# A genetic algorithm-based parameter-tuning algorithm for multi-dimensional motion control of a computer numerical control machine tool

## L-Y Kuo<sup>1</sup> and J-Y Yen<sup>2\*</sup>

<sup>1</sup>Mechanical Engineering Research Laboratory, Industrial Technology Research Center, Taichung, Taiwan <sup>2</sup>Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan

Abstract: This paper addresses an automatic parameter-tuning algorithm for the multi-axis motion control of a computer numerical control (CNC) machine centre. The traditional approach to tune the control parameters in the multi-axis machines is to tune each axis independently. Some high-end-precision machines offer cross-axis motion parameters for impedance compensation but this is usually not satisfactory for practical purpose. Because each axis on the machine centre contributes to more than one working plane, obtaining the optimal performance for motions involving more than one plane often results in axis coupling. This paper introduces a systematic method to tune the servo parameters for multi-axis motion control. The tuning algorithm is based upon an intelligent genetic algorithm (GA) and the parameters are tuned for each work plane. The method optimized the multi-axis motion performance. A modified GA is also proposed to solve the convergence problem induced by a large number of parameters in multi-axis motion tuning.

Keywords: motion control, machine tool control, compound genetic algorithm

## **1 INTRODUCTION**

The motion control algorithm in a computer numerical control (CNC) machine tool is very complex especially when the number of machine axes increases. There are usually a number of servo parameters that can be tuned for the proper motion characteristics, and the number of parameters adds up proportionally to the number of machine axes. Optimizing these parameters is a difficult task for the system engineers. A servo engineer can often tune the control parameters for one axis to achieve desirable performance, but it is not likely that he or she can tune the control parameters for multi-axis motion to give a similar performance. Some high-endprecision controllers offer a 'dual-axis circular test' to help to tune the servo performance within a certain plane, but there is still a lack of commercially available controllers with functions to tune motion with higher degrees of freedom. Some researchers try to tune separate axes and then use them together. This approach will not work because interaction between the axes is so severe and a noticeable tracing error would result. Basically the axes in a multi-axis system must be treated simultaneously to reach the required multi-axis control.

Even though many papers have been published on two-dimensional in-plane motion control, very few have addressed the control of higher-dimensional motion. Even fewer have addressed multi-axis machine tool servo systems. The expensive equipment involved may be one reason for the lack of research. From the literature, many papers have studied the control stability property in a machine tool servo system [1-8]. Recent developments by machine intelligence researchers have also incorporated artificial intelligence into the machine servo system [9–15]. These new results included the successful use of machine intelligence to tune the feedback servo parameters automatically, and some of the intelligent systems were able to generate control rules on their own [16-19]. However, these results were all restricted to the stability of the position control loop. Tarng et al. [19] have addressed the computer numerical controller as a system, but they have focused their work on controlling the feed rate to generate the best cutting results. The work by Kuo and Yen [20] has addressed the servo parameter tuning of a machine tool controller, but the work is also restricted to the in-plane motion of the machine.

The MS was received on 22 February 2001 and was accepted after revision for publication on 4 October 2001.

<sup>\*</sup>Corresponding author: Department of Mechanical Engineering, National Taiwan University, No. 1 Roosevelt Road, Sec. 4, Taipei, Taiwan 10617.

This research tries to use machine intelligence to help to tune higher-dimensional multi-axis motion. The genetic algorithm (GA) rule is used in the tuning process. To consider the effect of a multi-axis system, a novel approach to deal with the interference between the various axes is introduced. A five-axis machine centre involves sufficient complexity to serve as the multi-axis test station. The controller is developed in an accompanying project supported by the Ministry of Economic Affairs, Taiwan. It features an open structure personal computer-based controller that works with a Siemens 840D computer numerical controller [21]. Siemens offers a software interface, the so-called 'OEM Package', through which software developed in house can communicate with the Siemens 840D controller.

In the following, Section 2 will provide a brief introduction to the computer numerical controller and the effects of the motion control parameters in interest. Section 3 will describe the improved GA tuning method and how to obtain the final motion control parameter values. The experimental results and some discussions about the data will be presented in Section 4, and Section 5 will present the conclusions.

## 2 THE CONTROL LAW AND IMPORTANT MOTION CONTROL PARAMETERS

The experiment is based on a five-axis machine centre developed in a 'High speed machining of complex surfaces' project sponsored by the Ministry of Economic Affairs, Taiwan. The computer numerical controller itself is a Siemens 840D with 2 ms interpolating time. Because of the open architecture offered by the Siemens 840D, the present authors could design supervisory control software to incorporate on-line parameter tuning. The structure of the Siemens controller in use is shown in Fig. 1.

