Científica Vol. 9 Núm. 4 pp.197-208
© 2005 ESIME-IPN. *ISSN 1665-0654.* Impreso en México

Soft-Computing Techniques in the Trajectory Planning of Multirobot Manipulators Systems (Part I)

Emmanuel-Alejandro Merchán-Cruz[1]
Guillermo Urriolagoitia-Calderón[2]
Luis-Héctor Hernández-Gómez[2]
Gabriel Villa-y-Rabasa[2]
Luis-Armando Flores-Herrera[1]
Juan-Alejandro Flores-Campos[2]

[1]Sección de Estudios de Posgrado e Investigación (SEPI),
Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME),
Instituto Politécnico Nacional (IPN).
Unidad Profesional "Azcapotzalco",
Av. de las Granjas 682, Col. Sta. Catarina,
CP. 02250, México DF.
MÉXICO
[2]Sección de Estudios de Posgrado e Investigación,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
Instituto Politécnico Nacional. Unidad Profesional
"Adolfo López Mateos". México, DF.
MÉXICO

email: eamerchan@ipn.mx

Recibido el 29 de octubre de 2004; aceptado el 22 de junio de 2005.

1. Abstract

This work is the first of a series of two articles on the application of soft-computing techniques for the trajectory planning of multi-robotic systems. In this first article the motion planning problem of robot manipulators and the particular case of Multi-movers are defined. A brief introduction of the basic elements of soft-computing is given, and the applicability of such elements for the trajectory planning problem of robot manipulators is discussed. Also, the artificial potential field method is presented for the modelling and characterization of obstacles and as a mean of driving a planner towards the desired goal. A genetic algorithm (GA) based path is presented to illustrate the applicability of such techniques for the path planning in a 2D space and a global path planner for planar robot manipulators is developed based on GA optimisation considering static obstacles.

2. Resumen (Técnicas de computación inteligente para la planeación de trayectorias de sistemas con múltiples robots manipuladores)

El trabajo que aquí se presenta es la primera de dos partes sobre la aplicación de herramientas de computación inteligente (*soft-computing*) para la generación de trayectorias de sistemas multirrobóticos. En esta primera parte se define el problema de planeación de movimientos de manipuladores robóticos, en particular el problema de *Mult-imovers*, donde múltiples robots se mueven en un espacio de trabajo común. Se da una breve introducción de los elementos utilizados en la computación inteligente y se discute su aplicabilidad para la planeación de trayectorias de sistemas robóticos. El método del campo de potencial artificial (CPA) se presenta para el modelado y caracterización de obstáculos, así como para que el algoritmo planeador de trayectorias siga el gradiente de descenso de la superficie del CPA hacia el objetivo deseado. Así mismo, se ilustra la aplicación de un planeador basado en algoritmos genéticos para un espacio bi-dimensional y se desarrolla un planeador global basado en optimización genética para manipuladores planos considerando obstáculos estáticos en su espacio de trabajo.

Key words: Trajectory planning, soft-computing, genetic algorithms, fuzzy logic, multiple-manipulators, robotics.

3. Introduction

Ever since the first robot manipulator was installed in 1961 by Devol and Engelberger for General Motors, the planning and execution of movements has been a key part in the development of robotic systems and a topic of discussion for many researchers of various backgrounds.

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 [1, 2]. Whichever the 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 while executing a given task.

Systems where robots have to share a common workspace with other machines, including other robot manipulators, are the result of the demands of industry, where time, space and productivity are closely linked with each other. If the volume of production is very high and the life cycle of the product is quite long, the trajectory planning involved in the operations to manufacture a new product can be performed off-line over a time span of months. Thereafter, it would only be necessary to stop the production line to check the accuracy of the planned trajectories of the robots and adjust accordingly before restarting the production line. However, in systems where the volume of production is small to medium-sized, and a single manufacturing cell has to care for a great variety of products, the process of trajectory planning for the various products has to be highly effective to ensure that the time the production line is off will not affect the final cost of the product. Hence, there is a demand for algorithms that allow more flexibility in the system without the need for off-line planning, thus increasing the productivity of the whole system.

Following this line of thought, we explore the application of soft-computing techniques to improve the performance of local 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 its 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.

4. Development

4.1. Motion planning

Motion planning for robot manipulators has been extensively studied during the last two decades by a number of researchers. A comprehensive survey on the common techniques used to solve this problem can be found in [4, 6]. Depending on the nature of the problem, some authors classify the motion planning problem in two categories: global and local motion planning [6-8].

Global motion planning requires a complete description of the workspace of the robot where all obstacles are clearly identified before searching a path. It often takes place on those 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.

In local motion planning, no prior knowledge of the system is required in terms of possible obstacles in the workspace; this type of planning has 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 of moving obstacles is available [9-11].

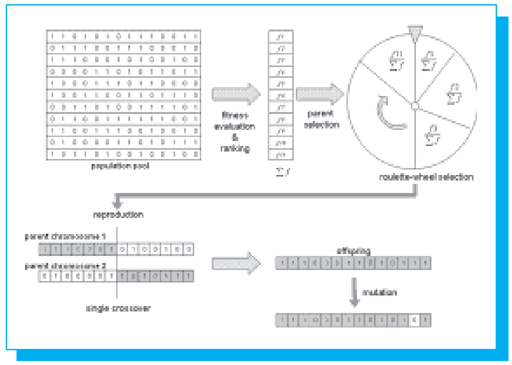


Fig. 1. Genetic algorithm description.

4.2 The Multi-movers 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 intersects 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 [12, 13].

The *hierarchical or prioritized approach* presented by Freund and Hoyer [14], 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

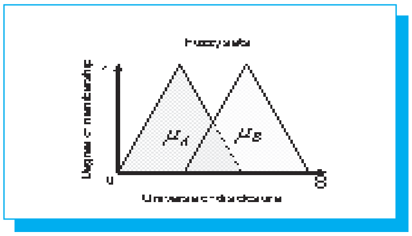


Fig. 2. Fuzzy sets representation.

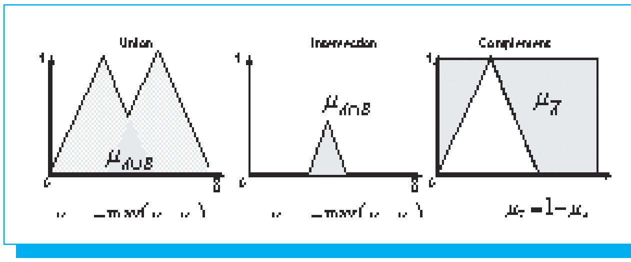


Fig. 3. Fuzzy operators.

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 [15], 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 modeled by spheres and the motion of the robots is restricted to straight-line paths. An extension of this method is presented by Chang *et al*. in [16], where the robots are represented as polyhedral and the minimal delay-time value, necessary for collision free coordination, is determined.

Bien and Lee [17], present a method based on decoupled planning, where the resulting trajectory is collision-free and time-optimal. This method uses a coordination diagram and is restricted to systems where only one area of possible collision is identified. Lee [18], extends this method to consider cases where more than one collision region is identified.

The use of evolutionary techniques in the motion coordination of two manipulators is explored by Ridao et al. in [19]. 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.

4.3 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 [20], "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".

4.3.1 Genetic Algorithms

Developed by Holland [21], and further implemented by Goldberg [22], 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 GA 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. Figure 1 illustrates

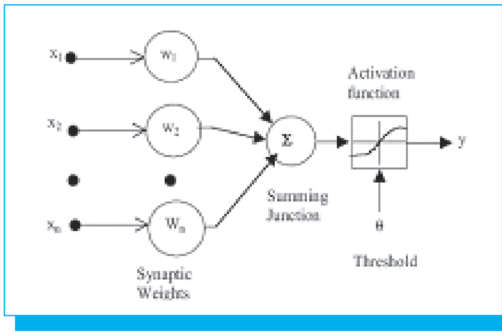


Fig. 4. Artificial neuron.

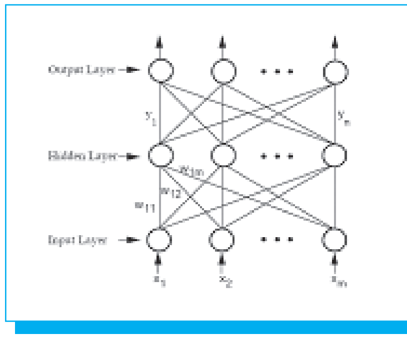


Fig. 5. Multi-layer perceptron.

the general outline of a genetic algorithm with the common operations that take place to produce a single offspring.

The reproduction operator allows individual strings to be copied for possible inclusion in the next generation. The chance that a string will be copied is based on the string's fitness value, calculated from the fitness function. For each generation, the reproduction operator chooses strings that are placed into a mating pool, which is used as the basis for creating the next generation. A commonly used mechanism for parent selection for the reproduction is the roulette-wheel selection.

The crossover refers to the mating of the two strings randomly selected by "spinning" the selection wheel, resulting in a new offspring that forms part of the new population; this operation is performed until a whole new population of the same size as the original is created.

The mutation operator randomly affects only one allele on an offspring. The chance of a mutation occurring is defined by the mutation probability specified in the algorithm. This operator helps to ensure that "new" individuals are included in the new population of solutions to be evaluated.

4.3.2 Fuzzy Logic

Proposed by Zadeh [23] and successfully implemented for the first time by Mamdani [24], 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] (Figure 2), 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. These operations for fuzzy are illustrated in Figure 3 and are defined as:

Union: the union of two fuzzy sets A and B is given by taking the maximum of the degrees of membership of the elements in A and B.

Intersection: the intersection of two fuzzy sets A and B is given by taking the minimum of the degrees of membership of the elements in A and B.

Complement: the complement of a fuzzy set A is obtained by subtracting the degree of membership of the various elements in the domain from 1.

Fuzzy logic uses the before mentioned logical operators in conjunction with «IF <condition> THEN <action>» rules. For example, a rule to control an air conditioner might say: «If the room is hot and moist, circulate the air.» However, unlike Boolean logic, the condition clause does not simply evaluate to «true» or «false»; it is associated with a «degree» of truth providing a variable output as a result. In order to make practical decisions employing fuzzy logic, the following procedures have to be implemented: *fuzzification, inference, composition and defuzzification.*

Fuzzification: establishes the fact base of the fuzzy system. First, it identifies the input and output of the system. Fuzzification then defines appropriate IF THEN rules and uses raw data to derive a membership function.

Inference: as inputs are received by the system, inference evaluates all IF THEN rules and determines their truth values. If a given input does not precisely correspond to an IF THEN rule, then partial matching of the input data is used to interpolate an answer.

Composition: combines all fuzzy conclusions obtained by inference into a single conclusion. Different fuzzy rules might have different conclusions, so it is necessary to consider all rules.

Defuzzification: converts the fuzzy value obtained from composition into a "crisp" value; this process is often complex since the resulting fuzzy set might not translate directly into a crisp value. Defuzzification is necessary, since controllers of physical systems require discrete signals.

4.3.3. Artificial Neural Networks

Artificial neural networks (ANN), more commonly know as neural nets, are an attempt to simulate the way in which the brain interprets and processes information. Based on the early work presented by McCulloch [25], neural nets have been adopted in many cases to model nonlinear systems due to their ability to map nonlinear functions.

A neural net is formed by several interconnected individual processing units called neurons. A neuron (Figure 4) can have any number of inputs (x_j) and the output (y) is given by sum of the weighted inputs and the activation function.

The values w_1, w_2, w_3, …, w_n are weight factors associated with each node to determine the strength of input row vector $X = [x_1, x_2, x_3, \ldots, x_n]^T$. Each input is multiplied by the associated weight of the neuron connection X_W. Depending upon the activation function, if the weight is positive, X_W commonly excites the node output; whereas, for negative weights, X_W tends to inhibit the node output.

The activation function is a mathematical function that a neuron uses to produce an output referring to its input value. This input value has to exceed the specified threshold value that determines if an output to other neurons should be generated.

The structure of a network is determined by how the interneuron connections are arranged and the nature of the connections. How the strengths of the connections are adjusted or trained to achieve a desired overall behaviour of the network is governed by its learning algorithm. Neural networks can be classified according to their structures and learning algorithms. In terms of their structures, neural networks can be divided into two types: feedforward networks and recurrent networks.

Feedforward networks can perform a static mapping between an input space and an output space: the output at a given instant is a function only of the input at that instant. The most popular feedforward neural network is the multi-layer perceptron (MLP): all signals flow in a single direction from the input to the output of the network. Figure 5 shows an MLP with three layers: an input layer, an output layer, and an intermediate or hidden layer. Neurons in the input layer only act as buffers for distributing the input signals x_i to neurons in the hidden layer. Each neuron j in the hidden layer operates according to the model of Figure 4. That is, its output y_j is given by:

$$y_j = \sum_{i=1}^{n} f\left(X_i W_i\right) \tag{1}$$

Finally, the outputs of the neurons in the output layer are computed similarly.

Recurrent networks are networks where the outputs of some neurons are fed back to the same neurons or to neurons in layers before them. Thus signals can flow in both forward and backward directions. Recurrent networks are said to have a dynamic memory: the output of such networks at a given instant reflects the current input as well as previous inputs and outputs. Examples of recurrent networks include the Hopfield network, the Elman network and the Jordan network.

4.4 Discussion

It can be appreciated 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 where 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 [22-28].

4.5 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 by Khatib [29], combines "attractive" and "repulsive" potential field to model the workspace and the obstacles therein. Figure 6 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, *Pa*, is given by reducing the Euclidean error to the desired goal, while the repulsive potential, *Po*, is given by:

$$Po = \sqrt{Po_x^2 + Po_y^2} \tag{2}$$

subject to the following conditions:

$$\begin{aligned} &\text{if } DO > (s+r) && \begin{cases} Po_x = 0 \\ Po_y = 0 \end{cases} \\ &\text{if } r \le DO \le (s+r) && \begin{cases} Po_x = \beta(s+r-OD)\cos\phi \\ Po_y = \beta(s+r-OD)\sin\phi \end{cases} \\ &\text{if } DO < r && \begin{cases} Po_x = m\cos\phi \\ Po_y = m\sin\phi \end{cases} \end{aligned} \tag{3}$$

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

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

$$P = Pa + Po \tag{4}$$

This representation provides enough information to guide, through a search algorithm, a manipulator 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 [30-32]. 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.

Another alternative in the construction of the potential field presented by Vadakkepat in [33], 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, like those of equation 5, and form this field, 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.

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 [6]. A common characteristic of these methods is the addition of a procedure to escape local minima.

4.6 Path planning with GA

4.6.1 Real Coded GA for the Path Planning in a 2D space

This algorithm employs Real coding for the chromosomes which have been chosen to be of variable length.

The algorithm starts by producing a random population of possible solutions from where an optimal solution is to be «evolved». The algorithm assigns a fitness value according to the length and suitability of the path. Parents are selected and the mating process produces a single offspring path, the mating process searches for adequate places where the crossover will take place in order to produce an offspring whose length is shorten.

Figure 7 illustrates the artificial potential field for the considered workspace with two circular obstacles at (50,65) and (70,55) and a goal at (10,10). The path planner has to find a feasible trajectory, free of collision, starting at any point in the workspace. In this example the starting position is located at (95,85). Figure 8 illustrates two possible solutions; the solid trajectory corresponds to that solved by the GA path planner while the dashed one corresponds to the solution obtained from a conventional path planner. As it can be seen, the trajectory obtained by the traditional algorithm is longer that the one obtained by the GA planner as this being an optimal solution, and unlike the conventional planner, keeps the robot away from the obstacles at all times rather than following the contour of the obstacles.

Figures in 9a show the evolution of the trajectories and the nature of the GA can be clearly appreciated. Figure 9a illustrates the initial population considered by the GA, it can be seen the wide range of solutions being considered from where the optimal solution is evolved. Figure 9b shows the population after 5 generations and Figure 9c illustrates the

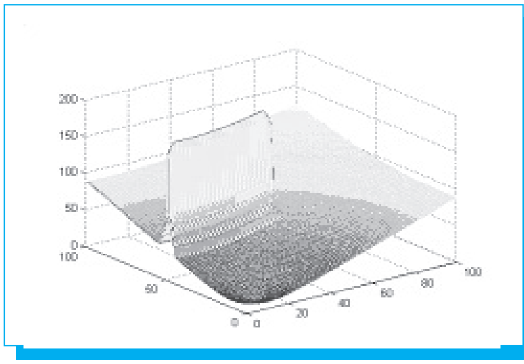


Fig. 6. Combined potential field example.

solutions derived after 10 generations. Finally, Figure 9e illustrates the final solution obtained after only 25 generations where it can be appreciated that the GA has converged to a solution, which can be verified on Figure 9f as the fitness function has become stable throughout the generations.

4.6.2 Implementation on the path planning of Manipulators

Path planning refers to the process of determining the intermediate positions/configurations of the links of the manipulator between the start configuration and a configuration that corresponds with the desired position of the tip of the manipulator when it reaches a specified goal. When time is included in this process, the path planning problem becomes a trajectory planning problem.

For each intermediate position of the manipulator along its specified path or trajectory, suitable values for the joint coordinates have to be determined in order to place the tip of the manipulator at each specified spatial position. This

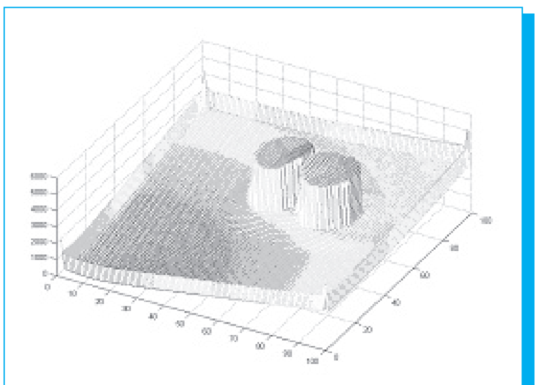


Fig. 7. Potential field two obstacles and a goal at (10,10).

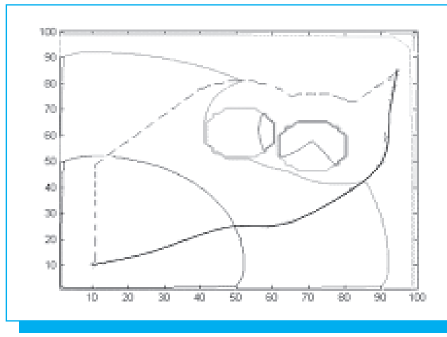
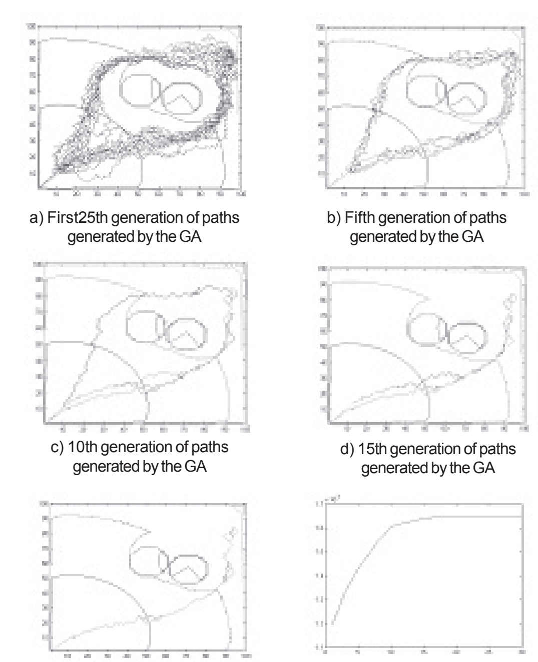


Fig. 8. Path comparison (dashed) conventional path planner (solid) GA path planner.

requires solution of the inverse kinematics at each point along the path.

Unfortunately, solving the inverse kinematics analytically is a very time-consuming computational process, because of the requirement to solve the inverse kinematics for every point along the path.



a) First25th generation of paths generated by the GA

b) Fifth generation of paths generated by the GA

c) 10th generation of paths generated by the GA

d) 15th generation of paths generated by the GA

e) 25th generation of paths generated by the GA

f) Average fitness of the GA

Fig. 9. Generation of paths generated by the GA.

$\Delta\theta$'s [1 1 2 0 0 1 2 2 ... 0 0]

$(\Delta\theta_1,\Delta\theta_2)_1\,(\Delta\theta_1,\Delta\theta_2)_2\,(\Delta\theta_1,\Delta\theta_2)_3\,(\Delta\theta_1,\Delta\theta_2)_4 \ldots (\Delta\theta_1,\Delta\theta_2)_n$

Fig. 10. Chromosome structure.

A GA is very attractive for path planning because it can determine suitable angular increments of the manipulator joints, without requiring analytic solution of the inverse kinematic model. The GA optimises the search for suitable configurations by reducing the end-effector error in the workspace defined in the fitness function of the GA.

The initial population of the GA is coded to represent a positive, a negative and a null displacement of the joint variables from their start configurations using a ternary representation. The step size is chosen to be of 1° and the chromosome length is set to be variable. Figure 10 illustrates the structure of the chromosome based on the described coding where 1 implies a positive displacement of the joint set equal to the determined step size, 2 indicates a negative displacement and, finally, 0 indicates a null displacement. The chromosome is arranged so that, for the case of a 2*dof*, every pair of alleles correspond to particular joint values that drive the manipulator to the desired goal from set 1 to set n. The GA evaluates multiple trajectories as each and every chromosome in the initial population pool represents a different and unique solution (trajectory) from where the best solution is derived.

The fitness function is given by:

$$f = \frac{1}{e^{error_n} + \sum_{i=1}^{n} e^{error_i}} \tag{5}$$

where:

$$error_n = \sqrt{\left(x_f - x_n\right)^2 + \left(y_f - y_n\right)^2} \tag{6}$$

and:

$$\begin{aligned} x_n &= l_1 \cos\left(\theta_{1n}\right) + l_2 \cos\left(\theta_{1n} + \theta_{2n}\right) \\ y_n &= l_1 \sin\left(\theta_{1n}\right) + l_2 \sin\left(\theta_{1n} + \theta_{2n}\right) \end{aligned} \tag{7}$$

(x_f, y_f) = goal coordinates

As it was mention above each chromosome represents a series of joint displacements. The resolution of the GA can be defined to make the step size $\Delta\theta$ equal to the minimum displacement to be considered per time unit and restricted to a maximum value. For illustrative purposes the step size considered in these examples is fixed to ±1° including 0° to represent no displacement of a joint. A ternary representation is used for coding the chromosomes, if the resolution of the GA is to be changed the

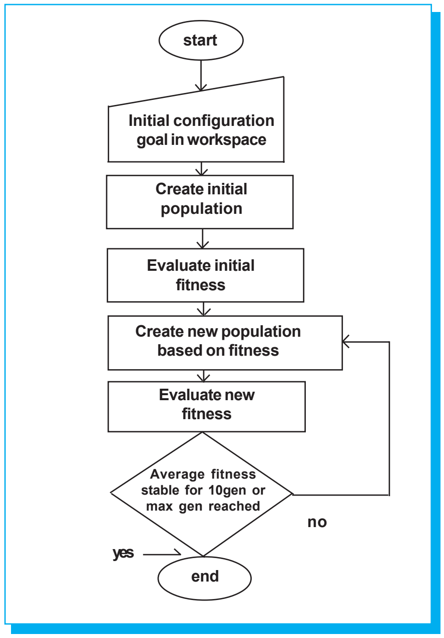


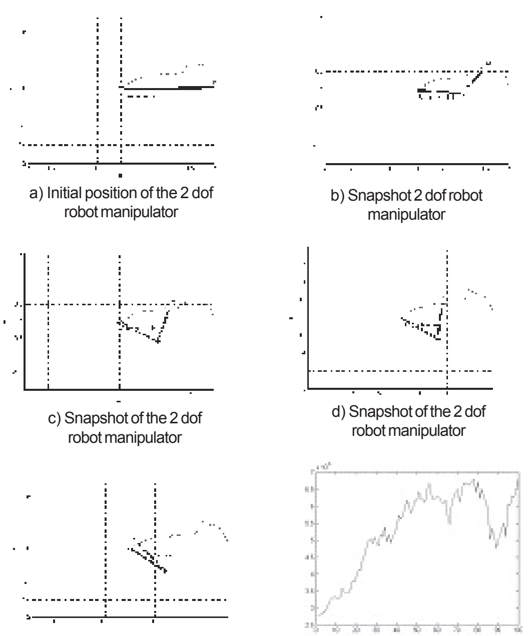
Fig. 11. GA based global path planning algorithm.

coding of the chromosomes will have to reflect it. For example if the minimum $\Delta\theta$ to be considered is equal to 0.5°, and the maximum is equal to 5° it would be necessary to use a base 21 numeric representation to code the chromosomes to include all possible positive and negative values. Figure 11 illustrates the path planner algorithm.

The initial length of the chromosomes is established according to the complexity of the problem. For the simple case of path planning in a workspace free of obstacles illustrated in Figure 17, where the 2 dof manipulator starts in an initial configuration (0°, 0°) and has a goal at (0.1, 0.01), the length of the chromosome is set to 500 elements or 250 sets of $\Delta\theta$. For this example, an initial population of 200 chromosomes is considered.

Figures in 12 show the manipulator moving along the trajectory solved by the planner. The final length of the chromosome containing the illustrated solution is 356 alleles or 178 sets of $\Delta\theta$, which describe the trajectory of the manipulator from its starting configuration to the desired goal of the tip in the workspace. For this example, the time of solution after 100 generations was 87 seconds. Figure 12f shows the average fitness of the population along the generations. It can clearly be appreciated that the average fitness improved until the manipulator reached the maximum number of generations.

For the case of obstacles present in the workspace, the potential field approach defined earlier is used to characterise them. The fitness function of the GA is defined as a multi-



a) Initial position of the 2 dof robot manipulator

b) Snapshot 2 dof robot manipulator

c) Snapshot of the 2 dof robot manipulator

d) Snapshot of the 2 dof robot manipulator

e) Snapshot of the 2 dof robot manipulator at its goal

f) Average fitness

Fig. 12. Snapshot and averages fitness.

objective function, where the total length of the path, the error of the manipulator to its goal throughout the path and the value of the potential field to ensure that the path is free of collisions, are considered. In equation 8 the multi-objective fitness function is defined as:

$$f = \frac{1}{error_n \sum_{i=1}^{n} e^i \left(error_i\right) + \sum_{i=1}^{n} Po_i} \qquad (8)$$

where: n = total number of $\Delta\theta$'s sets

$error$ = Euclidean error to the goal

Po = Potential Field caused by obstacle

Equation 8 corresponds to the fitness function for the search of the goal defined by the combined potential functions of the goal and obstacles present in the workspace of the manipulator. The accumulated error is considered as well in order to optimise the length of the paths, as the algorithm returns the path with the lowest accumulated error, i.e. the shortest path among the possible solutions considered by the GA during its execution.

The initial length of the chromosomes in the population determines the maximum length of the path described by the manipulator. As the initial population is randomly generated, the initially described paths could take the manipulator towards obstacles or away from the goal when the chromosomes are evaluated. As each chromosome is evaluated, those paths that drive the manipulator closer to the goal as a whole and are free from collision are considered as potential parents accordingly to their fitness. If the initial size of the chromosomes is set to 100 alleles and during the evolution and evaluation of possible solutions a particular chromosome reaches the desired goal as a result of the displacements described on the first 70 alleles, the rest of the chromosome is ignored and the length of the chromosomes in the population for the following generation is reduced to that of the shortest described path. This mechanism is referred to as a Chromosome Length Reduction Mechanism (CLRM). As the length of the chromosomes in the population is reduced, as a result of the aforementioned mechanisms, the number of operations involved in the process of decoding the Chromosomes for the evaluation of the fitness is reduced, thus reducing the time required by the algorithm to produce a solution.

The example illustrated in Figure 13 shows a 2 dof manipulator with an initial configuration close to a line obstacle and a goal is set at (1.8, 0.5). It also shows the evolution of the solution from the GA illustrating the best path obtained after 1, 15, 30 and finally after 75 generations for an initial population of 500 chromosomes with a length of 800 alleles. The path obtained after 75 generations is given by a chromosome with a final length of 438 alleles. Figure 14 shows the final path obtained after 18 generations and some best paths obtained in intermediate generations to illustrate the evolution of the solution. Figures in 15 show the manipulator along the obtained path for case I, while Figure 16 shows the average fitness and the fitness of the best solution for each generation.

This approach can be extended to consider a greater number of degrees of freedom in the system by simply modifying the chromosome structure presented in Figure 15 to include an m number of $\Delta\theta$'s as illustrated in Figure 18.

By increasing the number of variables in the system, the length of the chromosomes also increases and it is necessary to increase the size of the initial population to ensure that a suitable solution is found.

Figures 16 and 17 present a 3dof manipulator and three static obstacles in the workspace. Figure 16 illustrates the final path (solid) obtained by the global path planning algorithm and an intermediate path obtained in an earlier generation (dashed). The initial configuration of the manipulator is at (–90°,0°,0°) and its goal at (1.9, 0.25). The initial size of the chromosomes in the initial population for this case was set to 1200 alleles and the final size of the best path obtained was

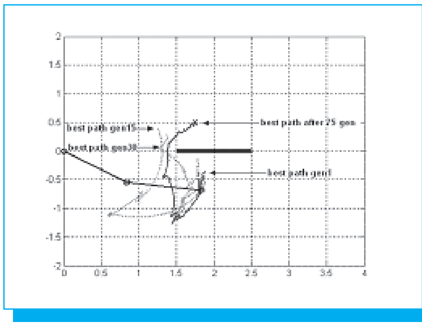


Fig. 13. Robot manipulator and paths obtained at different generation case I.

212 alleles after 850 generations and 3 562 seconds, with an initial population of 500 chromosomes.

Figure 31 illustrates the same system for an initial configuration at (–31°, 61°, –31°) and same goal. The final path is described in a 366 alleles string obtained after 837 generations.

Tables 1 and 2 summarize the performance of the GA global planner for robot manipulators with variations in the initial chromosome length and in the size of the initial population. The algorithm returns the best path found after the average fitness of the population has become stable and the fitness of the best candidate solution has remained the same for the last 10 generations. The results reported were obtained using a PC with a Pentium IV processor at 1.8 GHz.

The performance of the global path planner indicates that the chromosome length reduction mechanism improves the performance in terms of required time of solution. The effect of the mechanism has a greater impact when the initial length of the chromosome and the size of the population are increased.

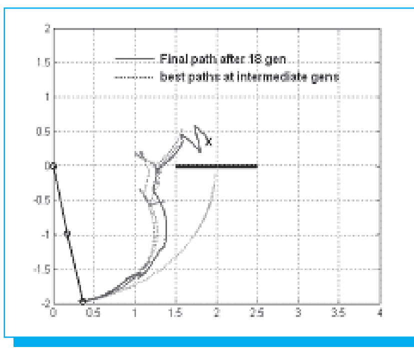
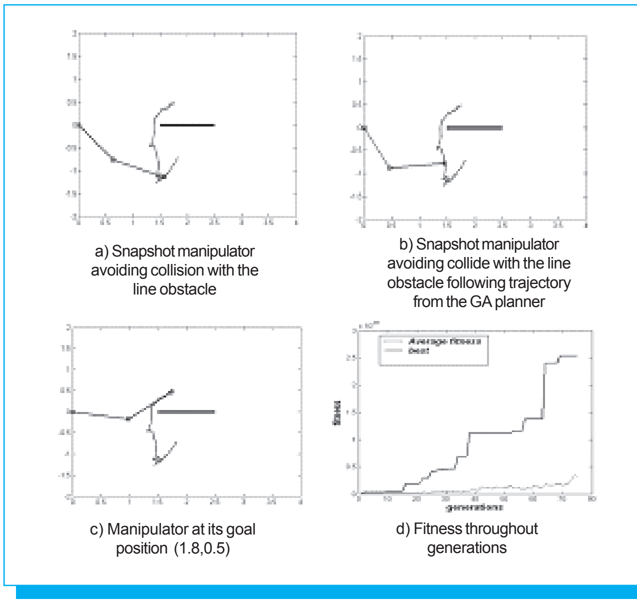


Fig. 14. Robot manipulator and paths obtained at different generations case II.



a) Snapshot manipulator avoiding collision with the line obstacle

b) Snapshot manipulator avoiding collide with the line obstacle following trajectory from the GA planner

c) Manipulator at its goal position (1.8,0.5)

d) Fitness throughout generations

Fig. 15. Snapshot and average fitness.

From tables 1 and 2, it can also be observed that the required number of generations to obtain a suitable path is related to the size and length of the initial population. When a large population is considered, the required number of generations decreases as more possible solutions are evaluated per generation. When the length of the chromosomes is reduced the necessary number of generations to obtain a path decreases as a set of configurations that drive the manipulator to the goal is to be evolved considering a reduced number of joint displacements.

5. Conclusions

This paper has introduced the necessary concepts to understand the trajectory planning problem of robot

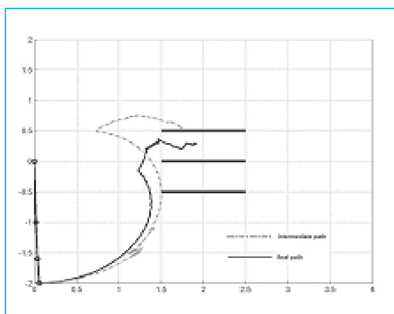


Fig. 16. Robot manipulator and paths obtained at different generations Case I.

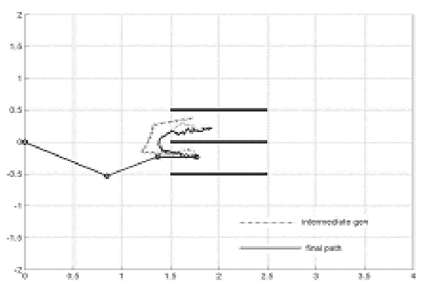


Fig. 17. Robot manipulator and paths obtained at different generation Case II

manipulators as well as soft-computing techniques. A GA based global path planner has been proposed for the solution of the problem in the workspace of a robot manipulator. Solving the problem directly in the workspace of the manipulator eliminates the process of mapping the obstacles into the C-space, thus reducing the time involved in the path planning process. The algorithm implements a Chromosome Reduction Mechanism to improve the search performance which results in both a decrement in the number of generations needed by the algorithm to return a suitable path and a reduction in the overall time of execution of the planner. Since the final length of the path that drives a manipulator from its starting configuration to its desired goal its unknown, it is not possible to specify the length of the Chromosomes in the initial population of the GA to find a suitable path, being necessary to set the length of the Chromosome arbitrarily. The implementation of the CLRM has proven to optimize the search capabilities of the GA for this particular implementation of path planning, as it allows the initial consideration of a large set of joint displacements that describe a random path of the manipulator and then, once the algorithm finds paths that reach the desired goal, the CLRM reduces the length of the Chromosomes of the following generation. The variation of the initial population size considered by the algorithm affects the number of generations required by the planner to return a feasible path, as can be seen in tables 1 and 2.

$$\Delta\theta\text{'s}\ [\ 1\ 1...1\quad 2\ 0...1\quad 0\ 1...2\quad 2\ 2...1\ \ \mathbf{...}\ \ 0\ 0...0\ ]$$

$$(\Delta\theta_1,\Delta\theta_2,...,\Delta\theta_m)_1\,(\Delta\theta_1,\Delta\theta_2,...,\Delta\theta_m)_2\,(\Delta\theta_1,\Delta\theta_2,...,\Delta\theta_m)_3\,(\Delta\theta_1,\Delta\theta_2,...,\Delta\theta_m)_4...(\Delta\theta_1,\Delta\theta_2,...,\Delta\theta_m)_n$$

Fig. 18. Chromosome structure for m degrees of freedom.

Table 1. GA global planner performance for 2 dof manipulator.

2 dof manipulator		without CLRM*		with CLRM*	
Case I					
Chromosome length (alleles)	Population size (chromosomes)	Solution after (generations)	time (seconds)	Solution after (generations)	time (seconds)
800	500	97	97	75	830
800	300	184	184	137	963
600	500	198	198	183	1419
600	300	378	378	310	1793
Case II					
800	500	45	45	18	239
800	300	62	62	56	394
600	500	33	33	22	197
600	300	76	76	47	390

*Chromosome Length reduction Mechanism

The simulation results for the algorithm extended to the case of a 3dof planar manipulator with 3 static obstacles indicate that, when the layout of the obstacles constrains the possible path to be described by the tip of the manipulator, the algorithm requires an increased number of generations to produce a path when compared to the number of generations required for the case of a 2 dof manipulator with one static obstacle.

Even though a considerable drop in the execution time was achieved in searching for a suitable path by the implementation of the CLRM, the overall execution time of the planner is not yet suitable for an implementation in real systems. The path obtained is affected by the imposed constraint for the values of $\Delta\theta = \pm 1°$ resulting in jerky movements of the tip of the manipulator along the described path.

The work presented in this first part, shows that the performance of the GA based global path planner is not yet ideal. In the second part to follow, the implementation of a GA based local path planer for robot manipulators is explored, where static and dynamic environments when robot manipulators share a common workspace are considered.

6. References

[1] R. Z. Tombropoulos, J. R. Adler, and J. C. Latombe, "CARABEAMER: A treatment planner for a robotic

Table 2. GA global planner performance for 3 dof manipulator.

3 dof manipulator		without CLRM*		with CLRM*	
Case I					
Chromosome length (alleles)	Population size (chromosomes)	Solution after (generations)	time (seconds)	Solution after (generations)	time (seconds)
1200	600	975	4357	850	3562
1200	300	1753	4528	1433	3897
600	600	1830	4607	1675	4265
600	300	1926	4418	2125	4337
Case II					
1200	600	875	3910	837	3508
1200	300	1614	4169	1523	4142
600	600	1961	4937	1865	4749
600	300	3649	8370	3588	7323

*Chromosome Length reduction Mechanism

radiosurgical system with general kinematics", Medical *Image Analysis*, vol. 3, pp. 237-264, 1999.

