Department of Computer Science. Amherst, MA, USA., University of Massachusetts.
- Connolly, C. I., J. B. Burns, et al. (1990). Path planning using Laplace's equation. *Proc. of the 1990 IEEE International Conference on Robotics and Automation*: 2102-2106.
- Connolly, C. I. & R. A. Grupen (1993). On the application of harmonic functions to robotics. *Journal of Robotc Systems* 10(7): 931-946.
- Coste-Maniere, E., L. Adhami, et al. (2003). Optimal Planning of Robotically Assisted Heart Surgery: Transfer Precision in the Operating Room. *8th International Symposium on Experimental Robotics*: 721-732.
- Chang, C., M. J. Chung, et al. (1994). Collision avoidance of two general manipulators by minimum delay time. *IEEE Transactions on Systems, Man and Cybernetics* 24(3): 517-522.
- Chen, M. & A. M. S. Zalzal (1997). "A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration." *Journal of Robotc Systems* 14(7): 529-544.
- Chen, P. C. & Y. K. Hwang (1998). SANDROS: A dynamic graph search algorithm for motion planning. *IEEE Transactions on Robotics and Automation* 14(3): 392-403.
- Davidor, Y. (1991). *Genetic algorithms and robotics: a heuristic strategy for optimization*, World Scientific.
- Doyle, A. B. (1995). *Algorithms and computational techniques for robot path planning*. Bangor, The University of Wales.

- Doyle, A. B. & D. I. Jones (1996). Robot path planning with genetic algorithms. *Proc. of 2nd Portuguese Conference on Automatic Control (CONTROLO)*: 312-318.
- Eldershaw, C. (2001). *Heuristic algorithms for motion planning*. Oxford, UK, The University of Oxford: 149.
- Fortune, S., G. Wilfong, et al. (1987). Coordinated motion of two robot arms. *International Conference on Robotics and Automation*.
- Fraichard, T. & C. Laugier (1992). Kinodynamic Planning in a Structured and Time-Varying 2D Workspace. *Proc. of the 1992 International Conference on Robotics and Automation*: 1500-1505.
- Freund, E. & H. Hoyer (1985). Collision avoidance in multi-robot systems. *Robotics Research: Proc. of The Second International Symposium*. H. Hanafusa and H. Inoue. Cambridge, MA USA, MIT Press: 135-146.
- Gill, M. A. C. & A. Y. Zomaya (1998). Obstacle avoidance in multi-robot systems, *World Scientific*.
- Goldberg, D. E. (1983). Computer-aided gas pipeline operation using genetic algorithms and rule learning (*Doctoral dissertation*), University of Michigan.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press.
- Hwang, Y. K. & N. Ahuja (1992). Gross motion planning - A survey. *ACM Computing Surveys* 24(3): 219-291.
- Isto, P. (2003). *Adaptive probabilistic roadmap construction with multi-heuristic local planning*. Department of Computer Science and Engineering. Espoo, Finland, Helsinki University of Technology: 62.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research* 5(1): 90-98.
- Kim, J. & P. K. Kolsha (1992). Real time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation* 8(3): 338-349.
- Koditschek, D. E. (1991). The control of natural motion in mechanical systems. *Journal of Dynamic Systems, Measurement, and Control* 113: 547-551.
- Koditschek, D. E. (1991). Some applications of natural motion. *Journal of Dynamic Systems, Measurement, and Control* 113: 552-557.
- Kubota, N., T. Arakawa, et al. (1997). Trajectory generation for redundant manipulator using virus evolutionary genetic algorithm. *Proc. of International Conference on Robotics and Automation (ICRA)*. Albuquerque, NM, USA., IEEE: 205-210.
- Latombe, J. C. (1991). *Robot Motion Planning*. Boston, MA, USA, Kluwer Academic.
- Lee, B. H. & C. S. G. Lee (1987). Collision free motion planning of two robots. *IEEE Transactions on Systems, Man and Cybernetics* 17(1): 21-32.
- Lee, D. (1996). *The map-building and exploration strategies of a simple sonar-equipped mobile robot: An experimental, quantitative evaluation*. UK, Cambridge University Press.
- Liu, Y. (2003). Interactive reach planning for animated characters using hardware acceleration. *Computer and Information Science*. Pittsburg, University of Pennsylvania: 95.
- Lozano-Peréz, T. (1983). Spatial planning: A configuration space approach. *IEEE Transactions on Computing* C-32(2): 108-119.
- Lozano-Peréz, T. & M. A. Wesley (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM* 22(10): 560-570.