The computer numerical servo controller generally includes three basic servo loops. They are the *position control loop*, *speed control loop* and *current control loop*. Because motion control is achieved by consecutive positioning, this research will focus on the tuning of the position control. The friction compensation parameters are also considered, even though they belong to the current control loop, because the friction effect causes 'spikes' in the trajectory when there is a change in the driving axes. Therefore, the parameters of interest are the position servo gains, the feedforward gains, the dynamic response time constants and the friction compensation time constants within the speed control loop. To understand the tuning process, a brief description of these parameters is provided.

## 2.1 The effect of the position servo gains and feedforward gains

The computer numerical controller tries to track the tool path given by the part program command. Since the tracking error determines the quality of the machine tool control, it is necessary to analyse the relationship between the tracking error and the servo parameters.

Let  $K_{\rm f}$  denote the feedforward gain in the position control loop and let  $K_{\rm p}$  denote the position feedback gain. The dynamic response effect is temporarily neglected because the analysis is based on only one axis. The transfer function of the simplified system for each axis is thus

$$G(s) = \frac{K_{\rm f}s + K_{\rm p}}{s + K_{\rm p}} \tag{1}$$

where s is an operator of system model in the frequency domain. In general, the interpolated data (pulses) may be further divided by the control hardware (probably with the digital differential analysis method); however, it will be assumed that the input is a group of digital data for simplicity. The input datum points are denoted  $R_1$ ,  $R_2$ ,  $R_3$ , ... and u(t) represents the unit step function. The Laplace transform of the input signal is thus

$$R(s) = \frac{1}{s} [R_1 + (R_2 - R_1) e^{-T_1 s} + (R_3 - R_2) e^{-T_2 s} + \dots + (R_n - R_{n-1}) e^{-T_{n-1} s} - R_n e^{-T_n s}]$$
(2)

where s is the Laplace operator,  $T_1, T_2, \ldots$  are the time instants when the input datum points  $R_1, R_2, \ldots$  are introduced.

Because the output transfer function is Y(s) = R(s)G(s),

$$Y(s) = \frac{1}{s} [R_1 + (R_2 - R_1) e^{-T_1 s} + (R_3 - R_2) e^{-T_2 s} + \dots + (R_n - R_{n-1}) e^{-T_{n-1} s} - R_n e^{-T_n s}] \times \left(\frac{K_f s + K_p}{s + K_p}\right)$$
(3)



Fig. 1 Block diagram of the computer numerical controller



Fig. 2 The command-time relationship

Consider the speed command of the axis, v(t), and let  $\varepsilon$  represent the steady state error (Fig. 2). Then

$$\begin{aligned} w(t) &= F_1[u(t) - u(t - T_1)] \\ &+ F_2[u(t - T_1) - u(t - T_2)] \\ &+ F_3[u(t - T_2) - u(t - T_3)] \\ &+ F_{n-1}[u(t - T_{n-2}) - u(t - T_{n-1})] \\ &+ F_n[u(t - T_{n-1}) - u(t - T_n)] \end{aligned}$$
(4)

Since v(t) = dR(t)/dt and V(s) = R(s)s and therefore

$$Y(s) = G(s)R(s) \tag{5}$$

the time domain response can now be derived as

$$\begin{aligned} w(t) &= L^{-1}[Y(s)] \\ &= \left(\frac{K_{\rm f}-1}{K_{\rm p}} + t - \frac{K_{\rm f}-1}{K_{\rm p}} \,{\rm e}^{-K_{\rm p}t}\right) F_{\rm l}u(t) \\ &+ \left[\frac{K_{\rm f}-1}{K_{\rm p}} + (t - T_{\rm l}) - \frac{K_{\rm f}-1}{K_{\rm p}} \,{\rm e}^{-K_{\rm p}(t - T_{\rm l})}\right] \\ &\times (F_{2} - F_{\rm l})u(t - T_{\rm l}) \\ &+ \left[\frac{K_{\rm f}-1}{K_{\rm p}} + (t - T_{2}) - \frac{K_{\rm f}-1}{K_{\rm p}} \,{\rm e}^{-K_{\rm p}(t - T_{2})}\right] \\ &\times (F_{3} - F_{2})u(t - T_{2}) \\ &+ \left[\frac{K_{\rm f}-1}{K_{\rm p}} + (t - T_{n-1}) - \frac{K_{\rm f}-1}{K_{\rm p}} \,{\rm e}^{-K_{\rm p}(t - T_{n-1})}\right] \\ &\times (F_{n} - F_{n-1})u(t - T_{n-1}) \\ &- \left[\frac{K_{\rm f}-1}{K_{\rm p}} + (t - T_{n}) - \frac{K_{\rm f}-1}{K_{\rm p}} \,{\rm e}^{-K_{\rm p}(t - T_{n})}\right] \\ &\times F_{n}u(t - T_{n}) \end{aligned}$$
(6)

Checking the steady state error

$$E(s) = R(s) - Y(s) = R(s)[1 - G(s)]$$
  
=  $R(s)\frac{(1 - K_{\rm f})s}{s + K_{\rm p}} = \frac{(1 - K_{\rm f})V(s)}{s + K_{\rm p}}$  (7)

B02201 © IMechE 2002

the final value theorem gives the value

$$\lim_{s \to 0} \left[ s \frac{1 - K_{\rm f}}{s + K_{\rm p}} V(s) \right] = \frac{1 - K_{\rm f}}{K_{\rm p}} \lim_{s \to 0} \left[ s V(s) \right] \tag{8}$$

If  $\lim_{t\to\infty} [v(t)] = \lim_{s\to0} [sV(s)]$  exists, say  $V_s$ , then  $\lim_{t\to\infty} [e(t)] = [(1 - K_f)/K_p]V_s$ . Basically, using 1 for the feedforward gain should be able to reduce the steady state error. However, there is always some modelling error in the real-world application. The noise problem still exists. Also, large  $K_p$  gain might excite unmodelled high-frequency resonance. Therefore, fine tuning the position and velocity gains is still necessary.

#### 2.2 Dynamic response matching

It is necessary in the CNC system to adjust the dynamics of some axes. An obvious application when this is necessary is in the rigid tapping process. There are two commonly used methods in the control of rigid tapping. The first is to control the z axis so that it will follow the spindle motion. The second method is to match the z axis and the spindle dynamics so that the interpolation process can treat the spindle as an additional feed axis. Basically, the axis impedances should be tuned so that they are identical. Although adjusting the servo feedback gains can achieve a similar effect, changing these gains might also affect axis overshoot and disturbance rejection. Common practices do not sacrifice axis performances for impedance purpose. Instead, dynamic response matching (second block in Fig. 1) is used to synchronize different axes and to reduce the following errors. As for the actual setting, all the axes are matched against the slowest axis. For example, if the time constant for the base axis is 2.5 ms, then the time constant parameters for all the axes are set to 0.0025 s.

#### 2.3 Friction compensation

CNC machine tools are always heavy in weight. This makes friction an important issue in the machine tool dynamics. 'Circular inspection' is a standard way to present the effect of friction. When the machine moves in a circle, there are position errors due to friction change at the boundary of each quadrant. These errors are commonly known as 'spikes'. When one axis switches its moving direction, it encounters a change in the friction and results in a delay in the motion of that axis. Typically, this problem is compensated in the current loop. An additional current (proportional to the torque) is added at the moment when an axis starts from zero speed and this current offset is shut down after a period of time. Consequently there are two parameters for this function: 'the friction compensation amplitude' and 'the friction compensation time constant'.

## 3 COMPOUND GENETIC ALGORITHM METHOD TO OPTIMIZE MULTI-PLANE PERFORMANCE OF MACHINE CONTROL

The previous section described several functions involved in machine tool control. These functions all carried parameters that have to be skilfully tuned. Basically, an experienced engineer might be able to tune these parameters for single-axis motions. It is almost impossible for anyone to tune the controller for simultaneous multi-axis motion. Note that good singleaxis performances do not guarantee good overall multiaxis performance. To help to solve this problem, the present authors have developed a 'compound genetic algorithm' (CGA) that could tune the parameters for simultaneous multi-axis motion to meet the contouring criteria.

For the proposed algorithm, first make a list of the control parameters that should be considered. Five parameters were chosen in this case (Table 1). Each parameter represents a 'gene' in the GA method, and Table 2 provides definitions for the variable names.

Choose one of the work planes on the five-axis machine for a start. The XY plane is selected in this case so that motion along the x axis and y axis are to be considered. From Table 1, each axis has five parameters; thus there are, in total, ten parameters defined as the genes (Table 2) [22, 23].

With the genes defined, the procedure is as follows:

1. Choose a mating pool and decide the number of parent chromosomes and child chromosomes. Each chromosome carries ten genes.

|--|

Number	Function	Mapped parameter number in the Siemens controller
1	FRICT_COMP_TIME	MD 32540
2	FRICT_COMP_CONST_MAX	MD 32520
3	VELO_FFW_WEIGHT	MD 32610
4	DYN_MATCH_TIME	MD 32910
5	POSCTRL_GAIN	MD 32200

Proc Instn Mech Engrs Vol 216 Part B: J Engineering Manufacture

#### Table 2Physical meanings of the genes

- 1 X axis friction compensation time constant
- 2 Y axis friction compensation time constant
- 3 *X* axis friction compensation constant maximum value 4 *X* axis friction compensation constant maximum value
- 4 Y axis friction compensation constant maximum value
   5 X axis velocity feedforward weight
- 6 Y axis velocity feedforward weight
- 7 X axis dynamic match time constant (units, ms)
- 8 Y axis dynamic match time constant (units, ms)
- 9 X axis position control loop P gain
- 10 Y axis position control loop P gain
- 2. Randomly initialize the values of the genes of the parent chromosomes.
- 3. Produce the next generation by mating the parent chromosomes. The evolution contains crossover and mutation.
- 4. Design a fitness algorithm.
- 5. Evaluate the fitness of the parent and child chromosomes. According to the result, some of the children will replace parents, and this forms a new generation.
- 6. Repeat evolution until the fitness of all the parent chromosomes has converged.

Note that a circular test in CNC could identify the tracking errors caused by servo gains, errors caused by friction, errors caused by mismatch of dynamics between axes and errors due to the lack of feedforward control. Because of the rich information presented by circular motion (G02 or G03 in CNC), tracking errors of circular movement are taken as the fitness value. Note that the fitness function evaluates the difference between the motion command trajectory and the actual trajectory but not the time index when the position is reached. The fitness function is defined as

$$F = \frac{1}{N} \sum \text{abs} \left[ \sqrt{(X_i - X_c)^2 + (Y_i - Y_c)^2} - R \right]$$
(9)

where

F = fitness (the smaller the better)

- N = number of data acquired
- $X_i = X$ -position data acquired in circling
- $Y_i = Y$ -position data acquired in circling
- $X_{\rm c} = X$ -position datum of the centre of the standard circle (command circle)
- $Y_{\rm c} = Y$ -position datum of the centre of the standard circle (command circle)
- R = radius of the standard circle

The crossover process in the evolution follows the standard uniform crossover method and roulette circle selection is adopted for the selection method. A mutation process contains two factors: mutation rate and mutation amplitude. In this research, a 'dynamic mutation amplitude' or 'mutation rate' is introduced. For example, if the mutation amplitude starts at a value of 0.8 and is reduced to 0.4 (50 per cent) after some generations of

evolution, and 70 per cent of chromosomes fall inside the convergence range, it may be necessary to slow down the evolution process. Slowing down the evolution can refine the step size of the update. Some experiments in the next section will show that dynamic mutation provides a better evolution quality than fixed mutation.

Sometimes numerical values of the genes are encoded into binary values to gain a better convergence rate. Even in the crossover process, binary coded genes contained more diversity in the sense of 'cutting' and 'reassembling'; however, the increased computation overhead due to the coding process will lead to difficulty in real-time implementation. Therefore, numerical values are still used in the computation.

The discussions in the previous paragraphs are based on motion in a single XY plane. After successfully constructing a method for automatic parameter optimization, next multi-dimensional motion is considered. It is possible to apply the same XY plane method to the YZand ZX planes, but then it is hard to select the proper final value because each axis has participated more than once in the evaluation process. Therefore a new method to optimize the multi-axis motion performance of the machine tool is necessary.