[2] E. Coste-Maniere, L. Adhami, F. Mourgues, O. Bantiche, D. Le, D. Hunt, N. Swarup, K. Salisbury, and G. Guthart, "Optimal Planning of Robotically Assisted Heart Surgery: Transfer Precision in the Operating Room", *8th International Symposium on Experimental Robotics*, pp. 721-732, 2003.

[3] K. Sugihara and J. Smith, "Genetic algorithms for adaptive motion planning of autonomus mobile robots", University of Hawaii, Manoa, *Technical Report* February 1997 1997.

[4] Y. K. Hwang and N. Ahuja, "Gross motion planning- A survey", *ACM Computing Surveys*, V24, pp. 219-291, 1992.

[5] T. Fraichard and C. Laugier, "Kinodynamic Planning in a Structured and Time-Varying 2D Workspace", *Proc. of the 1992 International Conference on Robotics and Automation*, pp. 1500-1505, 1992.

[6] J. C. Latombe, *Robot Motion Planning*. Boston, MA, USA: Kluwer Academic, 1991.

[7] K. Althoefer, "Neuro-fuzzy motion planning for robotic manipulators", in *Department of Electronic and Electrical Engineering*. London, UK: King's College London, Unuversity of London, 1996, pp. 178.

[8] T. Lozano-Peréz, "Spatial planning: A configuration space approach.", *IEEE Transactions on Computing*, vol. C-32, pp. 108-119, 1983.

[9] Y. Liu, "Interactive reach planning for animated characters using hardware acceleration", in *Computer and Information Science*. Pittsburg: Univesity of Pennsylvania, 2003, pp. 95.

[10] D. Lee, *The map-building abd exploration strategies of a simple sonar-equiped mobile robot: An experimental, quantitative evaluation*. UK: CUP, 1996.

[11] R. A. Brooks, "A roboust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, vol. 2, pp. 14-23, 1986.

[12] S. Fortune, G. Wilfong, and C. Yap, "Coordinated motion of two robot arms", *International Conference on Robotics and Automation*, 1987.

[13] J. T. Schwartz and M. Sharir, "On the 'piano movers' problem II: General techniques for computing topological properties of real algebraic manifolds", *Advances in Applied Mathematics*, vol. 4, pp. 298-351, 1983.

[14] E. Freund and H. Hoyer, "Collision avoidance in multi-robot systems", in *Robotics Research: Proc. of The Second International Symposium*, H. Hanafusa and H. Inoue, MA USA: MIT Press, 1985, pp. 135-146.

[15] B. H. Lee and C. S. G. Lee, "Collision free motion planning of two robots", *IEEE Transactions on Systems*, Man and Cybernetics, vol. 17, pp. 21-32, 1987.

[16] C. Chang, M. J. Chung, and B. H. Lee, "Collision avoidance of two general manipilators by minimum delay time," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, pp. 517-522, 1994.

[17] Z. Bien and A. Lee, "A minimum-time trajectory planning method for two robots", *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 414-418, 1992.

[18] J. Lee, "A dynamic programming approach to near minimum-time trajectory planning for two robots", *IEEE Journal of Robotics and Automation*, vol. 11, pp. 160-164, 1995.

[19] M. A. Ridao, J. Riquelme, E. F. Camacho, and M. Toro, "An evolutionary and local search algorithm for motion planning of two manipulators", *Journal of Robotic Systems*, vol. 18, pp. 463-476, 2001.

[20] L.Zadeh, "What is soft computing?," www.ece.nus.edu.sg/stfpage/elepv/softc_def.html, vol. 2003, 1997.

[21] J. H. Holland, *Adaptation in natural and artificial systems*: University of Michigan Press, 1975.

[22] D. E. Goldberg, "Computer-aided gas pipeline operation using genetic algorithms and rule learning (Doctoral dissertation)," University of Michigan, 1983.

[23] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, vol. 8, pp. 338-353, 1965.

[24] E. H. Mamdani, "Application of fuzzy algorithms for control of a simple dynamic plant", *Proc. IEE*, vol. 121, pp. 1585-1588, 1974.

[25] W. S. McCulloch and W. Pitts, "A Logical Calculus of Ideas Imminent in Nervous Activity", *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.

[26] P. Bessiere, J. Ahuactzin, E. Talbi, and E. Mazer, "The 'Adriane's Clew' algorithm: Global planning with local methods", in *Proceedings of the International Conference on Intelligent Robots and System: IEEE*, 1993, pp. 1373-1380.

[27] A. B. Doyle, *Algorithms and computational techniques for robot path planning*. Bangor: University of Wales, 1995.

[28] A. B. Doyle and D. I. Jones, "Robot path planning with genetic algorithms", *Proc. of 2nd Portuguese Conference on Automatic Control* (CONTROLO), pp. 312-318, 1996.

[29] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, pp. 90-98, 1986.

[30] J. Kim and P. K. Kolsha, "Real time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 338-349, 1992.

[31] C. I. Connolly and R. A. Grupen, "On the application of harmonic functions to robotics", *Journal of Robotc Systems*, vol. 10, pp. 931-946, 1993.

[32] C. I. Connolly, J. B. Burns, and R. Weiss, "Path planning using Laplace's equation", *Proc. of the 1990 IEEE International Conference on Robotics and Automation*, pp. 2102-2106, 1990.

[33] P. Vadakkepat, C. T. Kay, and M. Wang, "Evolutionary artificial potential fields and their application in real time robot path planning", presented at *Proc. of the 2000 Congress of Evolutionary Computation (CEC2000)*, San Diego, California, 2000.

Indicaciones para los investigadores que deseen publicar en la revista Científica

Estructura de los artículos

Dentro del artículo la numeración comienza a partir del 'Resumen' o el 'Abstract', según sea el caso. Ésta continúa de acuerdo a los siguientes elementos:

1. Resumen
2. Abstract
3. Introducción
4. Desarrollo
5. Conclusiones
6. Referencias

Título

El título debe ser lo más corto posible (sin utilizar abreviaciones de preferencia) y no exceder de tres líneas. Se sugiere escribirlo con el uso de mayúsculas y minúsculas.

Datos del autor

Éstos muestran el o los nombres de los autores, la dirección de la institución a la cual pertenecen, el teléfono, fax y correo electrónico (de contar con ellos).

Resumen

Debe contener 200 palabras como máximo. Consiste en una pequeña descripción de la investigación y una breve información de los resultados del trabajo que permite a los lectores identificar con rapidez los aspectos fundamentales del artículo y discernir la relevancia de éste.

Abstract

Se incluye un resumen en inglés si el artículo está en español o viceversa. Éste aparece inmediatamente después del resumen en inglés o español, según sea el caso.