- Mamdani, E. H. (1974). Application of fuzzy algorithms for control of a simple dynamic plant. *Proc. IEE* 121(12): 1585-1588.
- Masehian, E. & D. Sedighizadeh (2007). Classic and Heuristic Approaches in Robot Motion Planning – A Chronological Review. *Proceedings of World Academy of Science, Engineering and Technology* 23: 101-106.
- Mbede, J. B., X. Huang, et al. (2000). Fuzzy motion planning among dynamic obstacles using artificial potential field for robot manipulators. *Robotics and Autonomous Systems* 32: 61-72.
- Mbede, J. B. & W. Wei (2001). Fuzzy and recurrent neural network motion control among dynamic obstacles for robot manipulators. *Journal of Intelligent and Robotic Systems* 30: 155-177.
- Merchán-Cruz, E. A., L. H. Hernández-Gómez, et al. (2007). Exploiting Monotony on a Genetic Algorithm Based Trajectory Planner (GABTP) for Robot Manipulators. 16th International Conference on Applied Simulation and Modelling, Palma de Mallorca, Spain, IASTED.
- Merchán-Cruz, E. A. & A. S. Morris (2004). GA based trajectory planner for robot manipulators sharing common workspace. *Proceedings of the IASTED International conference on Applied Simulation and Modelling*, Rodos, Greece.
- Merchán-Cruz, E. A. & A. S. Morris (2006). Fuzzy-GA-based trajectory planner for robot manipulators sharing a common workspace. *IEEE Transactions on Robotics* 22(4): 613-624.
- Nearchou, A. C. & N. A. Aspragathos (1997). A genetic path planning algorithm for redundant articulated robots. *Robotica* 15: 213-224.
- Parker, J. K., A. R. Khoogar, et al. (1989). Inverse kinematics of redundant robots using genetic algorithms. *International Conference on Robotics and Automation*, Scottsdale, AZ.
- Rana, A. (1996). *Soft computing algorithms for motion planning of robotic manipulators*. Sheffield, University of Sheffield.
- Rana, A. & A. M. S. Zalzal (1996). An evolutionary planner for near time-optimal collision-free motion of multi-arm robotic manipulators. *Proc. of International Conference on Control*. Exeter, UK. 1: 29-35.
- Ridao, M. A., J. Riquelme, et al. (2001). An evolutionary and local search algorithm for motion planning of two manipulators. *Journal of Robotic Systems* 18(8): 463-476.
- Rimon, E. & D. E. Koditschek (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation* 8(5): 501-518.
- Schwartz, J. T. & M. Sharir (1983). On the 'piano movers' problem II: General techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics* 4: 298-351.
- Sugihara, K. & J. Smith (1997). Genetic algorithms for adaptive motion planning of autonomus mobile robots. *Technical Report*. Manoa, University of Hawaii.
- Tombropoulos, R. Z., J. R. Adler, et al. (1999). CARABEAMER: A treatment planner for a robotic radiosurgical system with general kinematics. *Medical Image Analysis* 3(3): 237-264.
- Vadakkepat, P., C. T. Kay, et al. (2000). Evolutionary artificial potential fields and their application in real time robot path planning. *Proc. of the 2000 Congress of Evolutionary Computation (CEC2000)*, San Diego, California.

- Volpe, R. & P. K. Khosla (1990). Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Transactions on Systems, Man and Cybernetics* 20(6): 1423-1436.
- Wang, Q. & A. M. S. Zalzal (1996). Genetic control of near time-optimal motion for an industrial robot arm. *Proc. of International Conference on Robotics and Automation (ICRA)*. Minneapolis, MI, USA, IEEE: 2592-2597.
- Wise, D. K. & A. Bowyer (2000). A survey of global configuration-space mapping techniques for a single robot in a static environment. *International Journal of Robotics Research*: 762-779.
- Zadeh, L. (1997). What is soft computing? www.ece.nus.edu.sg/stfpage/elepv/softc_def.html 2003.
- Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control* 8(3): 338-353.
- Zavlangas, P. & S. G. Tzafestas (2000). Industrial robot navigation and obstacle avoidance using fuzzy logic. *Journal of Intelligent and Robotic Systems* 27: 85-97.

IntechOpen



Robot Manipulators

Edited by Marco Ceccarelli

ISBN 978-953-7619-06-0

Hard cover, 546 pages

Publisher InTech

Published online 01, September, 2008

Published in print edition September, 2008

In this book we have grouped contributions in 28 chapters from several authors all around the world on the several aspects and challenges of research and applications of robots with the aim to show the recent advances and problems that still need to be considered for future improvements of robot success in worldwide frames. Each chapter addresses a specific area of modeling, design, and application of robots but with an eye to give an integrated view of what make a robot a unique modern system for many different uses and future potential applications. Main attention has been focused on design issues as thought challenging for improving capabilities and further possibilities of robots for new and old applications, as seen from today technologies and research programs. Thus, great attention has been addressed to control aspects that are strongly evolving also as function of the improvements in robot modeling, sensors, servo-power systems, and informatics. But even other aspects are considered as of fundamental challenge both in design and use of robots with improved performance and capabilities, like for example kinematic design, dynamics, vision integration.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Emmanuel A. Merchan-Cruz, Alan S. Morris and Javier Ramírez-Gordillo (2008). Soft-Computing Techniques for the Trajectory Planning of Multi-Robot Manipulator Systems, Robot Manipulators, Marco Ceccarelli (Ed.), ISBN: 978-953-7619-06-0, InTech, Available from: http://www.intechopen.com/books/robot_manipulators/soft-computing_techniques_for_the_trajectory_planning_of_multi-robot_manipulator_systems

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen