Available online at www.sciencedirect.com

ScienceDirect

Computers & Industrial Engineering 52 (2007) 355–379

computers & industrial engineering

www.elsevier.com/locate/dsw

System gains configuration and coordination of redundant degrees of freedom by genetic algorithms for multi-axis machine system in manufacturing

Jih-Hua Chin *, Yu-Cheng Chen

Department of Mechanical Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, ROC

Received 21 April 2006; received in revised form 16 November 2006; accepted 3 January 2007
Available online 18 January 2007

Abstract

In multi-axis machine tool systems, the configuration of system gains and the coordination of redundant degrees of freedom are often a problem of insurmountable difficulty. This study explores the use of a nontraditional scheme, the genetic algorithm, in the configuration of system gains and exploitation of redundant degrees of freedom. The off-line gains configuration functioned as a kind of system design which may serve as a starting point for on-line adaptation. The allocation of redundant DOF was done on-line. The success in this work inspired the idea of future on-line GA application and the possibility of integration of GA with other non-traditional algorithms for manufacturing.

Keywords: Machine tools; Cross-coupling control; Genetic algorithms; Tracking; Motion coordination

1. Introduction

The improvement of modern manufacturing system is due to advances in many aspects. The advances toward high speed and high performance were often confronted with insufficient structural stiffness of traditional serial structure. This promotes the advent of machine tools of parallel structure. Parallel structure machine tools are capable of machining sophisticated curved surface. They are simple in construction but complicated in mathematics. Besides, they prevail only in a limited workspace which is still threatened by singular points.

It is thus conceivably advantageous to integrate the features of stiffness and speed of parallel structure into a proven traditional machine tool to form a new breed of multi-axis machine tool with spacious workspace and ability of continuous path tracking. A hybrid multi-axis machine tool as shown in Fig. 1 was built in this work to investigate that idea (Fig. 2).

* Corresponding author. Tel.: +886 3 5731965; fax: +886 3 5727485.
E-mail address: jhchin@mail.nctu.edu.tw (J.-H. Chin).

0360-8352/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2007.01.002

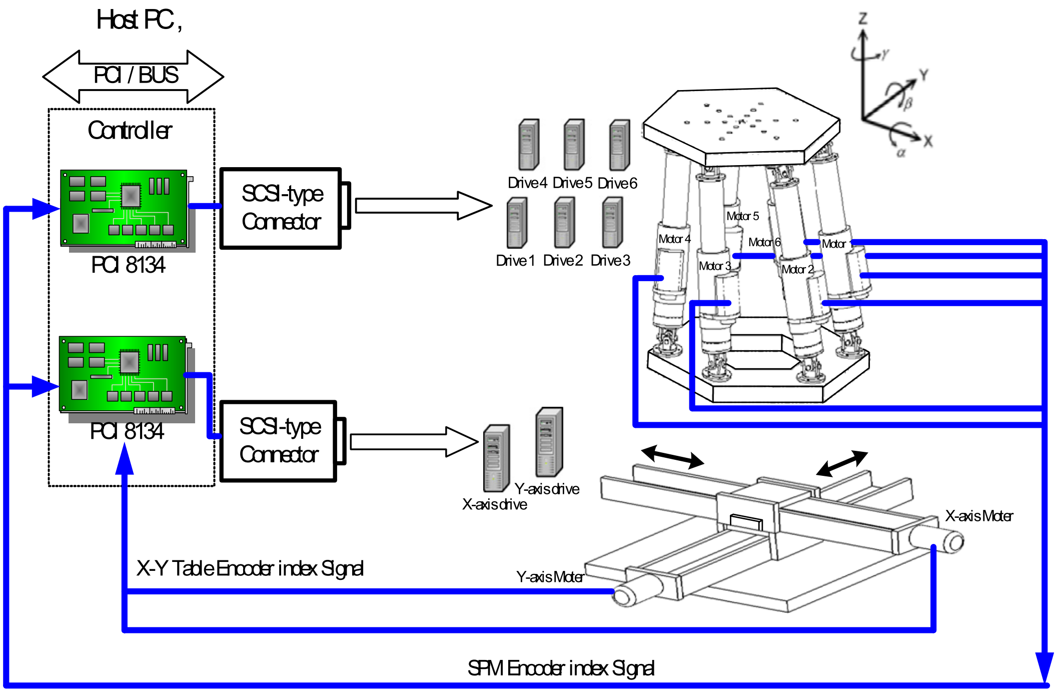


Fig. 1. A multi-axis machine tool capable of cross-coupled tracking control is composed of a *x-y* base and a parallel upper structure.

The concept of system structure was first introduced in (Lue, Cheng, & Chin, 2005) and it was shown that a cross-coupled pre-compensation tracking control was possible for such a multi-axis machine tool. However, problem arises in the system configuration involving gains determination and motion coordination between traditional machine tool base and the parallel upper structure. This is because there are too many gains, eight driving axis each requires a regular tracking gain, a speed pre-compensation gain, and a contour compensation gain. The number of gains prohibited an efficient trial and error or even any traditional systematic approach. Besides, since there are redundant degrees of freedom, a machining point can be approached by different machine movements, hence motion coordination or allocation emerges as an object (Fig. 3).

Past experiences showed that emerging new technologies often offer chances to solve old problems in a relatively easy way. For example, high contour precision was obtained by mathematically hard-won path generator (Chin & Lin, 1999), but a simple Fuzzy logic controller could compete with that rigorous path generator (Chin, Cheng, & Lin, 2003).

This work addressed problem of system gains configuration for a multi-axis machine tool implemented with sophisticated tracking controls which involved many gains. Besides, the problem of motion coordination for a system with redundant degrees of freedom was also addressed. Genetic algorithm was chosen to deal with the problems involved.

A quick review of literature shall help understanding the complexity in advanced tracking control. Historically position control is the basic performance of machine tools, which often suffices for straight line machining if implemented with well configured gains. But for profile machining, higher contour fidelity set a need for contour algorithms. As early as 1957, Sarachik and Ragazzini (1957) proposed the idea of a cross-coupled system. But modern cross-coupled control was initiated by Koren's work (1980). In cross-coupled system, there are cross-coupled gains to be determined. Later Koren and Lo (1991) proposed variable gains. Chin and Tsai (1993) used a speed pre-compensation in tracking control which is also useful for flexible robot tracking (Chin & Lin, 1997a) but pre-compensation gains were needed. Subsequently, Chin and Lin (1997b) combined the cross-coupled control with the speed precompensation to achieve a system which outstands at higher speed and higher curvature, but the system requires both cross-coupled gains and pre-compensation gains. For

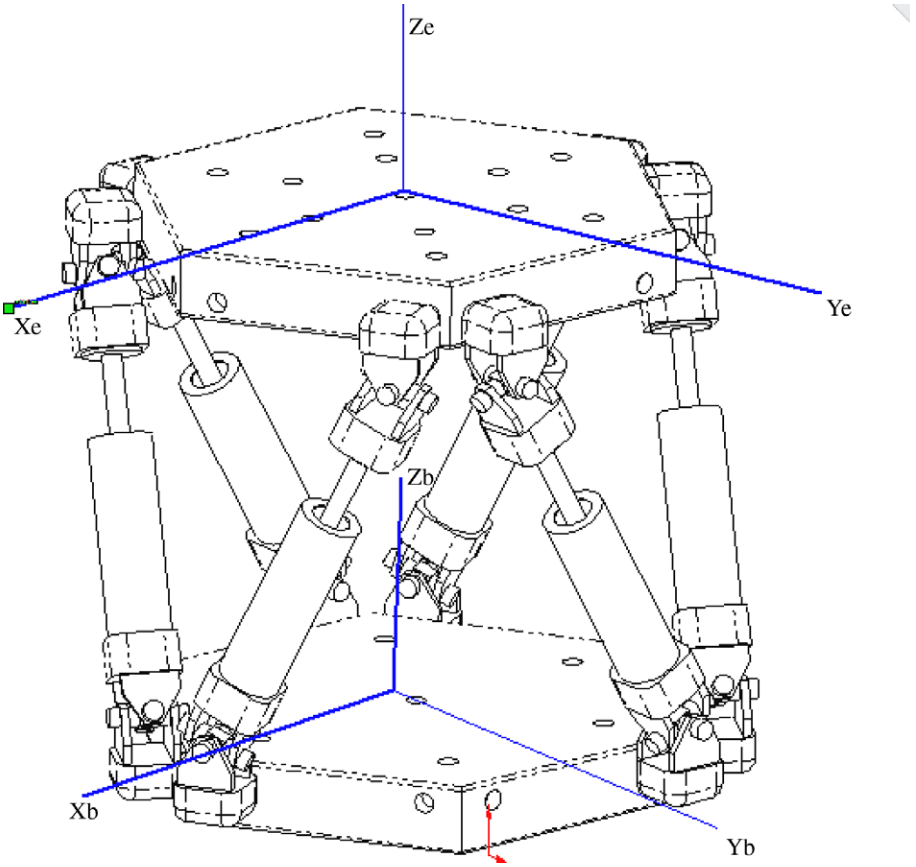


Fig. 2. Sub-system of parallel structure in multi-axis machine tool system.

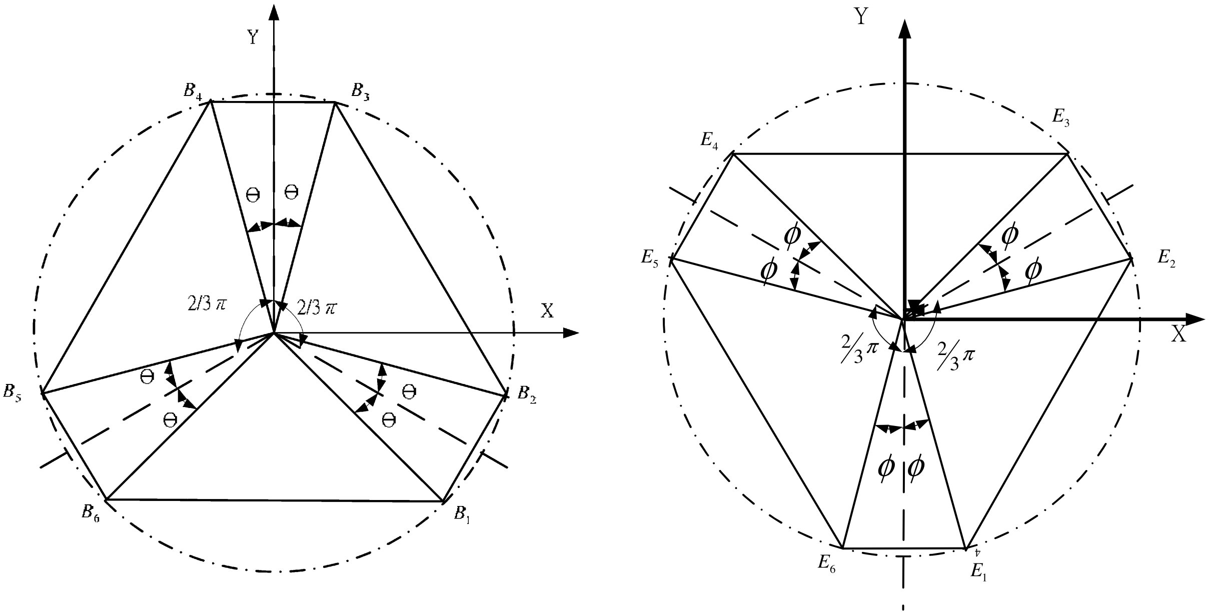


Fig. 3. Base platform (left) and upper platform (right) of parallel structure.

the system with multi-axis like the one in (Lue et al., 2005), the selection of gains is a work of insurmountable difficulty. The conceptual work in (Lue et al., 2005) used tentative gains obtained by trial-and-error to bring out simulated results. The real system was built in this work and the gains as well as motion allocation problems were addressed systematically which were then implemented and tested empirically.

Genetic algorithm (GA) (Fogel, 1988; Scott, 1990) simulated natural selection law to create an artificial survival and evolution process. The evolution process can be seen as a kind of multi-dimensional search in the problem domain (Goldberg, 1989). General reviews of the development of GA in different kinds of fields are reported in (Man, Tang, & Kwong, 1996; Srinivas & Patnaik, 1949). Compared with other optimization algorithms GA is less likely trapped in local optimization (Yamamoto & Inoue, 1995). Although GA requires longer computation time, successful implementations were done in determination of variable structure gains and switching vector (Hussain N, Al-Duwaish, & Zakariya, & Al-Hamouz, 1998; Zakariya, Al-Hamouz, Hussain, & Al-Duwaish, 1989).

Genetic algorithms have been used in many fields of knowledge, for example planning and scheduling, engineering designs, speech recognition, pattern recognition, parameter and system identification etc (Man et al., 1996). There are also limited application of GA in control, machine tools and robotics (Tarng, Chuang, & Hsu, 1999), however, due to the problem of computation time and the nature of unpredictability (Man et al., 1996), the development of GA in manufacturing and machine tools needs more works before it becomes mature.

The purpose of this study is to explore the genetic algorithms in the configuration of system gains and the determination of motion coordination/allocation for multi-axis machine tools with redundant degrees of freedom in the manufacturing.

2. System construction

The concept of system shown in Fig. 1 was disclosed in (Lue et al., 2005) and constructed in this study. For the sake of completeness, a brief presentation of the kinematics equations was given here.

2.1. Inverse kinematics

Inverse kinematics finds the lengths of the driving links $L_1, L_2, L_3, L_4, L_5, L_6$ from the given centroid position and orientation of the upper platform $(x, y, z, \alpha, \beta, \gamma)$.

The 6 joint locations of upper platform in the base coordinates can be expressed as:

$$[{}^bE_i] = [{}^bT_e] \ * \ [{}^eE_i], \quad i = 1\text{–}6 \tag{1}$$

where $[{}^eE_i] = [L_eC\phi_i, L_eS\phi_i, 0, 1]^{\mathrm{T}}, i = 1\text{–}6$

The transformation matrix is as follows:

$$ {}^bT_e = \begin{bmatrix} C\beta \cdot C\gamma & -C\beta \cdot S\gamma & -S\beta & x \\ -S\alpha \cdot S\beta \cdot C\gamma + C\alpha \cdot S\gamma & S\alpha \cdot S\beta \cdot S\gamma + C\alpha \cdot C\gamma & -S\alpha \cdot C\beta & y \\ C\alpha \cdot S\beta \cdot C\gamma + S\alpha S\gamma & -C\alpha \cdot S\beta \cdot S\gamma + S\alpha \cdot C\gamma & C\alpha \cdot C\beta & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_1 & o_1 & a_1 & p_1 \\ n_2 & o_2 & a_2 & p_2 \\ n_3 & o_3 & a_3 & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The six joint locations of base platform in the base coordinates are

$$[{}^bB_i] = [L_bC\theta_i, L_bS\theta_i, 0]^{\mathrm{T}}, \quad i = 1\text{–}6 \tag{3}$$

The leg lengths can be obtained by calculating:

$$|L_i| = |{}^bE_i - {}^bB_i|, \quad i = 1\text{–}6 \tag{4}$$

2.2. Forward kinematics

Forward kinematics finds the centroid position and orientation of the upper platform $(x, y, z, \alpha, \beta, \gamma)$ from given leg lengths $L_1, L_2, L_3, L_4, L_5, L_6$, but the solutions are not unique.

For a set of $L_{i=1\text{–}6}$, six nonlinear equation shall be solved for $(x, y, z; \alpha, \beta, \gamma)$:

$$F_i(x, y, z, \alpha, \beta, \gamma) = [({}^bE_i - {}^bB_i)_x^2 + ({}^bE_i - {}^bB_i)_y^2 + ({}^bE_i - {}^bB_i)_z^2] - L_i^2$$

$$= \sum_{j=1}^{3} ({}^eE_{x,i}n_j + {}^eE_{y,i}o_j + p_j - {}^bB_{j,i})^2 - L_i^2 = 0, \quad i = 1\text{–}6 \tag{5}$$

A numerical approach was used to solve the above equations. The Taylor series of the above equations are:

$$F_i(P_{(1)}, L_{(1)}) = F_i(P_0, L_{(1)}) + \sum \left(\frac{\partial F_i}{\partial P} |_{P_0} \right) \Delta P = 0, \quad i = 1\text{–}6 \tag{6}$$

where the partial derivatives are

$$\sum \left(\frac{\partial F_i}{\partial P} |_{P_0} \right) \Delta P = F_i(P_{(1)}, L_{(1)}) - F_i(P_{(0)}, L_{(1)}), \quad i = 1\text{–}6$$

Eq. (6) is useful in determining the errors ΔP, and the new position/orientation.

$$J\Delta P = \Delta F \tag{7}$$

3. Genetic algorithms for gains configuration and motion coordination

3.1. Basics of genetic algorithms

The basic idea of genetic algorithms is to presume that the solution of a problem is an entity or an individual represented by a set of parameters. The entity or individual, which can be encoded as strings of values in binary form, evolves in the process of crossover, mutation, reproduction and selection until a predefined number of generations is reached or the most fitted appears.

Some basic concepts in GA used in this work are briefed as follows:

(1) *Encoding:* This is a process converting the set of variables to a string of values in binary form. In this work the variables are gains of different tracking levels.
(2) *Population size:* This is the number of strings in the population.
(3) *Decoding:* This is a process converting the string values to values of physical significance. If a parameter P with a string length of L is to be converted to a decimal value b, then the range of its decimal value is $[0, 2^L - 1]$. Let the corresponding search range be $[P_{\min}, P_{\max}]$, then the string can be linearly assigned a physical value according the following formula

$$P = P_{\min} + \frac{P_{\max} - P_{\min}}{2^L - 1} \times b$$

(4) Fitness: This is the criterion of selection. In the problem involved, tracking precision index IAE is used as the fitness criterion in system gains configuration and the least movement of the parallel sub-system is used as the fitness criterion in the motion allocation.
(5) Fitness scaling: This is a measure to prevent premature convergence, which occurs when the fitness values of individuals become highly biased. Fitness scaling scales the fitness into a reasonable range. After scaling, the maximum fitness $f_{\max}$ can be several times of the average fitness f'_{avg} which can be set equal to the un-scaled average fitness f_{avg}, see Fig. 4. An example of fitness scaling is as follows:

$$f' = a \cdot f + b$$

$$f'_{\max} = C \times f'_{\text{avg}}$$

$$f_{\text{avg}} = f'_{\text{avg}}$$

where a, b are coefficients to be determined, C is the scaling factor, f: fitness before scaling, f_{avg}: average fitness before scaling, f': average fitness after scaling, $f'_{\max}$: the maximum fitness after scaling

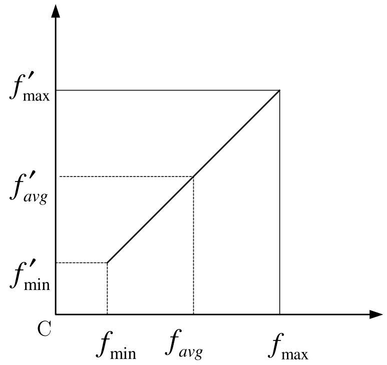


Fig. 4. Example of fitness scaling.

(6) Reproduction: A popular reproduction mechanism is the roulette wheel selection scheme. The wheel is divided into sectors each having area corresponds to the ratio of fitness of a string to the overall or average fitness. The string with the highest fitness value has the highest proportion of areas thus enjoys the highest probability of reproduction.If the number in the population is n, and f_i is the fitness of the ith string, then the reproduction probability of ith string p_i is

$$p_i = \frac{f_i}{\sum_{i=1}^{n} f_i} \quad i = 1\text{–}n$$

The accumulated probability q_i is calculated by

$$q_i = \sum_{k=1}^{i} p_k \quad i = 1\text{–}n$$

Allocate q_i onto the roulette wheel and generate a random number r in the range of $[0, 1]$. If $q_{i-1} \leqslant r \leqslant q_i$, then pick up ith string into crossover pool to do crossover maneuvering.

(7) *Crossover:* There are one-point, multi-point or uniform crossover. Uniform crossover generates offspring from the parents based on a randomly generated template string or mask. If the bit in the template string or mask is 1, then the corresponding bits in the parent strings exchanged. Whether the exchange really occurs is subjected to a crossover rate p_c which is usually between 0.5 and 1.0.

(8) *Mutation:* Mutation convert randomly selected bit from 0 to 1 and vice versa. In nature the mutation rate is low but in GA the mutation rate p_m is usually between 0.001 and 0.005. Mutation functions not only as genetic trial in new direction but also as a restoring mechanism which can retrieve the lost but good genetic material.

(9) *Maximum generation:* This is a termination criterion for the generation cycles. Other criterion is also possible, for example, the appearance of the string with the required fitness.

3.2. GA procedure for system gains configuration

The structure of the cross-coupled pre-compensated controlled (CCPM) multi-axis machine tool is shown in Fig. 5. By making gains of different control loops zero, the multi-axis CCPM reduced to different tracking controls, for example, when K_V,K_ε were set to zero, the system was a conventional tracking control without cross-coupled compensation. If only K_V was set to zero, the system was a typical cross-coupled control system (CCS).

Table 1 shows the gains for different levels of tracking controls. Because there are six DOF $(x, y, z, \alpha, \beta, \gamma)$, the gains in multi-axis CCPM system contain six terms for each control level.

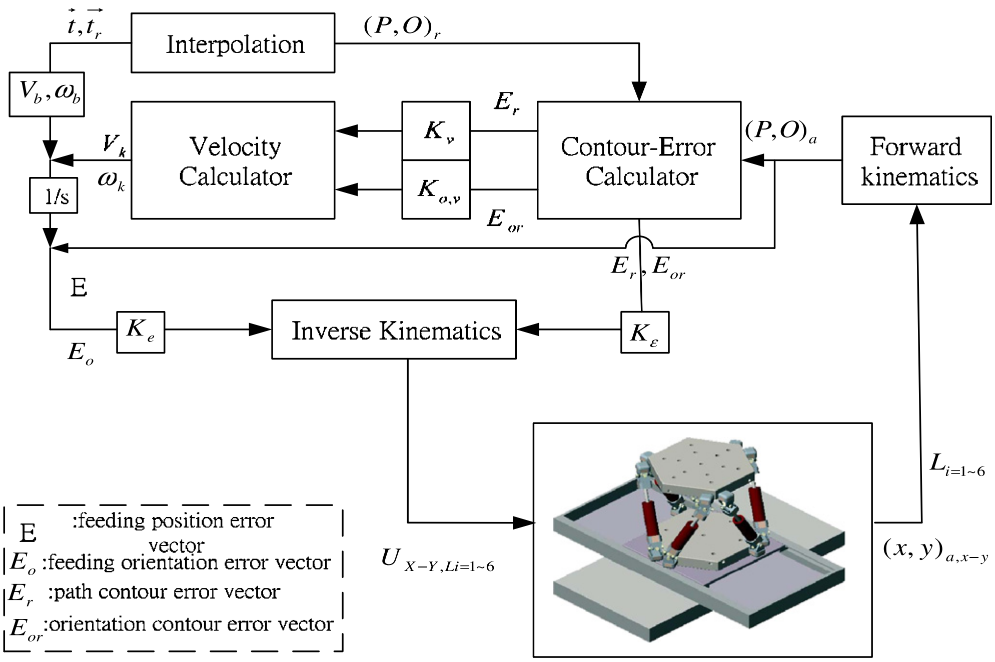


Fig. 5. Multi-axis cross-coupled pre-compensation control system [1].

Table 1
Gains of different tracking levels

Levels	Pre-compensation gains K_v	Cross-coupling gains K_ε	Conventional Tracking gains K_e
US	0	0	Non-zero
CCS	0	Non-zero	Non-zero
CCPM	Non-zero	Non-zero	Non-zero

$$K_e : K_{ex}, K_{ey}, K_{ez}, K_{e\alpha}, K_{e\beta}, K_{e\gamma}$$
$$K_\varepsilon : K_{\varepsilon x}, K_{\varepsilon y}, K_{\varepsilon z}, K_{\varepsilon\alpha}, K_{\varepsilon\beta}, K_{\varepsilon\gamma}$$
$$K_V : K_{Vx}, K_{Vy}, K_{Vz}, K_{V\alpha}, K_{V\beta}, K_{V\gamma}$$

This makes the number of gains to 18 and either analytical or trial and error process would become insurmountably tedious and difficult.

In GA configuration, K_e, K_ε, K_V become variables and the index of average absolute contour errors is chosen as the fitness function:

$$\mathrm{IAE} = \frac{1}{N}\sum_{i=1}^{n} |\varepsilon(i)|$$

The GA configuration of gains is performed loop-wise, i.e. first set K_ε, K_V zero and determine K_e for typical uncoupled tracking system. Based onK_e, next set K_V zero and determine K_ε for cross-coupled tracking system. Thirdly, with the known K_e, K_ε, determine K_V for pre-compensated cross-coupled tracking system (CCPM).

The following GA procedure for system gains configuration was created:

1. Randomly create initial population of design variables sets K_e, K_ε, K_V.
2. Using the design variables sets in the multi-axis tracking system to run a spatial trajectory. Fitness values IAE are calculated.
3. Execute selection, reproduction, crossover, and mutation to produce new generation of variables set.

4. Back to step 2 to compute the fitness values IAE for the new generation, until maximum generation is reached.
5. Pick up the best gains K_e, K_ε, K_V.

At this stage, the multi-axis machine tool performs its job by exploiting the tracking ability of XY-base. That is, all displacements in *x*-, *y*-direction are performed by X-Y base. An GA-selection for the best-fit solution of motion allocation is investigated in the next section.

3.3. GA procedure for the best-fit solution of motion coordination between redundant degrees of freedom

The parallel upper structure of the multi-axis machine tool shown in Fig. 1 has six degrees of freedom ($x_{\mathrm{PS}}, y_{\mathrm{PS}}, z, \alpha, \beta, \gamma$) while the serial X-Y base has 2 degrees of freedom (x_{XY}, y_{XY}). Displacement in x and y can be obtained by ($x_{\mathrm{PS}}, y_{\mathrm{PS}}$) from upper structure or (x_{XY}, y_{XY}) from lower base. The motion coordination or allocation becomes essential for the sake of efficiency or precision or even detouring singular points if possible. Since the X-Y base has better tracking precision from nature, it is presumably right to let X-Y base take more responsibility in tracking. This leads to the object of pursuing least motion by the parallel structure in the sense of trajectory tracking.

In the process of machining a curved surface the centroid coordinates and orientation of the upper moving plate are given as command. This is the target trajectory $(x, y, z, \alpha, \beta, \gamma)_i, i = 1 \ldots n$

The *n*th and $(n - 1)$th point on the target trajectory are
$p_n = (x,y,z,\alpha,\beta,\gamma)_n$ and $p_{n-1} = (x,y,z,\alpha,\beta,\gamma)_{n-1}$, respectively.

The displacement of the multi-axis machine tool between these two points is

$$\Delta p_n = (x, y, z, \alpha, \beta, \gamma)_n - (x, y, z, \alpha, \beta, \gamma)_{n-1} = (\Delta x, \Delta y, \Delta z, \Delta\alpha, \Delta\beta, \Delta\gamma)_n \tag{8}$$

It is reasonable to expect that better results could be obtained by displacement allocation between X-Y base and the upper parallel structure as follows:

$$\Delta p_{n.PS} = ((1 - \mu_x)\Delta x, (1 - \mu_y)\Delta y, \Delta z, \Delta\alpha, \Delta\beta, \Delta\gamma)_n \tag{9}$$

$$\Delta p_{n.XY} = (\mu_x \Delta x, \mu_y \Delta y)_n \tag{10}$$

where PS means parallel structure and μ_x,μ_y are weighting ratio, $0 \leqslant \mu_x \leqslant 1$, $0 \leqslant \mu_y \leqslant 1$ So at the point *n* the parallel structure is responsible for the following feeding

$$p_{n,PS} = p_{n-1,PS} + \Delta p_{n,PS} \tag{11}$$

while *X–Y* base is responsible for

$$p_{n,XY} = p_{n-1,XY} + \Delta p_{n,XY} \tag{12}$$

And the point *n* is the result of contributions from both parallel and serial structure:

$$p_n = p_{n,PS} + p_{n,XY} \tag{13}$$

The question now arises: how to optimize the displacement allocation through μ_x,μ_y so that an object (here the fitness function) is satisfied. In order to feed the multi-axis machine tool efficiently, the summation of square displacement of each moving axis of parallel structure is chosen as the fitness function:

$$f(\mu_x, \mu_y) = \sum_{i=1}^{6} (\Delta l_i)^2$$

The problem is stated as the following format:

$$\text{To minimize} f(\mu_x, \mu_y) = \sum_{i=1}^{6} (\Delta l_i)^2, \quad i = 1 - 6 \tag{14}$$

Subject to $0 \leqslant \mu_x \leqslant 1$, and $0 \leqslant \mu_y \leqslant 1$

The following GA procedure was created for optimum allocation between redundant degrees of freedom.

1. Assume that the displacement at ith trajectory point is $(\Delta x, \Delta y, \Delta z, \Delta\alpha, \Delta\beta, \Delta\gamma)$.
2. Create an random initial design variable pair μ_x,μ_y.
3. Calculate the corresponding fitness function $f(\mu_x, \mu_y) = \sum_{i=1}^{6}(\Delta l_i)^2$.
4. Execute selection, reproduction, crossover, and mutation to produce new generation.
5. Go back to step 3 to calculate the fitness function of the new generation. Repeat this procedure until the maximum generation is reached.
6. The optimized allocation ratio μ_x,μ_y emerged.
7. Repeat steps 1–6 for trajectory point $i + 1$, until the entire trajectory is finished.

4. Experiments and confirmations

4.1. Experimental setups and ARX model

Fig. 6 shows the multi-axis machine tool with its parallel upper structure and XY-base. The range of parallel link is 258 mm $\leqslant L_i \leqslant$ 308 mm. Radii of base and upper platform are $R_b = 150$ mm, $R_e = 120$ mm. And angles are $\theta_i = 18°$, $\phi_i = 18°$. The experimental setup is seen in Fig. 1, in which the motions of parallel and serial substructure are controlled by a personal control through two PCI-8134 four-axis motion control card. PCI-8134 send impulse commands to the driving motors every 50 ms. Displacements are monitored and fedback by six optical scales of 1 μm resolution.

The parallel upper structure has six identical driving mechanisms and the transfer function of which was identified and modeled by ARX model (Autoregressive External Input Model) as follows.

