*International Journal of Systems Science*
volume 35, number 3, 15 March 2004, pages 167–178

Taylor & Francis
Taylor & Francis Group

# Evolving a multiobjective obstacle avoidance skill of a seven-link manipulator subject to constraints

THRISHANTA NANAYAKKARA†, KEIGO WATANABE‡*, KAZUO KIGUCHI‡ and KIYOTAKA IZUMI‡

*A remarkable part of the skills of human experts rendering services in hazardous and dynamic factory environments is their ability to continue to perform a job while reacting to adapt their posture to sudden changes in the environment such as upcoming obstacles. Possession of these typical human skills that can be defined in a set of linguistic objectives will help to build flexible and more dexterous robot manipulators for factory automation systems that will meet the challenges of the growing complexity of the modern automated manufacturing environments. In this paper a novel soft-computing-based method is proposed for redundancy resolution of industrial robot manipulators to perform dexterous obstacle avoidance motions subject to the working constraints of the end effector. The new approach is based on evolving redundant rows of the Jacobian matrix of the manipulator and intelligent manipulation of user-defined kinematic functions for redundancy resolution. An evolutionary approach has been adopted to enhance the search efficiency in the objective space of a linguistic multi-objective problem (MOP) for the task of avoiding an obstacle moving towards the manipulator while keeping the end-effector position and orientation vector static. The objective functions of the MOP appear in the weighted sum of objectives in a gradually growing manner. The basic natural phenomena in the process of complex skill acquisition by human beings are reflected in the proposed approach. The effectiveness of the proposed method is demonstrated through experiments carried out using an industrial seven-link manipulator called PA-10, manufactured by the Mitsubishi Heavy Industries Ltd.*

## 1. Introduction

The new generation of industrial robots is needed not only to render routine services in static environments, but also in dynamic environments with moving objects where human-like posture adaptation skills are desired. This will meet the challenges of working in changing environments without making any change in the phase of the job being done. Particularly in telemanipulation applications, the slave side manipulator usually works in hazardous and uncertain environments where sudden changes such as obstacles moving towards the manipulator can well be expected. In such cases, the master side may have limited aid in terms of visual sensory-based information from the slave side or the master side may lack sufficient degrees of freedom to make an appropriate posture adaptation in the slave manipulator. Working in constrained environments such as space stations also poses a significant challenge on the ability of the manipulator to perform posture adaptations to suit the environment autonomously. To face this kind of situation, robot controllers should have more intelligence and autonomy to continue to perform the task while adapting the posture or the joint angular configuration to react to the environmental changes. From a mechanical point of view, redundant manipulators come in handy for this type of situation, because the extra degrees of freedom can be effectively exploited to handle posture adaptation in response to environmental changes.

Received 23 July 2001. Revised 22 January 2003. Accepted 19 February 2004.

† Graduate School of Science and Engineering, Faculty of Engineering Systems and Technology, Saga University, 1-Honjomachi, Saga 840-8502, Japan.

‡ Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering, Saga University, 1-Honjomachi, Saga 840-8502, Japan.

* To whom correspondence should be addressed.
e-mail: watanabe@me.saga-u.ac.jp

*International Journal of Systems Science* ISSN 0020–7721 print/ISSN 1464–5319 online © 2004 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207720410001684530

Among the very recent work done on redundancy resolution of manipulators, optimization of joint torques has been discussed by Chiaverini (1997) and Hollerbach and Suh (1987). Liang and Liu (1999) presented a method to design joint postures for redundant manipulators at different points of the end-effector trajectory. Optimizing joint accelerations near kinematic singularities using redundancy resolution has been discussed by Neil and Chen (1999). Configuration control approach based on augmentation of the hand Jacobian is also a viable approach for redundancy resolution (Seraji *et al.* 1993, Fiorini *et al.* 1997). This approach is based on the redundancy resolution at the position level through augmentation of the manipulator forward kinematics by a set of user-defined kinematic functions. The number of user-defined manipulation variables depends on the number of redundant joints in the manipulator. Li *et al.* (1998) have presented a method to control redundant manipulators subject to multiple criteria. Wang *et al.* (1999) have proposed a new method based on Lagrange networks for kinematic control of redundant manipulators. However, in these methods, the ability to incorporate linguistic conditions in the behaviour is very limited. In the case of using redundancy to acquire expert skills of human operators, it is important that the methodology provides room to incorporate the linguistic requirements or conditions imposed by humans.

Work has been done on developing human-like robot manipulation skills by Doeringer and Hogan (1998), Miyamoto and Kawato (1998) and Williamson (1998). Miyamoto and Kawato (1998) have presented an approach based on bidirectional theory for capturing the upswing motion of a tennis serving racket. This approach uses the trajectory data of a human expert's hand movement to develop the same skill in a robot manipulator. A discussion on the serial processing in human movement production has been given by Doeringer and Hogan (1998). This provides some important evidence about the working mechanism of the human mind that can be useful in robot manipulation skill development. The development of rhythmic arm movements such as spinning a crank using a neural network (NN) oscillator was presented by Williamson (1998).

The fact that PA-10 is a seven-link manipulator allows only one user-defined kinematic function to be introduced to augment the forward kinematics of the manipulator. The challenge in this problem is to use this one user-defined function to possess as many natural features of an obstacle avoidance movement of a real human arm as possible. The approach discussed in this paper attempts to solve this problem by adopting an evolutionary approach to optimize a set of linguistic objective functions to evolve a practically meaningful motion in redundancy resolution for the task of avoiding an obstacle moving towards the manipulator while keeping the end-effector position and orientation vector static. Using the proposed method, the dexterous motions of human experts can be realized by properly designing a multiobjective problem (MOP) based on linguistic expressions to characterize their motions. Then a Pareto-optimal front can be generated using the evolutionary algorithm (EA) where, if a set of solutions of the MOP consists of the decision vectors for which the corresponding objective vector cannot be improved any further without the deterioration of any of the components of the objective vector, then it is called 'Pareto optimal'.

A fuzzy neural network (FNN) is evolved to control the user-defined kinematic function, and a radial basis function neural network (RBF/NN) is used for adaptively calculating the corresponding complementary row of the Jacobian matrix of the manipulator, using an EA (Nanayakkara *et al.* 1999a, b). The off-line evolution of the FNN and the RBFNN tries to maximize a *gradually growing* multiobjective fuzzy-set-based objective function that contains the features required in the skill to be acquired. They include that the joint velocity commands generated to avoid an obstacle should lie in the null space of the end-effector position and the orientation space, that the minimum distance of the elbow of the manipulator should be kept with the obstacle and that the optimum movement conditions should be applied on each joint of the manipulator during the motion. Therefore this problem belongs to the class of MOPs (Liu *et al.* 2000, Tahk and Sun 2000). Among the work found in the recent literature on MOPs, Yun *et al.* (2001) discussed a method called generalized data envelopment analysis combined with genetic algorithms (GAs) to generate efficient frontiers in MOPs. It provides many Pareto-optimal solutions in a small number of generations of a GA and can be applied to convex as well as non-convex functions. However, the method faces difficulties with more than three objective functions. A multiobjective optimization technique for optimizing the parameters of a fuzzy supervisory controller for an injection moulding process was presented by Vegelatos *et al.* (2001). The method is characterized by taking the weighted sum of the objective functions in the MOP to allow prioritization of objectives. Rajesh *et al.* (2001) have used a method based on an adapted non-dominated sorting GA to solve the MOP of simultaneous maximization of product hydrogen and export steam flow rates of steam reformer plants. An evolutionary approach for path planning of mobile robots based on a weighted multiobjective function has been discussed by Xiao *et al.* (1997). The weighted sum of the objective functions for total distance of travelling, smoothness of the path and distances to obstacles is considered.