As mentioned earlier, the in-plane circular test gives rich information on the quality of the servo performance of the associated axes. Thus, to reach a better performance in multi-dimensional motion control, an evaluation that includes circular tests in all the three Cartesian planes sounds reasonable. In order to place emphasis on the friction compensation, the points measured near the intersections of the four quadrants are assigned heavier weights. The centre of the command circle is set at the origin for the experiments;  $X_c$ ,  $Y_c$  and  $Z_c$  are set to zero; therefore the fitness algorithm can be expressed as

$$F = \frac{A+B+C}{D} \tag{10}$$

with

$$A = \sum_{i=1}^{A_N} \left( \sqrt{X_i^2 + Y_i^2} - R \right) + \sum_{j=1}^{A_M} w_{A_j} \left( \sqrt{X_j^2 + Y_j^2} - R \right)$$
(11)

$$B = \sum_{i=1}^{B_N} \left( \sqrt{Y_i^2 + Z_i^2} - R \right) + \sum_{j=1}^{B_M} w_{B_j} \left( \sqrt{Y_j^2 + Z_j^2} - R \right)$$
(12)

$$C = \sum_{i=1}^{C_N} \left( \sqrt{Z_i^2 + X_i^2} - R \right) + \sum_{j=1}^{C_M} w_{C_j} \left( \sqrt{Z_j^2 + X_j^2} - R \right)$$
(13)

$$D = \sum_{i=1}^{A_N} + \sum_{i=1}^{B_N} + \sum_{i=1}^{C_N} + \sum_{j=1}^{A_M} w_{A_j} + \sum_{j=1}^{B_M} w_{B_j} + \sum_{j=1}^{C_M} w_{C_j}$$
(14)

where

- F = fitness value (the smaller the better)
- $A_N$  = number of regular points in the XY plane data
- $B_N$  = number of regular points in the YZ plane data
- $C_N$  = number of regular points in the ZX plane data
- $A_M$  = number of near-quadrant points in the XY plane data
- $B_M$  = number of near-quadrant points in the YZ plane data
- $C_M$  = number of near-quadrant points in the ZX plane data
- $w_A$  = weight for the near-quadrant XY plane error
- $w_B$  = weight for the near-quadrant YZ plane error
- $w_C$  = weight for the near-quadrant ZX plane error
- $X_i = X$  position data
- $Y_i = Y$  position data
- $Z_i = Z$  position data
- $X_i$  = near-quadrant points for the X axis
- $Y_i$  = near-quadrant points for the Y axis
- $\vec{Z_i}$  = near-quadrant points for the Z axis
- $X_c = X$  position of the centre of the standard circle (command circle)
- $Y_{\rm c} = Y$  position of the centre of the standard circle (command circle)
- $Z_c = Z$  position of the centre of the standard circle (command circle)
- R = radius of the standard circle

This is the fitness function for the three-dimensional motion. Because there are three axes involved in the evolution, the new chromosome should also contain the parameters for the three-axis controller (Table 3).

The number of genes in the new chromosome increases to 15 and the complexity increases accordingly. Suppose that 8 bits are used to represent a parameter value; then the ten genes for an in-plane motion analysis will contain  $8^{10}$  possible variations. The 15 genes for the threedimensional motion analyses will be 24000 times more complex. It is necessary to have an improved evolution algorithm. The experimental results in the next section will also demonstrate this fact.

 Table 3
 The genes on a chromosome in the three-axis case

1 X axis friction compensation time constant (un Y axis friction compensation time constant (un	(units, ms) (units, ms)
2 Vavis friction compensation time constant (un	(units, ms)
2 I axis metion compensation time constant (un	
3 Z axis friction compensation time constant (un	(units, ms)
4 X axis friction compensation constant maximu	num value
5 Y axis friction compensation constant maximu	num value
6 Z axis friction compensation constant maximu	num value
7 X axis velocity feed forward weight	
8 Y axis velocity feed forward weight	
9 Z axis velocity feed forward weight	
10 X axis dynamic match time constant (units, ms	ms)
11 Y axis dynamic match time constant (units, ms	ms)
12 Z axis dynamic match time constant (units, ms	ms)
13 X axis position control loop P gain	
14 Y axis position control loop P gain	
15 $Z$ axis position control loop $P$ gain	

The modification made in this study is a systematic approach to improve the 'quality' of the initial genes. The single-plane analysis used randomly initialized genes, but the convergence rate in the case with 15 genes becomes so slow that it becomes useless. At this point, attempts should be made to benefit from the existing results. Use is made of the results from the single-plane GA and the average used as a reasonable initial guess for the multi-dimensional GA evolution. This is called the CGA method, as already mentioned. The procedures are as follows:

- 1. Follow the method introduced before to apply GA on the *XY* plane and to obtain a set of parameters for the *X* and *Y* axes. Use an easy convergence bound value as it is not the final result of the desired parameters.
- 2. Repeat step 1 for the Y and Z axes.
- 3. Repeat step 1 for the Z and X axes.
- 4. Average the results of steps 1, 2 and 3 to obtain a set of good initial parameter values.
- 5. Determine  $A_N$ ,  $B_N$ ,  $C_N$ ,  $A_M$ ,  $B_M$ ,  $C_M$ ,  $w_A$ ,  $w_B$  and  $w_C$  in the fitness algorithm.
- 6. Apply GA evolution. Note that dynamic mutation is adopted in the evolution.