Introducción

Expresa los antecedentes, límites, alcances y relevancia del problema. Plantea las premisas del mismo, y expresa el objetivo del trabajo. Se sugiere no extenderla como si fuera una exposición analítica.

Desarrollo

Se refiere al desarrollo del tema que puede ser teórico, experimental, teórico-experimental o la descripción de un nuevo diseño. Es la parte medular y está compuesta por el planteamiento del problema y análisis del mismo, mencionando los materiales, métodos y técnicas. Las subdivisiones de este apartado se dejan al criterio del autor.

Conclusiones

Establece la respuesta global del problema, son los objetivos alcanzados, las hipótesis comprobadas, modificadas o rechazadas.

Referencias

Es la lista de fuentes bibliográficas: libros, artículos, manuales, memorias, etcétera. Deben aparecer en el orden en el cual se mencionan dentro del artículo.

Anexos

Los agradecimientos, simbología, notación y otros anexos, se consideran dentro del cuerpo del artículo y se dejan a consideración del autor. Si el trabajo ha sido previamente presentado en alguna institución científica o realizado con la ayuda de una subvención o fondo especial se indica por medio de una nota de pie en la primera página.

Consideraciones sobre el formato general

Para obtener la mejor calidad al realizar la edición de los manuscritos en el formato de la revista, se recomienda que: *a*) el texto deberá ser capturado preferentemente en un procesador de textos (*word*); *b*) el tipo de letra del texto: *Times New Roman* de 10 puntos con márgenes superior e inferior y derecho de 2.5 cm, margen izquierdo de 3 cm a dos columnas con espaciamiento de 0.77 cm entre ellas; *c*) las ecuaciones y fórmulas numeradas para mayor claridad, acrónimos y abreviaciones deben definirse la primera vez que sean usados en el texto; *d*) los pies de las figuras deben estar debajo de éstas con la indicación 'Fig. 1'; además, el título de los ejes de las figuras a menudo es causa de confusión, por lo que se sugiere usar palabras en lugar de símbolos. Los encabezados de las tablas deben estar arriba de éstas; *e*) las figuras y tablas en el documento deben entregarse en archivos por separado (.tif, .jpeg), es importante que al pasar cualquier imagen por un escáner debe hacerlo con una resolución de 300 ppp (puntos por pulgada). De antemano gracias.

Fe de erratas: En el número 3 (julio-septiembre) del volumen 9 (2005), en el artículo 'New Proposal of Corrections Factors for DC Voltages' aparece el segundo autor como V. del Moral, el cual debió registrarse con el nombre de V. M. Salázar del Moral. Se ofrece una disculpa por los inconvenientes causados por esta omisión.

Information for Authors Submitting Papers to the Científica Journal

The manuscript must be prepared in following a Camera Ready format with all its section numbered starting with the abstract, as shown below

1. **Abstract**
2. **Resumen (Spanish abstract if it is possible).**
3. **Body of the manuscript**
4. **Reference list**
5. **Acknowledgements if any**
6. **Reference list**
7. **Appendix if any**

Title

Must be concise and no longer than 3 lines using Capital and lower case letters.

Authors Name and Affiliations

The authors name must be written below the title using a one column format staring with the given name followed by one or two family names, if two family names applies. Below the authors names must be written the affiliation including the address quality, fax, telephone number or email.

Abstract

The abstract with about 200 words must give a brief description about the research including some comments about the experimental or simulation results and conclusion.

Resumen (Spanish abstract)

It is desirable that, if it is possible, a Spanish abstract be provided.

Body of the Manuscript

The body of the manuscript must include an introduction in which the relevance of the research must be explained. A review of related research works by either, the same or another authors must be included.

The body of the manuscript also must include the theoretical aspects of the research, as well as experimental or simulation results is any, together with a Conclusions Section.

Format

All manuscripts must be written in letter size paper, only by one side, with the following requirements. a) It is recommended that, if possible, the text be written using a word processor WORD 6.0 or higher. b) The text must be written in two columns with a separation between them of 0.77cm., using a 10 points Times New Roman font or similar, with lower, upper and right margins equal to 2.5 cm and let margin equal to 3:0 cm. c) All the equations must be numbered and written using an equations editor. d) All symbols or abbreviations must be defined the first time that they be used in the text. e) All figures must inserted in the manuscript. f) All figures must be numbered and its captions must be inserted below them. IT is strongly recommended to use, when it be possible, words instead of symbols in the graphic axis. The table captions must be inserted above the corresponding table. g) All pictures and scanned figures must be high quality pictures for proper reproduction.

References

References must appear in the format given below. For multiple authors all family names and given names initials must be given. Titles of the journals must be all in lower case except the fist letter of each word. All references must be cited by number in brackets, in the order that they appear in the text.

Instituto Politécnico Nacional

Escuela Superior de Ingeniería Mecánica y Eléctrica

Científica: La Revista Mexicana de Ingeniería Electromecánica

Unidad Profesional Adolfo López Mateos

Edificio 1, Planta baja, Dirección, Col. Lindavista, CP. 07738, Tel. 5729 600 ext. 54518, Fax 55860758

email: revistacientifipn@yahoo.com.mx

El Instituto Politécnico Nacional

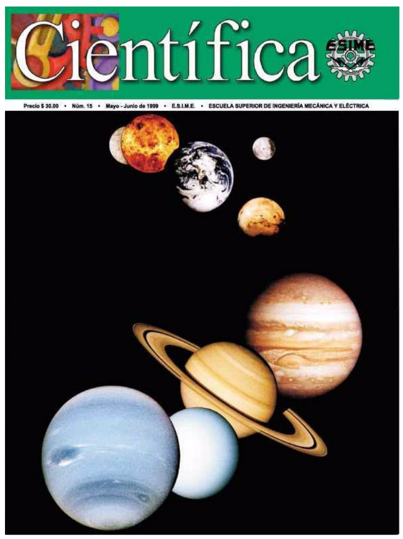
y la Escuela Superior de Ingeniería Mecánica y Eléctrica

publican

Científica

La Revista Mexicana de Ingeniería Electromecánica

Publicación incluida en el Índice de Revistas Mexicanas de Investigación Científica y Tecnológica del CONACyT



Suscripciones y venta de ejemplares:

Edificio 1, primer piso, Dirección ESIME Zacatenco, Unidad Profesional Adolfo López Mateos, Col. Lindavista, CP 07738, México, DF.

Tel. 5729 6000
exts. 54518/54555
email:
revistacientifipn@ yahoo.com.mx