Actual experiments were performed using an industrial seven-link redundant manipulator called PA-10, to prove the effectiveness of the proposed method.

The rest of this paper is organized as follows. Section 2 presents some issues related to skilful obstacle avoidance motion in robot manipulators. The new way of designing of the objective function and the basic features of the evolutionary algorithm used are elaborated in section 3. Section 4 presents the experimental results using the PA-10 manipulator and finally, in section 5, conclusions are given.

## 2. Evolution of the skilful obstacle avoidance motion

### 2.1. *Configuration control approach for redundancy resolution*

The skilful motions of robot manipulators can be seen as performing a given task with human-like body motions. Normally the robot can be employed to perform a job at the end effector, using its end-effector Cartesian position and orientation control. Yet, in skilful motions, the extra features desired in the motion should be realized using the redundancy of the manipulator. A viable technique for redundancy resolution and motion control of redundant manipulators has been proposed by Seraji *et al.* (1993). This approach is based on the redundancy resolution at the position level through augmentation of the manipulator forward kinematics by a set of user-defined kinematic functions such as $\psi(\boldsymbol{\theta}) = \{\psi_1(\boldsymbol{\theta}), \ldots, \psi_r(\boldsymbol{\theta})\}$, where $r$ is the number of redundant manipulator joints. This is different from the conventional methods of using the pseudoinverse of the Jacobian to resolve the redundancy at the velocity level. In this approach the six manipulator position and orientation coordinates $Y = [p_x, p_y, p_z, \alpha, \beta, \gamma]^{\mathrm{T}}$ are augmented by the user-defined kinematic function $\psi(\boldsymbol{\theta})$ to yield the $7 \times 1$ configuration vector $\boldsymbol{r} = [Y^{\mathrm{T}}, \psi(\boldsymbol{\theta})]^{\mathrm{T}}$. Then the redundancy resolution can be expressed by $\psi(\boldsymbol{\theta}) = \psi_{\mathrm{d}}(t)$ that will be accomplished simultaneously with the basic task of controlling the hand motion $Y(\boldsymbol{\theta}) = Y_{\mathrm{d}}(t)$, where $p_x$, $p_y$, $p_z$, $\alpha$, $\beta$, $\gamma$ are the positions in the $x$, $y$, $z$ directions of the tip and the orientation variables of yaw, pitch and roll of the end effector respectively. $\psi_{\mathrm{d}}(t)$ and $Y_{\mathrm{d}}(t)$ are the desired time variations of $\psi(\boldsymbol{\theta})$ and $Y(\boldsymbol{\theta})$ respectively.

Then the augmented differential kinematics model of the manipulator is obtained (Hollerbach and Suh 1987, Seraii *et al.* 1993):

$$\dot{\boldsymbol{r}}(t) = \begin{pmatrix} J_{\mathrm{e}}(\boldsymbol{\theta}) \\ \vdots \\ J_{\mathrm{c}}(\boldsymbol{\theta}) \end{pmatrix} \dot{\boldsymbol{\theta}}(t) = J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t), \tag{1}$$

where $J_{\mathrm{e}}(\boldsymbol{\theta})$ is the $6 \times 7$ hand Jacobian matrix, $J_{\mathrm{c}}(\boldsymbol{\theta}) = \partial\psi(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ is the $1 \times 7$ Jacobian matrix associated with the kinematic function $\psi(\boldsymbol{\theta})$ and $J(\boldsymbol{\theta})$ is the $7 \times 7$ augmented Jacobian matrix. Note that $J_{\mathrm{e}}(\boldsymbol{\theta})$ can be found in the paper by Wang *et al.* (1999).

Assuming that $\det[J(\boldsymbol{\theta})]$ is not-null, the controller with joint velocity commands can be given by

$$\dot{\boldsymbol{\theta}}_{\mathrm{d}} = [J^{\mathrm{T}}(\boldsymbol{\theta}) W_{\mathrm{t}} J(\boldsymbol{\theta})]^{-1} J^{\mathrm{T}}(\boldsymbol{\theta}) W_{\mathrm{t}}(\dot{\boldsymbol{r}}_{\mathrm{d}} + K(\boldsymbol{r}_{\mathrm{d}} - \boldsymbol{r}_{\mathrm{o}})), \tag{2}$$

where $\dot{\boldsymbol{\theta}}_{\mathrm{d}}$ is the manipulated joint velocity control command vector, $W_{\mathrm{t}}$ is the task error weighting matrix, $K$ is the gain matrix, and $\boldsymbol{r}_{\mathrm{d}}$ and $\boldsymbol{r}_{\mathrm{o}}$ are the desired and measured manipulator responses respectively of the configuration vector in the paper by Fiorini *et al.* (1997). It should be noted here that, when applying the velocity control command given in equation (2) to equation (1), it results in $\dot{\boldsymbol{\theta}}(t) = \dot{\boldsymbol{\theta}}_{\mathrm{d}}$ for the input implementation and in $\boldsymbol{r}_{\mathrm{o}} = \boldsymbol{r}(t)$ for the output measurement. This controller is depicted in figure 1.

The challenge in this method is the designing of the user-defined kinematic function to suit a given requirement. In the paper by Seraji *et al.* (1993), user-defined kinematic functions are designed for applications such as optimum joint movement, collision avoidance and optimum elbow motion. It is apparent from these examples that the designing of these kinematic functions using mathematical equations is a very difficult task and lacks flexibility to incorporate complex linguistic features that a human arm movement would have. Yet, skilful manipulator motions require human-like body motions that inherently possess many characteristic features learnt through conscious training for a long period of time. Some of the salient features of a skilful human motion would be maintaining smooth trajectories, maintaining optimum motions in the sense that the motions do not overreact to sensory inputs, performing parallel tasks, etc. Capturing these human-like characteristic features to develop a controller to react to avoid a moving obstacle without inhibiting the task at the end effector is the objective of this study. Therefore the requirement for using a more appropriate soft-computing-based approach is apparent from the above discussion. The following

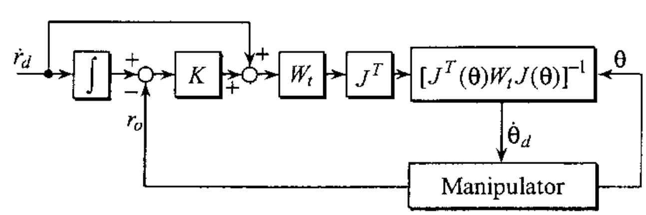



**Figure 1. Basic configuration controller.**

section discusses how the proposed method effectively addresses the challenge of designing the user-defined kinematic function and the corresponding row of the Jacobian matrix, to realize human-like maipulator motion in an obstacle avoidance application.

### 2.2. *Fuzzy neural network and radial basis function neural network based obstacle avoidance using redundancy resolution*

To achieve the objective of developing a user-defined kinematic function and the corresponding row in the Jacobian matrix that will result in a dexterous manipulator motion reflecting some key features of a human arm motion, the method adopted here has two key features: one is that, for redundancy resolution, the user-defined kinematic function $\psi(\boldsymbol{\theta})$ is given by the output of an FNN, and the other is that the corresponding row in the Jacobian matrix $J_c(\boldsymbol{\theta})$ is provided by an RBFNN.

Then the configuration vector is given by

$$\boldsymbol{r} = [Y^{\mathrm{T}},\ \psi(\boldsymbol{\theta})]^{\mathrm{T}} = [p_x, p_y, p_z, \alpha, \beta, \gamma, \psi(\boldsymbol{\theta})]^{\mathrm{T}}. \tag{3}$$

A very special case could be that $\psi(\boldsymbol{\theta}) = \theta_1$, where $\theta_1$ is the angle of the first joint of the manipulator. For such a case, the corresponding row in the Jacobian matrix is given by

$$J_c(\boldsymbol{\theta}) = \frac{\partial\psi(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = [\,1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]. \tag{4}$$

Then the value of $\boldsymbol{\theta}_1$ can be controlled by an FNN to generate desired obstacle avoidance movement. Yet this method is restricted to obstacle avoidance movement, and general applications will face difficulties. Therefore in the proposed method an FNN is used to generate a general user-defined kinematic function $\psi(\boldsymbol{\theta})$ and an RBFNN with the joint angle vector of the manipulator as the input is employed to provide the corresponding row in the Jacobian matrix given by

$$J_c(\boldsymbol{\theta}) = \frac{\partial\psi(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = [NN(\boldsymbol{\theta})]. \tag{5}$$

The RBFNN is depicted in figure 2. In the proposed method, $J_c(\boldsymbol{\theta})$ is given as the output of an RBFNN according to

$$[J_c(\boldsymbol{\theta})]_k = [NN(\boldsymbol{\theta})]_k = \sum_{l=1}^{N} W_{kl}\phi_l(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}_l, \sigma_l), \quad k = 1, \ldots, 7, \tag{6}$$

where $[J_c(\boldsymbol{\theta})]_k$ is the $k$th element of the $J_c(\boldsymbol{\theta})$, $N$ is the number of radial basis functions (RBFs) that construct the NN and $W_{kl}$ is the weight from the $l$th RBF to $k$th output node as shown in figure 2. In addition,

$$\phi_l(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}_l, \sigma_l) = \exp\left[-\left(\frac{(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_l)^{\mathrm{T}}(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_l)}{\sigma_l^2}\right)\right], \tag{7}$$

where $\boldsymbol{\theta} \in \Re^7$ and $\bar{\boldsymbol{\theta}}_l \in \Re^7$ are the input joint angle vector and the vector of centres of the RBF respectively, and $\sigma_l^2$ is the variance of the $l$th RBF.

The construction of the controller needs a method to determine the value of $\dot{\psi}_d$. The proposed method uses an FNN so that the input of the FNN is the distance of the elbow of the manipulator to the obstacle and produces the output $\dot{\psi}_d$. The basic picture of the proposed controller is shown in figure 3.

The architecture of the FNN used in this application is shown in figure 4, where $g$ denotes the operator to invert the sum of membership values obtained in the membership layer. In this FNN, a simple fuzzy reasoning is applied using one piece of sensor information (Watanabe and Izumi 1998). The approximation problem of single-input single-output fuzzy systems has been discussed in detail by Zeng and Singh (1994),

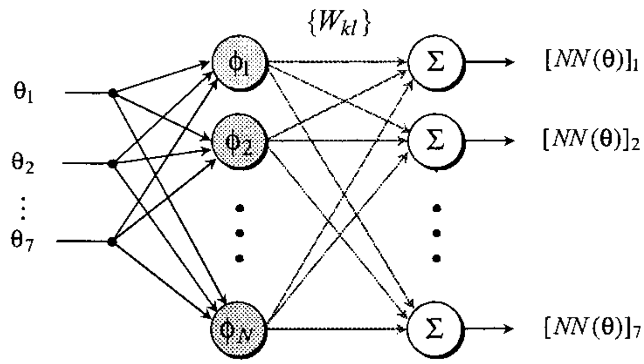



**Figure 2. The architecture of the RBFNN.**

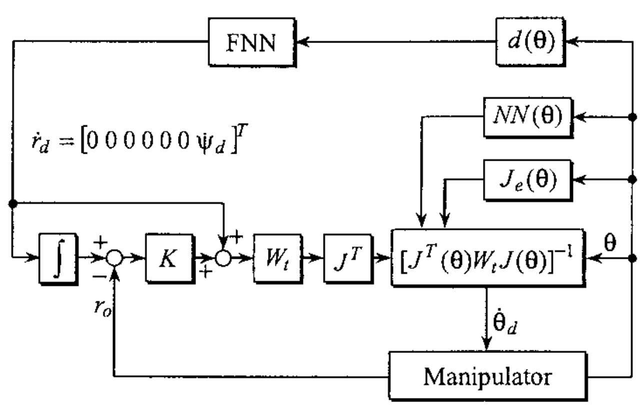



**Figure 3. Basic configuration controller for obstacle avoidance using an FNN.**

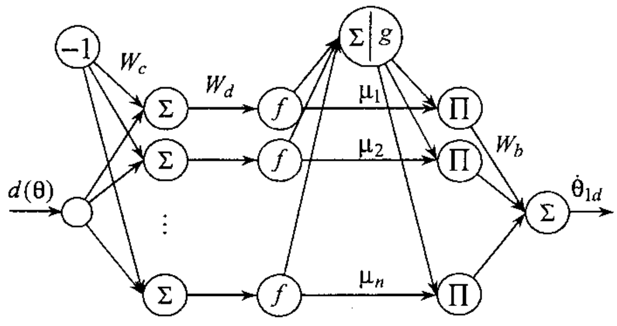



**Figure 4. The architecture of the FNN.**

in which the performance of the type of FNN adopted in this application is explained. The input $d(\boldsymbol{\theta})$ is the distance of the manipulator elbow to the obstacle, and the output $\dot{\psi}_d$ is the desired value of the manipulated variable. A Gaussian-type membership function $f$ is adopted. The resultant fuzzy reasoning is given by

$$\dot{\psi}_d = \sum_{i=1}^{n} p_i W_{bi}, \tag{8}$$

where $n$ denotes the total number of rules, which is set to 10 in this application so as to obtain a more refined reasoning result, $W_{bi}$ is the constant in the conclusion of the $i$th rule, and $p_i$ is the normalized rule confidence given by

$$p_i = \frac{\mu_i}{\sum_{j=1}^{n} \mu_i}, \tag{9}$$

$$\mu_i = \exp\{\ln(0.5)[d(\boldsymbol{\theta}) - W_{ci}]^2 W_{di}^2\}. \tag{10}$$

Here, $W_{ci}$ denotes the centre value (e.g. the mean value of a Gaussian-like membership function) associated with the $i$th membership function, and $W_{di}$ denotes the reciprocal value of the deviation from the centre $W_{ci}$ to which the $i$th Gaussian function of the input data on the support set has the value 0.5.

The working environment considered in this application imposes the following requirements. The end effector has to be kept static to perform a job and the elbow should avoid an obstacle moving towards it. This demands that the position and orientation be kept unchanged while the body adapts its posture to avoid the obstacle. This situation is ideal to explain the ability of a redundant manipulator to achieve multiple objectives simultaneously.

Therefore, in this application the desired rate of change of the configuration vector is given by

$$\dot{\boldsymbol{r}}_d = [\dot{Y}_d^T,\ \dot{\psi}_d]^T = [0,\ 0,\ 0,\ 0,\ 0,\ 0,\ \dot{\psi}_d]^T, \tag{11}$$

where $\dot{\psi}_d$ is the desired value of the user-defined manipulation variable.