In the analysis of the efficiency of the method for this application, the total time of evolution of a non-CGA method is approximately  $T = 3(2\pi R/F)N + T_cN$ , where T is the total time for evolution,  $2\pi R$  is the length of circular path, F is the feed rate (vector speed) of the machine tool, N is the number of generations taken to complete the evolution and  $T_c$  is the computation time for each generation. The time required for the CGA method, on the other hand, can

be represented as

$$T = \left(\frac{2\pi R}{F}\right) N_{xy} + \left(\frac{2\pi R}{F}\right) N_{yz} + \left(\frac{2\pi R}{F}\right) N_{zx} + 3\left(\frac{2\pi R}{F}\right) N_{xyz} + T_c N = \left(\frac{2\pi R}{F}\right) (N_{xy} + N_{yz} + N_{zx} + N_{xyz}) + T_c N$$
(15)

where  $N_{xy}$  denotes the number of generations for initial values derived on the XY plane,  $N_{yz}$  denotes the number of generations for initial values derived on the YZ plane,  $N_{zx}$  denotes the number of generations for initial values derived on the ZX plane and  $N_{xyz}$  denotes the number of generations for XYZ three-axis evolutions.

Comparing the CGA and non-CGA methods, if  $N_{xy} + N_{yz} + N_{zx} + N_{xyz} < 3N_{\text{non-compound}}$ , then the computation time of the CGA method will be less than that of the non-CGA method. In other words, the compound method is more efficient.

The experimental result in the next section will also show that the CGA is able to converge even with a large number of genes, and the CGA method is more efficient.

#### **4 EXPERIMENTAL RESULTS**

All the experiments were carried out on a five-axis machining centre, which is controlled by a Siemens 840D open-architecture computer numerical controller. An in-house autotuning program was developed based



Fig. 3 The human-machine interface of the GA program

on the GA. This program communicates with the 840D controller through the Siemens MPI bus. The architecture enables automation of the GA tuning process. The human-machine interface of the software is shown in Fig. 3.

The experiments were carried out in two parts. Experiment A showed the result of three-dimensional tuning without the CGA method. In this experiment parameters were directly tuned in their corresponding planes. To evaluate the fitness, a cycle of three circular machine movements (one circle on each Cartesian plane) was repeated. Experiment B showed the result of threedimensional tuning with the CGA method. It can be seen that the CGA method solved the problem in experiment A.

#### 4.1 Experiment A

As a comparison, the GA method was directly applied to the tuning on three single planes. The number of genes in a chromosome was 15. The experimental conditions are given in Table 4.

The convergence criterion was set at a tight value of  $10\,\mu$ m. After 60 generations of evolution, the fitness of the sixtieth parent chromosomes was between 24.487 and 26.173  $\mu$ m, and it had already taken 1.5 h (Figs 4 to 6). Figure 4 shows the resultant trajectory in the XY plane. Figures 5 and 6 show the resultant trajectories in the XZ and YZ planes respectively. Note that in these

Table 4         Experimental conditions for expendence	eriment A
--	-----------

Feed rate	5000 mm/min
Circle radius	$20\sqrt{2}\mathrm{mm} = 28.284\mathrm{mm}$
Centre positions	
X	-400
Y	-200
Ζ	-100
Number of parent chromosomes	45
Number of child chromosomes	30
Number of genes	15
Selection method in crossover	roulette circle method
Crossover method	uniform crossover
Mutation rate	0.3
Mutation amplitude	0.8 originally, reduced to 0.4 after 70 per cent of chromosomes are in convergence range
Convergence range	10 μm

figures the error is multiplied ten times for visual convenience. The results look unsatisfactory. The convergence was very slow and there was no way to tell whether the GA would converge at all. An improved method was necessary.