## 3. Designs of the objective function and the evolutionary approach

### 3.1. *General features of multiobjective problems*

According to the discussion given in the above sections, the problem addressed in this paper is one of MOPs. An MOP can be defined as

$$\text{minimize/maximize} \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T,$$
$$\text{subject to}$$
$$\boldsymbol{x} \in S = \{\boldsymbol{x} \in \Re^{n_x} \mid g_j(\boldsymbol{x}) \leqslant 0, \quad j = 1, \ldots M\}, \tag{12}$$

where $\boldsymbol{x} = (x_1, \ldots, x_{n_x})^T$ is called the *decision vector*, $\boldsymbol{f}(\boldsymbol{x}) \in F$ is the *objective vector*, $F$ is the *objective space* and $S$ is the *parameter space*.

The important fact about MOPs is that there exists no requirement to obtain an optimal solution that optimizes all objective functions $f_i(x)$, $i = 1, \ldots, m$, simultaneously. Therefore the concept of an optimal solution based on the Pareto domination is given as follows.

**Definition** (Pareto-optimal solution)**:** A point $\hat{\boldsymbol{x}} \in S$ is said to be a Pareto-optimal solution to the MOP problem if there exists no $\boldsymbol{x} \in S$ such that $\boldsymbol{f}(\boldsymbol{x}) \leqslant \boldsymbol{f}(\hat{\boldsymbol{x}})$ if minimization or $\boldsymbol{f}(\hat{\boldsymbol{x}}) \leqslant \boldsymbol{f}(\boldsymbol{x})$ if maximization.

The set of solutions of the MOP consists of the decision vectors for which the corresponding objective vector cannot be improved any further without deterioration of any of the components of $\boldsymbol{f}(\boldsymbol{x})$. Such decision vectors are known as *Pareto-optimal.* Therefore, the best solution obtained for a weighted sum of objective functions is also Pareto optimal. The complete set of decision vectors that are Pareto optimal is known as the *Pareto-optimal set* or *Pareto-optimal front.*

### 3.2. *The multiobjective problem formulation of the proposed method*

The gradually growing objective function in this application is similar to weight control in MOPs with weighted sums of objectives (Xiao *et* al. 1997, Tahk and Sun 2000, Vegelatos *et al.* 2001). Weighted-sum-based multiobjective EAs such as the Hajela–Lin GA, and several other methods have been compared by Zitzler and Thiele (1999). Further examples have been given by Kim and Myung (1997) and Koziel and Michalewicz (1997) for constrained parameter optimization methods that use weight control in a weighted sum of a number of objective functions.

However, the weighted sum approach suffers from several drawbacks: firstly, it is computationally expensive: secondly, the optimal distribution of weights is not known to the user; thirdly, non-convex parts of the Pareto set cannot be found by minimizing convex combinations of the objectives. Thus, it is known that this approach does not give a particularly good idea of the shape for the Pareto-optimal surface if the objective functions are not convex, compared with other methods, such as weighted maximum or Pareto-based (non-dominated) approaches (Zitzler and Thiele 1999); here we merely adopt it as a basic formulation of the present method because of its simplicity. It is worth noting that the third drawback suggested above can be naturally overcome by using evolutionary approaches, which can handle non-convex problems.

In the proposed method, the component objective functions of the MOP are ranked according to their relative importance. Therefore the MOP growth pattern over the number of generations of the EA, is defined *a priori*. According to the proposed method, the dimension of the objective space expands in an incremental manner. While growing the MOP in an incremental manner, the best individuals in each stage are kept in the forthcoming generations. This kind of an MOP with a memory of the Pareto-optimal solutions in a sequence from lower dimensions to higher dimensions of the objective space is expected to have synergetic interactions among individuals of the EA to render a more exhaustive global search. In contrast, the conventional methods start from the highest dimension of the MOP and gradually search for a Pareto-optimal solution. This is expected to be computationally more expensive and prone to get stuck in sub-optimal solutions in the fitness landscape. Therefore, gradual growth of the MOP can help to obtain an efficient Pareto-optimal front with less computational burden. This fact is clearly reflected in the results obtained for the evolution of the FNN and the RBFNN shown in section 4.2.

The off-line training of the FNN and the complementary Jacobian RBFNN was carried out using the improved evolutionary algorithm (Nanayakkara *et al.* 1999a, b). The EA tries to maximize the multiobjective function with gradually increasing complexity, given by

$$J(G_n) = \begin{cases} J_1 = \dfrac{1}{t_N}\displaystyle\sum_{i=1}^{t_N} \exp\left(-20\{\dot{Y}^{\mathrm{T}}\dot{Y}\}\right) \quad \text{if } 0 < G_n \leqslant 200, \\ J_2 = \left(J_1 + \dfrac{1}{t_N}\displaystyle\sum_{i=1}^{t_N} \exp\left(-20\left\{[(P_{\mathrm{el}}(i) - P_{\mathrm{ob}}(i)) - \bar{P}]^2\right\}\right)\right) \\ \qquad\qquad \text{if } 200 < G_n \leqslant 400, \\ J_3 = \left(J_2 + \dfrac{1}{t_N}\displaystyle\sum_{i=1}^{t_N} \exp -[\boldsymbol{\theta}(i) - \boldsymbol{\theta}_0]^{\mathrm{T}} W_\theta [\boldsymbol{\theta}(i) - \boldsymbol{\theta}_0]\right) \\ \qquad\qquad \text{if } 400 < G_n \leqslant 500, \end{cases} \tag{13}$$

where $t_N$ is the total time span for obstacle avoidance in a single generation, $\dot{Y}$ is the rate of the manipulator end-effector position and orientation vector, $P_{\mathrm{el}}$ is the position of the elbow, $P_{\mathrm{ob}}$ is the position of the obstacle, $\bar{P}$ is the desired mean distance between the elbow and the obstacle, which is set to 8 cm in this application, $G_n$ is the number of the generations, $W_\theta = \mathrm{diag}(1, 50, 1, 50, 1, 50, 1)$ is the weight vector imposed on the movements of the joints of the manipulator and $\boldsymbol{\theta}_0$ is the initial joint angular values. The term $J_1$ of the objective function exerts pressure on developing the RBFNN and the FNN that results in a joint velocity command vector for elbow movement that lies in the null space of the end-effector position and orientation space. The right-hand side of the $J_2$ term serves to evolve an elbow movement that keeps a safe distance from the obstacle. The right-hand side of $J_3$ term helps to evolve an optimal joint movement by applying a higher penalty for pivot joint movements. This kind of gradually growing multiobjective evolution has the ability to evolve practically meaningful skills that will resemble the human arm movement skills that are essentially multiobjective optimal movements.

Therefore, it is clear from equation (13) that the control commands of the configuration controller in the joint velocity space should lie in the null space of the end-effector position and the orientation space. Furthermore, the configuration controller should generate control commands that result in an optimal obstacle avoidance motion in the sense that the motion characteristics follow some salient features of human arm motions.

The conventional experiments were carried out using all the terms of the objective function from the start of the evolutionary optimization process as given by

$$\begin{aligned} J(G_n) = {} & \frac{1}{t_N}\sum_{i=1}^{t_N} \exp\left(-20\{\dot{Y}^{\mathrm{T}}\dot{Y}\}\right) \\ & + \frac{1}{t_N}\sum_{i=1}^{t_N} \exp\left(-20\{[P_{\mathrm{el}}(i) - P_{\mathrm{ob}}(i)] - \bar{P}\}^2\right) \\ & + \frac{1}{t_N}\sum_{i=1}^{t_N} \exp\left(-[\boldsymbol{\theta}(i) - \boldsymbol{\theta}_0]^{\mathrm{T}} W_\theta [\boldsymbol{\theta}(i) - \boldsymbol{\theta}_0]\right). \end{aligned} \tag{14}$$

### 3.3. *The features of the evolutionary algorithm*

A new EA was proposed by Nanayakkara *et al.* (1999a, b). A brief description of the EA used is given in the following sections. The basic components of the new EA are depicted in figure 5.

*3.3.1. The representation.* The individuals are defined on the basis of real-valued vectors in the form

$$\boldsymbol{a} = (\boldsymbol{x}, \boldsymbol{\sigma}) \tag{15}$$

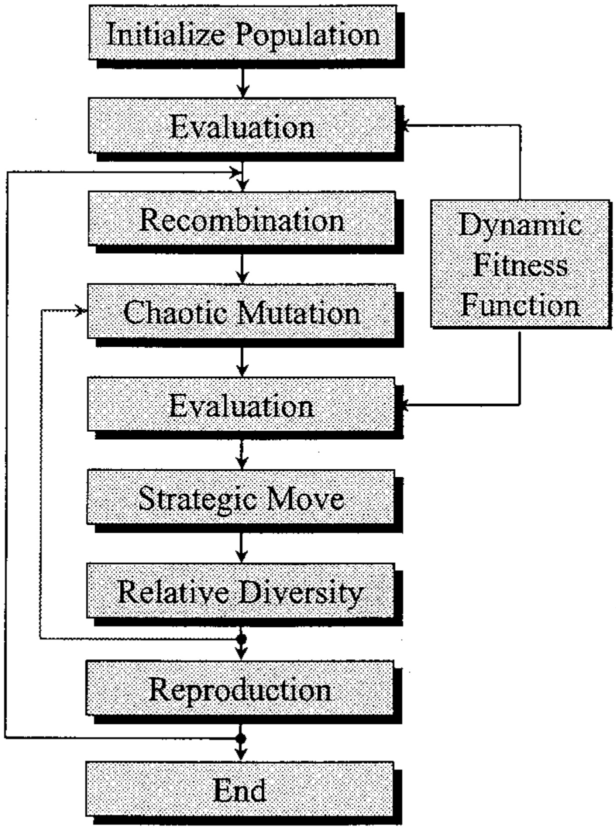



**Figure 5. The basic components of the new EA.**

to deal with continuous parameter optimization problems. Here $\boldsymbol{x} \in \Re^{n_x}$ is the object variable vector and $\boldsymbol{\sigma} \in \Re^{n_x}$ is the strategy parameter vector used for the mutation of individuals.

*3.3.2. Recombination.* In the proposed method, arithmetical crossover (uniform intermediate (Fogel and Ghozeil 1996)) is adopted with a relatively small probability (e.g. 0.2).

*3.3.3. Mechanism of mutation.* Mutation is performed to each individual in the form

$$x_i' = x_i + \sigma_i(\varsigma_k)N_i(0, 1), \tag{16}$$

where $x_i$ is the object variable vector, $N_i(0, 1)$ is a Gaussian random value generator with a normal distribution of zero mean and unity variance, and the chaotic standard deviation $\sigma_i(\zeta_k)$ is defined by

$$\sigma_i(\zeta_k) = (B_i^*|\xi_k|)^{1/2}, \tag{17}$$

$$B_i^* = c|B_i|, \tag{18}$$

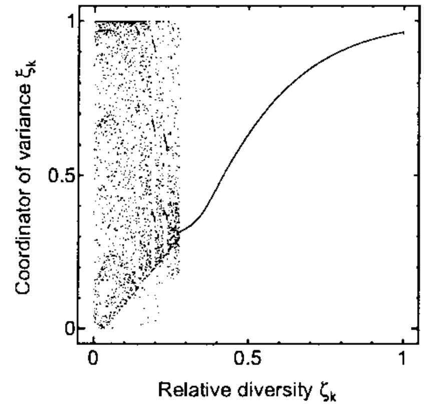



**Figure 6. The chaotic mutation mechanism.**

in which $B_i$ is the range of initialization of the object variable $x_i$, and $c$ is set to be 0.125. It should be noted that the value of $c$ can be altered depending on the degree of dynamics in the environment. The output $\xi_k$ of the chaotic neuron is shown in figure 6 (Aihara *et al.* 1990, Choi 1998). It can be seen in figure 6 that the output undergoes chaotic motions when the input $\zeta_k$ draws closer to zero. The parameter values adopted in this application were $K = 0.5$, $\beta_0 = 0.5$, $\beta_1 = 2.0$, $\beta_2 = 0.4$, $\beta_3 = 0.1$ and $\theta = 0.2$. For details of the dynamics of this neuron and the effect of the parameters on the characteristics, see Choi (1998).

**Definition:** The relative diversity $\zeta_k$ in the $k$th generation is defined as

$$\zeta_k = \frac{D_k}{D_0}, \tag{19}$$

where $D_k$ is the diversity of the population of the $k$th generation, which is given by

$$D_k = \frac{[F_{\max}]_k - \bar{F}_k}{[F_{\max}]_k}, \tag{20}$$

in which $[F_{\max}]_k$ is the fitness of the best individual in the population and $\bar{F}_k$ is the mean fitness of the population in generation $k$. This type of normalized diversity measurement is the most suitable for this application because it helps the EA to keep a track of the changing objective function.

This mechanism of mutation also achieves two objectives: one is that the algorithm has a low probability of becoming stagnated at a local minimum; the other is the rich diversity maintained even after convergence, which makes it possible to face potential changes in the environment.

*3.3.4. Adaptive strategic moves based on the local knowledge.* The algorithm uses knowledge of the

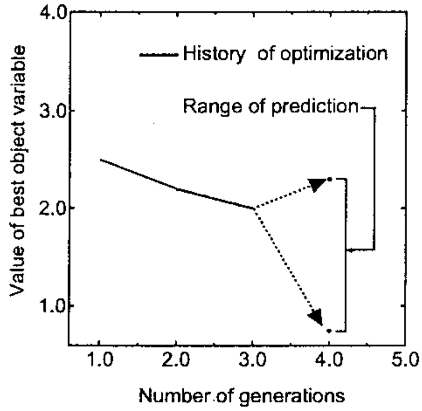



**Figure 7. The basic concept of strategic moves.**

optimization history to make strategic moves to enhance the search ability and the directivity. An example situation for a one-dimensional case is depicted in figure 7. Such usage of local knowledge in GAs has been employed by Yuret (1994). In this method, a short-term memory of $n_{\mathrm{p}}$ previous best individuals is kept. Then, in each generation, $\mu_{\mathrm{p}}$ strategic moves are made as

$$x_{ij}^{\mathrm{s}}(k+1) = U_{ij}(B_{\mathrm{L}}, B_{\mathrm{U}})x_{ij}^{*}(k) - \sum_{r=1}^{n_{\mathrm{p}}} x_{ij}^{*}(k-r), \qquad (21)$$

where $i = 1, \ldots, \mu_{\mathrm{p}}$, $j = 1, \ldots, n_x$, $\mu_{\mathrm{p}}$ is set to be a lower percentage of the population size $\mu$, $n_x$ is the total number of object variables in an individual, $x_{ij}^{\mathrm{s}}(k+1)$ is the $j$th element of the $i$th strategic estimate of the solution, $x_{ij}^{*}(k)$ is the best solution of the current generation, and $x_{ij}^{*}(k-1), \ldots, x_{ij}^{*}(k-n_p)$ are the best solutions up to $n_{\mathrm{p}}$ previous generations. $U_{ij}(B_{\mathrm{L}}, B_{\mathrm{U}})$ is a uniform random number generator with lower bound $B_{\mathrm{L}}$ and upper bound $B_{\mathrm{U}}$, resampled anew for each prediction.