#### 4.2 Experiment B

In this experiment, the CGA was applied with the same experimental conditions, except that a tighter convergence level of  $7 \mu m$  was given. As the previous example showed, it was difficult to evolve in the 'most direct' sense or 'correct direction' when the gene parameters



**Fig. 4** The XY plane circular motion trajectory: experiment A

		Zoom X1
		Zoom
Genetic Tuning Graphic Display (µm)		X10
Project: r		
0919xy 💌		Zoom X50
00xy09.cdt		
00xy10.cdt		Zoom
00xy11.cdt		X100
00xy12.cdt		
60xz01.cdt		Print
60vz01 cdt		
Center Pos: -420000 -120000		
Mex Velue: 201071 01700		
Min Velue: 440200 140174		
Mill Value448266 -148174		
AVG Error: 22.1404	I	
Error Multiply Factor: 10		
	EXIT	
Auto Manual Graph Ger Tuning Tuning Display List	nes t	

Fig. 5 The XZ plane circular motion trajectory: experiment A



Fig. 6 The YZ plane circular motion trajectory: experiment A

were initialized with random values. Therefore, the CGA is followed. First, obtain single-plane GA results quickly on the XY, YZ and ZX planes respectively. The XY plane experiment provided good values for the X axis and the Y axis gene parameters. Similar situations were also true for the YZ and ZX planes. Then, there were in total two sets of X, Y and Z axis gene values. For the X axis the average of the two X axis values could be used as the initial value for the multi-axis GA tuning. Again, similar procedures were carried out for the Y and Zaxes. Based on these good initialization values the system could now evolve to meet the multi-axis performance criteria more easily. The final result was that for the XY single plane it took 38 generations to reach an average fitness of  $5.19\,\mu m$ . On the YZ plane it took 22 generations to reach an average fitness of 6.17 µm. On the ZX plane it took 40 generations to reach an average fitness of 5.44 µm. In the second step of the CGA method, which was three-plane tuning, it converges as fast as six generations to an average fitness of  $6.2 \,\mu\text{m}$ . The value has met the requirement of the convergence value. To see how good this method is, the tuning process was continued. Figure 7 shows the resultant trajectory in the XY plane. The trajectories in the XZ and YZ planes are similar and the figures are not shown. After another 32 generations it was found that the fitness is  $3.1 \,\mu\text{m}$  on the XY plane,  $3.4 \,\mu\text{m}$  on the XZ plane and  $1.5 \,\mu\text{m}$  on the YZ plane. Note that in these figures the error is multiplied 100 times for visual convenience.

It is clear that the difficulty in convergence was solved. Although it took more time to 'prepare' the initial values, the total evolution was still much better not only in the evolution time required but also in the resulting performance.

## 5 CONCLUSIONS

In this paper, a method to optimize motion control parameters automatically in the machine tool computer numerical controller was introduced. A dynamic mutation method was developed and proved to be effective in the sense of convergence quality. Furthermore, an improved GA, the CGA, was presented. The proposed method resolved the difficulty of the large numbers of genes in the multi-axis tuning process and supplied a systematic architecture to optimize the multi-axis CNC performance tuning, where previously there was no efficient and unambiguous way available. The experimental results show that the proposed algorithm could reach convergence to within  $3.4 \,\mu$ m, which is very difficult even for an experienced engineer.



Fig. 7 The XY plane circular motion trajectory: experiment B

#### 438

## ACKNOWLEDGEMENTS

This project is supported in part by the Industrial Technology Research Institute, Taiwan, under Project 893K51AQ2, which is a subcontract from the Ministry of Economic Affairs, Taiwan, and in part by the National Science Council under Project NSC 89-TPC-7-002-008. Also the authors are grateful for the support of Feeler Machine Tool Company in providing the experimental set-up.