Then, depending on the fitness ranking of the population, the lowest $\mu_{\mathrm{p}}$ individuals are replaced by the new estimates. The fact that many moves are made in a search space induced by the knowledge gathered from the recent history of optimization allows a wide search with a sense of directivity. This technique carries some sense of imagination due to human cognition as it uses a simple history of $n_{\mathrm{p}}+1$ best solutions from the history to make many moves into the future, leading to more sophisticated searching. One other advantage is that this helps to overcome local minima, because the searching never ends even if $n_{\mathrm{p}}+1$ consecutive generations give the same solution.

### 3.4. *Selection mechanism*

Tournament selection mechanism is adopted to produce the next generation. In the tournament selection

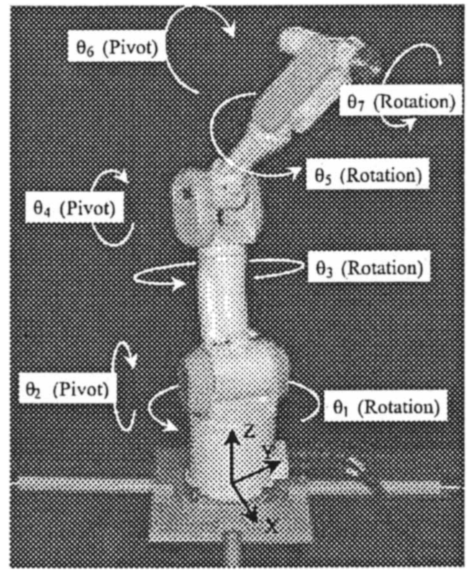



**Figure 8. The PA-10 seven-link manipulator used for the experiment.**

method for each individual $\boldsymbol{a}_j \in P(t) \cup P'(t)$, where $P(t) = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\mu\}$ is the parent population and $P'(t)$ is the population of mutated individuals, a random uniform sample of size $T$ is picked up from the population such that $T \geqslant 1$. Then a score $w_j \in (0, T)$ is attributed to each $\boldsymbol{a}_j$ such that the score is equal to the number of individuals in $T$ that is less fitter than the individual. Based on this score, the $2\mu$ individuals are ranked and the best $\mu$ individuals are selected for the next generation.

## 4. Experimental results and discussion

### 4.1. *Experimental set-up*

The PA-10 manipulator used in this experiment is shown in figure 8, which consists of seven links with seven degrees of freedom (three rotation axes and four pivot axes). The main specifications of the PA-10 manipulator are as follows: the length of the manipulator is 1345 mm, the weight of the manipulator is 35 kgf, the maximum combined speed with all axes is $1.55\,\mathrm{m\,s^{-1}}$, the payload weight is 10 kgf and the output torque is 9.8 N m.

In the experiments, it was assumed that the distance between the obstacle and the elbow of the manipulator is known. In the actual case there was no sensor attached to measure the distance. The obstacle moves towards the elbow from the Cartesian coordinate position ($x = 0.2214\,\mathrm{m}$, $y = -0.08\,\mathrm{m}$, $z = 0.7068\,\mathrm{m}$) at the speed ($\dot{x} = 0.0\,\mathrm{m\,s^{-1}}$, $\dot{y} = 0.08\,\mathrm{m\,s^{-1}}$, $\dot{z} = 0.0\,\mathrm{m\,s^{-1}}$). At the time of start, the PA-10 manipulator elbow position was at ($x = 0.2214\,\mathrm{m}$, $y = 0.0002\,\mathrm{m}$, $z = 0.7068\,\mathrm{m}$), and

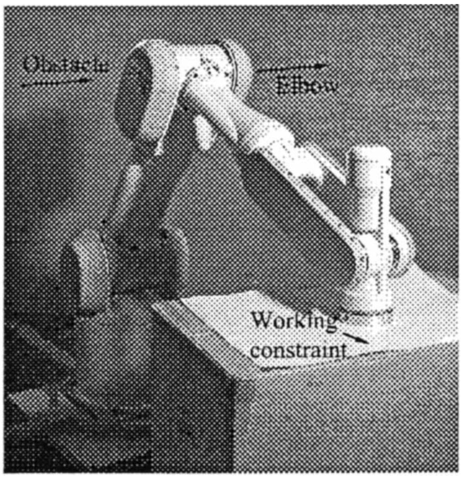



**Figure 9. The obstacle avoidance situation of the PA-10 manipulator in the experiment.**

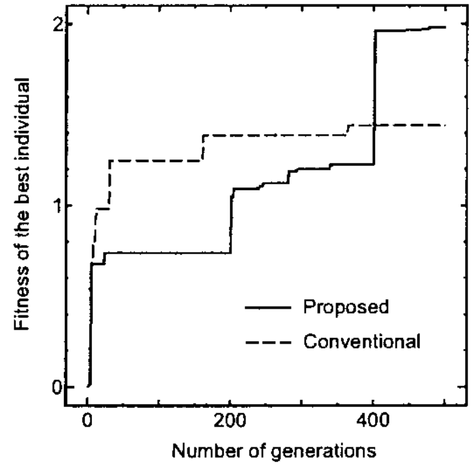



**Figure 10. Comparison of the evolutionary histories of best fitness.**

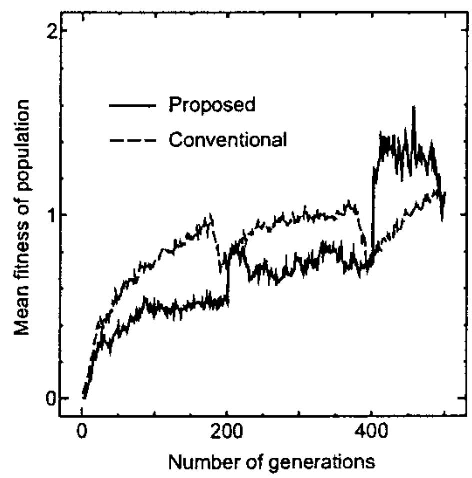



**Figure 11. Comparison of the evolutionary histories of mean fitness.**

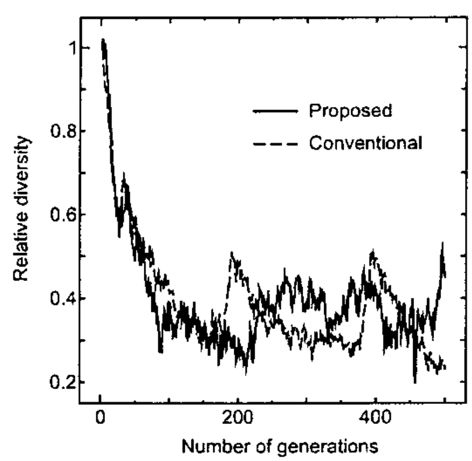



**Figure 12. Comparison of the evolution of the relative diversity.**

the joint angles at $\theta_1 = 0.000\,192$ rad, $\theta_2 = 0.514\,321$ rad, $\theta_3 = -0.000\,092$ rad, $\theta_4 = 1.564\,822$ rad, $\theta_5 = 0.000\,591$ rad, $\theta_6 = 1.037\,39$ rad and $\theta_7 = 0.000\,552$ rad. The experimental manipulator posture is shown in figure 9.

The ranges of initialization of the parameters of the FNN were $W_b = [-2, 2]$, $W_c = [0, 0.02]$ and $W_d = [0, 1]$ and these of the RBFNN were as follows: weights in $[-0.002, 0.002]$, centres of RBFs in $[-0.02, 0.02]$ and the variances of RBFs in $[0.1, 1]$. The population size was 50; 100% mutation rate and 20% recombination were adopted. The value of $\mu_p = 20\%$ and $n_p = 2$, as used by Nanayakkara *et al.* (1999a, b).