## REFERENCES

- 1 Younkin, G. Modeling machine tool feed servo drives using simulation techniques to predict performance. In Conference Record of the Industry Applications Society Annual Meeting, 1989, Vol. 2, pp. 1699–1706
- 2 Alter, D. M. and Tsao, T.-C. Dynamic stiffness enhancement of direct linear motor feed drives for machining. In Proceedings of the 1994 American Control Conference, 1994, Vol. 3, pp. 3303–3307
- 3 Choi, B. K., Choi, C. H. and Lim, H. Model-based disturbance attenuation for CNC machining centers in cutting process. *IEEE-ASME Trans. Mechatronics*, June 1999, 4(2), 157–168.
- 4 Cuttino, J. F., Miller, Jr, A. C. and Schinstock, D. E. Performance optimization of a fast tool servo for single-point diamond turning machines, *IEEE–ASME Trans. Mechatronics*, June 1999, 4(2), 169–179.
- **5** Arakawa, A. and Miyata, K. Simultaneous optimization algorithm for determining both mechanical-system and controller parameters for positioning mechanisms. In Proceedings of the 1996 4th International Workshop on *Advanced Motion Control (AMC'96)*, Tsu, Japan, 18–21 March 1996, 1996, Vol. 2, pp. 625–630
- **6** Xiao, B.-X. The main control mode and fuzzy control strategy of CNC system for gear hobbing and grinding machine. In Proceedings of the IEEE International Conference on *Industrial Technology*, 1996, pp. 643–646 (IEEE, New York).
- 7 Liu, J., Yamazaki, K. and Yokoyama, Y. Dynamic gain motion control with multi-axis trajectory monitoring for machine tool systems. In Proceedings of the 1998 International Workshop on *Advanced Motion Control* (*AMC'98*), 1998, pp. 316–321.
- 8 Liu, J., Yamazaki, K. and Yokoyama, Y. Dynamic gain motion control with multi-axis trajectory monitoring for machine tool systems. In Proceedings of the 1998 International Workshop on *Advanced Motion Control* (*AMC'98*), 1998, pp. 316–321.
- 9 Fornaro, R. J. and Dow, T. A. A high-performance machine tool controller. In Conference Record of the 1988 IEEE

Industry Applications Society Annual Meeting, 1988, Vol. 2, pp. 1429–1439 (IEEE, New York).

- 10 Wang, P. and Kwok, D. P. Optimal design of PID process controllers based on genetic algorithms. *Control Engng Practice*, August 1994, **2**(4), 641–648.
- 11 Wu, C.-J. and Huang, C.-H. A hybrid method for parameter tuning of PID controllers. J. Franklin Inst., July 1997, 334(4), 547–562.
- 12 Lin, S.-C. and Chen, Y.-Y. Design of self-learning fuzzy sliding mode controllers based on genetic algorithms. *Fuzzy Sets Systems*, 1 March 1997, 86(2), 139–153.
- 13 Ishigami, H., Hasegawa, Y., Fukuda, T. and Shibata, T. Automatic generation of hierarchical structure of fuzzy inference by genetic algorithm. In Proceedings of the 1994 IEEE International Conference on *Neural Networks*, Orlando, Florida, 27–29 June 1994, Vol. 3, pp. 1566–1570 (IEEE, New York).
- 14 Wang, L. and Yen, J. Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter. *Fuzzy Sets Systems*, 1 February 1999, 101(3), 353–362.
- 15 Tesar, A. and Drzik, M. Genetic algorithms for dynamic tuning of structures. *Computers Structs*, 17 October 1995, 57(2), 287–295.
- 16 Ulyanov, S. V., Yamafuji, K., Miyagawa, K., Tanaka, T. and Fukuda, T. Intelligent fuzzy motion control of mobile robot for service use. In Proceedings of the 1995 IEEE–RSJ International Conference on *Intelligent Robots and Systems*, Pittsburgh, Pennsylvania, 5–9 August 1995, Vol. 3, pp. 486–489 (IEEE, New York).
- 17 McGookin, E. W., Murray-Smith, D. J., Li, Y. and Fossen, T. I. Ship steering control system optimisation using genetic algorithms. *Control Engng Practice*, April 2000, 8(4), 429–443.
- 18 Wu, T. S. and Liu, J. C. Fuzzy control of rider-motorcycle system using genetic algorithm and auto-tuning. *Mechatronics*, June 1995, 5(4), 441–455.
- 19 Tarng, Y. S., Chuang, H. Y. and Hsu, W. T. Intelligent cross-coupled fuzzy feedrate controller design for CNC machine tools based on genetic algorithms. *Int. J. Mach. Tools Mf.*, October 1999, **39**(10), 1673–1692.
- **20** Kuo, L.-Y. and Yen, J.-Y. Servo parameter tuning for a 5axis machine center based upon GA rules. *Int. J. Mach. Tools Mf.*, 2001, **41**(11), 1535–1550.
- 21 Kuo, L.-Y. Real-time and multi-tasking 5-axis controller design with open architecture. Project Report 893k51AQ0, The Minister of Economic Affairs, Taiwan, executed by Mechanical Engineering Research Laboratory, Industrial Technology Research Center, Taiwan, 1999– 2000.
- **22** Mitsuo, G. Genetic Algorithms and Engineering Design, 1997 (John Wiley, New York).
- 23 Sanchez, E., Takanori, S. and Lotfi, A. Z. 1997, Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives, 1997 (World Scientific, Singapore).