### 4.2. *Performance of the evolutionary algorithm*

The skilful obstacle avoidance movement was evolved for two cases: one is the proposed method of *gradually growing* objective function of the features of the skill and the other is the conventional method of including all the features from the beginning of the evolutionary optimization process. The evolutionary histories of the optimization of the FNN and the RBFNN for these two cases are compared in figure 10. Figure 11 presents the behaviours of the mean fitness for the two cases. In figures 10 and 11, jumps can be seen in the fitness of the best individual at 200 and 400 generations. This is because, in the gradually growing objective function, the new objectives that are added to the MOP at 200 and 400 generations are not perfectly mutually independent. The earlier objectives contain a part of the latter objectives of the motion characteristics. Therefore, when an objective is added, that part of the fitness in the new objective carried by the individuals becomes superimposed with the existing fitness that leads to a jump of total fitness. Figure 12 demonstrates the behaviour of the relative diversity defined in section 3.3.3. It should be noted that adequate relative diversity is kept in both cases, enabling the algorithm to keep the process of evolution from premature convergence (Leung 1997). This is achieved by the introduction of a chaotic mutation mechanism employing a chaotic neuron (Choi and Lee 1996, Choi 1998). The algorithm is fast in initial convergence owing to a strategic move mechanism in the algorithm. This mechanism tries to direct a few individuals to a region in the searching domain guided by the evolution of the best individual in the past generations. Yet it is clear that the proposed method shows superior performance because of the *gradually growing* objective function, against the complex objective function used from the beginning in the conventional method.

### 4.3. *Controller performance*

The configuration controller parameters in equation (2) were set as follows: $W_t = 2I_{7\times 7}$, and the error feedback gain $K = 0.2I_{7\times 7}$.

The comparison between the obstacle position and the elbow position is given in figure 13, for the time span of the experiment. It should be noted in figure 13 that the elbow remains at a distance of around 8 cm from the obstacle throughout the time span.

The behaviours of the position and orientation coordinates of the end-effector while avoiding the obstacle are given by figures 14 and 15. It is apparent from these figures that the end-effector motion is very low while the manipulator avoids the obstacle.

### 4.4. *General discussion*

This application demonstrates an idea that could be applied to improve the ability of redundant industrial manipulators skilfully to achieve several goals simultaneously. Incorporating the required linguistic features of the skills in the objective functions of an MOP and evolving the FNNs and RBFNNs can achieve this. It may also be adapted in an on-line manner. In this paper the evolved FNN implements a simple single-input single-output fuzzy reasoning technique to provide the desired variation in the user-defined kinematic function, and the evolved RBFNN provides the elements of the corresponding row in the Jacobian matrix. The key feature of the proposed method is that the behaviours of the FNN and RBFNN abide by multiple characteristic features that the designer wishes to have in the evolved skill. This has been achieved by designing a *gradually growing* fuzzy-set-based multiobjective evaluation function in the EA.

In this application, the obstacle moves towards the manipulator elbow at a constant velocity. Yet in a real practical situation it can have more complexity such as

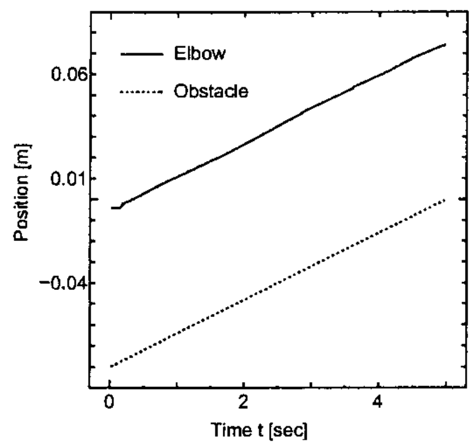



**Figure 13. The position of the elbow of PA-10 compared with that of the obstacle over time.**

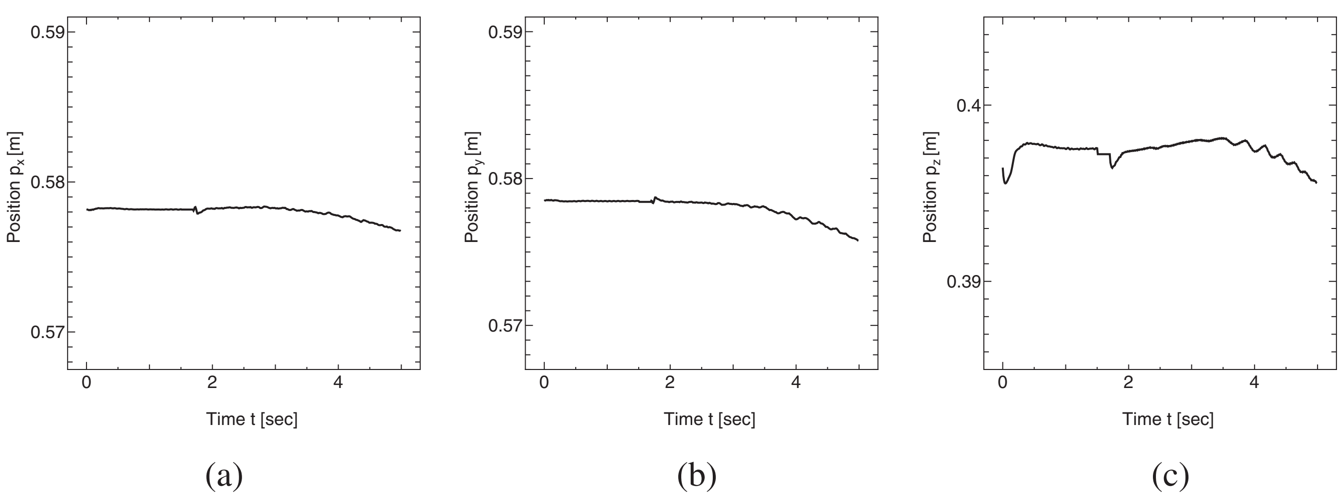
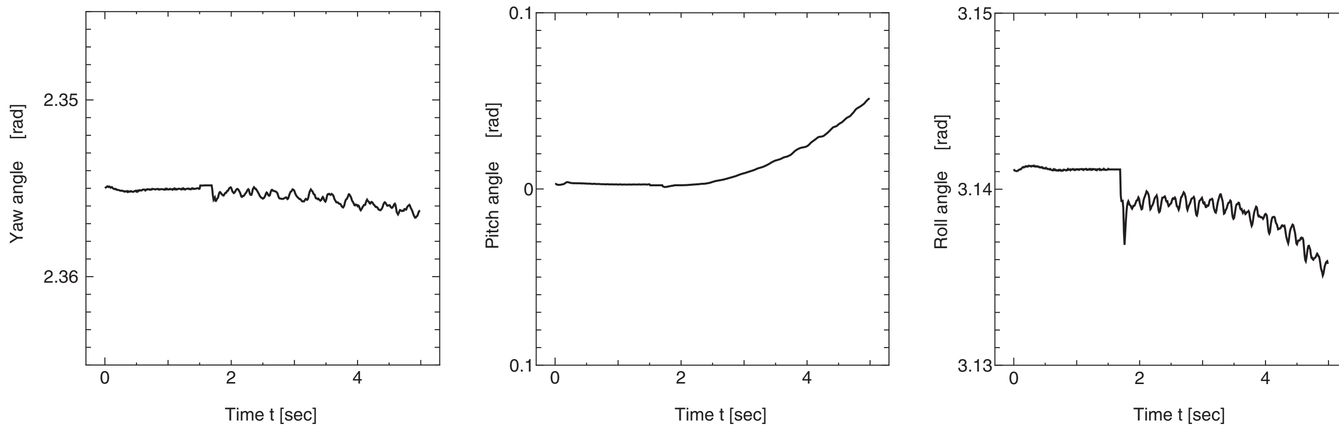



**Figure 14. The experimental end-effector behaviours: (a) position *x*; (b) position *y*; (c) position *z*.**




(a) (b) (c)

**Figure 15. The experimental end-effector behaviours: (a) yaw angle; (b) pitch angle; (c) roll angle.**

changing velocity or constant velocity that changes after some time. The fact that the IF–THEN rules of the trained FNN generalize the input–output mapping means that the proposed method is also expected to perform well in these situations. Moreover, the EA keeps enough population diversity as demonstrated in figure 12, which even helps it to deal with complex dynamic environments. In general, other types of skill could easily be developed using this method to resolve the redundancy of a redundant manipulator such as PA-10.

## 5. Conclusions

An evolutionary approach has been successfully employed in skilful redundancy resolution of PA-10 manipulator to avoid an obstacle while keeping the end-effector static. The key point of the proposed multi-objective fuzzy-set-based objective function is that the objectives of the MOP that are directly concerned with some of the features that the movement must possess are added to the main objective function one after another over the generations of the evolutionary optimization process. This kind of *gradually growing* objective function demands for the higher effectiveness of the EA to continue to evolve in dynamic environments. The new features of the chaotic mutation process and the mechanism of making strategic moves in the search space have proved to be effective in this challenging problem. Successful experimental results have been obtained for the industrial seven-link manipulator called PA-10. The new approach of designing fuzzy-set-based objective functions for human-like skill acquisition of robot manipulators using EAs improves the interface between the human experts and the mathematical optimization process, while imitating some important features of complex system evolution in the nature.

## References


Aihara, K., Takabe, T., and Toyoda, M., 1990, Chaotic neural networks. *Physics Letters* A, **144,** 333–340.

Chiaverini, S., 1997, Singularity robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, **13,** 398–410.

Choi, C., 1998, Dynamic bidirectional associative memory using chaotic neural network. PhD Thesis, Korea Advanced Institute of Science and Technology, South Korea.

Choi, C., and Lee, J.-J., 1996, Finding multiple local minima using chaotic jump. *International Journal of Intelligent Control and Systems*, **1,** 521–528.

Doeringer, J. A., and Hogan, N., 1998, Serial processing in human movement production. *Neural Networks*, **11**, 1345–1356.

Fiorini, P., Seraji, H., and Long, M., 1997, A PC-based configuration controller for dexterous 7-DOF arms. *IEEE Robotics and Automation Magazine*, **4,** 30–38.

Fogel, D. B., and Ghozeil, A., 1996, Using fitness distributions to design more efficient evolutionary computations. *Proceedings of the IEEE Conference on Evolutionary Computation*, Nagoya, Japan, 1996 (New York: IEEE), pp. 11–19.

Hollerbach, J. M., and Suh, K. C., 1987, Redundancy resolution of manipulators through torque optimization. *IEEE Journal of Robotics and Automation*, **3,** 308–316.

Kim, J.-H., and Myung, H., 1997, Evolutionary programming techniques for constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, **1,** 129–140.

Koziel, S., and Michalewicz, Z., 1997, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary Computation*, **7,** 19–44.

Leung, Y., 1997, Degree of population diversity—a perspective on premature convergence in genetic algorithms and its Markov chain analysis. *IEEE Transactions on Neural Networks*, **8,** 1165–1175.

Liang, T.-C., and Liu, J.-S., 1999, An improved trajectory planner for redundant manipulators in constrained workspace. *Proceedings of the IEEE International Conference on Robotics and Automation* (New York: IEEE) pp. 3153–3159.

Li, L., Gruver, W. A., Zhang, Q., and Chen, W., 1998, Real-time control of redundant robots subject of multiple criteria.

*Proceedings of the IEEE International Conference on Robotics and Automation* (New York: IEEE), pp. 115–120.

LIU, F.-H. F., HUANG, C.-C. and YEN, Y.-L., 2000, Using DEA to obtain efficient solutions for multi-objective 0–1 linear programs. *European Journal of Operational Research*, **126,** 51–68.

MIYAMOTO, H., and KAWATO, M., 1998, A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks*, **11**, 1331–1344.

NEIL, K. O., and CHEN, Y. C., 1999, Using redundancy to reduce accelerations near kinematic singularities. *Proceedings of the IEEE International Conference on Robotics and Automation* (New York: IEEE), pp. 31465–3151.

NANAYAKKARA, T., WATANABE, K., and IZUMI, K., 1999a, Evolving in dynamic environments through adaptive chaotic mutation. *Proceedings of the Fourth International Symposium on Artificial Life and Robotics*, Oita, Japan, 19–22, January 1999, Vol. 2, pp. 520–523.

NANAYAKKARA, T., WATANABE, K., KIGUCHI, K., and IZUMI, K., 1999b, Evolutionary optimization with strategic moves based on historical knowledge. *Proceedings of the First SOFT* (*Japan Society for Fuzzy Theory and Systems*) *Kyushu Branch Annual Conference*, pp. 21–28.

RAJESH, J. K., GUPTA, S. K., RANGAIAH, G. P., and RAY, A. K., 2001, Multi-objective optimization of industrial hydrogen plants. *Chemical Engineering Science*, **56,** 999–1010.

SERAJI, H., LONG, M. K., and LEE, T. H., 1993, Mortion control of 7-DOF arms: the configuration control approach. *IEEE Transactions on Robotics and Automation*, **9,** 125–139.

TAHK, M.-J., and SUN, B.-C., 2000, Coevolutionary augmented lagrangian methods for constrained optimization. *IEEE Transactions on Evolutionary Computation*, **4,** 114–124.

VEGELATOS, G. A., RIGATOS, G. G., and TZAFESTAS, S. G., 2001, Incremental fuzzy supervisory controller design for optimizing the injection molding process. *Expert Systems with Applications*, **20,** 207–216.

WANG, J., HU, Q., and JIANG, D., 1999, A Lagrangian network for kinematic control of redundant robot manipulators. *IEEE Transactions on Neural Networks*, **10,** 1123–1131.

WATANABE, K., and IZUMI, K., 1998, A fuzzy-neural realization of behavior-based control systems for a mobile robot. *Soft Computing for Intelligent Robotic Systems*, edited by L. C. Jain and T. Fukuda (Heidelberg: Physica), pp. 1–26.

WILLIAMSON, M. H., 1998 Neural control of rhythmic arm movements, *Neural Networks*, **11**, 1379–1394.

XIAO, J., MICHALEWICZ, Z., ZHANG, L., and TROJANOWSKY, K., 1997, Adaptive evolutionary planner/navigator for mobile robots. *IEEE Transactions on Evolutionary Computation*, 18–28.

YUN, Y. B., NAKAYAMA, H., TANINO, T., and ARAKAWA, M., 2001, Generation of efficient frontiers in multi-objective optimization problems by generalized data envelopment. *European Journal of Operational Research*, **129,** 586–595.

YURET, D., 1994, From genetic algorithms to efficient optimization. AI Technical Report 1569, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

ZENG, X.-J., and SINGH, M. G., 1994, Approximation theory of fuzzy systems—SISO case. *IEEE Transactions on Fuzzy Systems*, **2,** 162–176.

ZITZLER, E., and THIELE, L., 1999, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, **3,** 257–271.