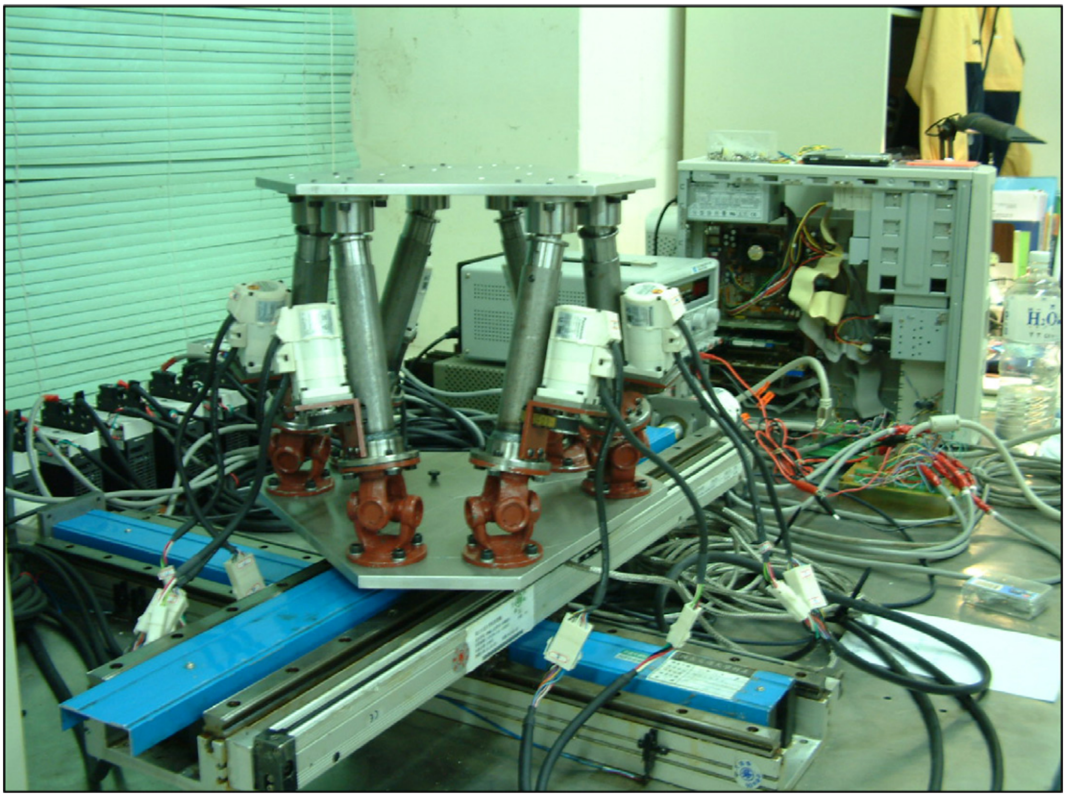


Fig. 6. Multi-axis machine tool with parallel upper structure and XY-base.

$$L_i(S) = \frac{0.1479S^2 + 3.161S + 297.8}{S^3 + 18.79S^2 + 1142S + 651.2}, \quad i = 1\text{–}6 \tag{15}$$

Fig. 7 is a comparison between the experimental and modeled movement of the parallel driving axis. It is seen the ARX model can represent the driving axis satisfactorily.

In a similar way, the transfer function of the XY-base can be identified and modeled as follows:

$$X(S) = \frac{0.237S^2 + 9.691S + 462.2}{S^3 + 12.79S^2 + 2526S + 43.27} \tag{16}$$

$$Y(S) = \frac{0.2041S^3 + 19.76S^2 + 878.7S + 18840}{S^4 + 48.05S^3 + 2865S^2 + 110900S + 9507} \tag{17}$$

Figs. 8 and 9 are the comparisons between experimental and modeled x-axis and y-axis behavior, respectively. Again the modelings are also satisfactory.

4.2. Basic structure and tool-workpiece orientation

Fig. 10 shows the GA optimized multi-axis pre-compensated cross-coupled system for the hybrid machine tool. This hybrid machine tool is conceived for servicing as a feeding system with centroid coordinates of its upper platform in the fixed coordinates system $(x, y, z, 0, 0, -1)$.

Let S be a continuous curved surface on which a continuous trajectory P_n is to be tracked. The following basic orientations describe how the hybrid multi-axis CNC machine maintains tool and workpiece relationship.

(1) Maintaining trajectory point P_i at the same height z as that of tool tip.
(2) Keeping the normal **N** to the machined surface parallel to the tool axis by movement of (α, β).
(3) Keeping the tangent **T** at the machining point collinear with the feeding direction by movement of γ.

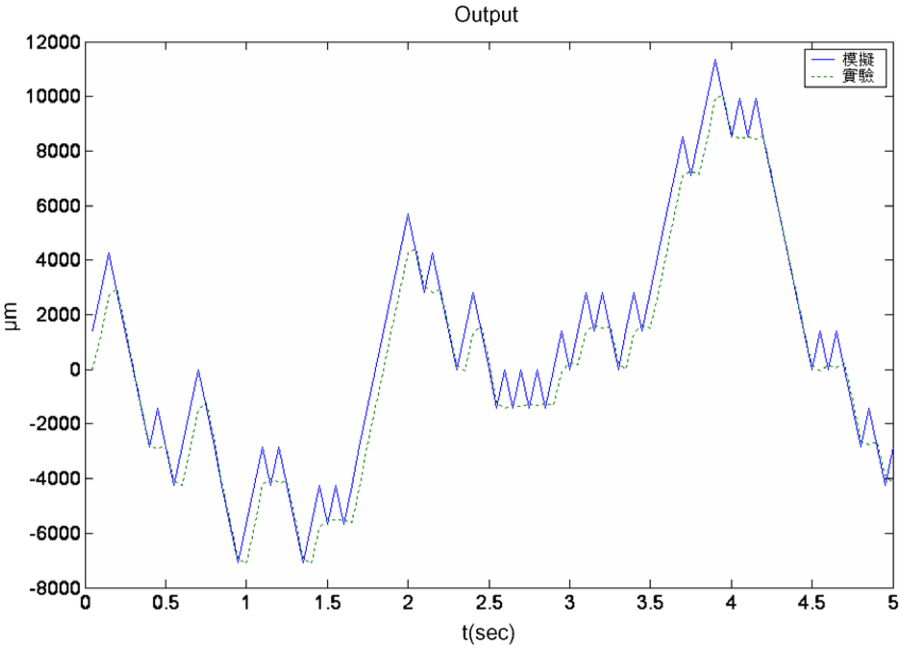


Fig. 7. Comparison between experimental and simulated displacement of parallel link.

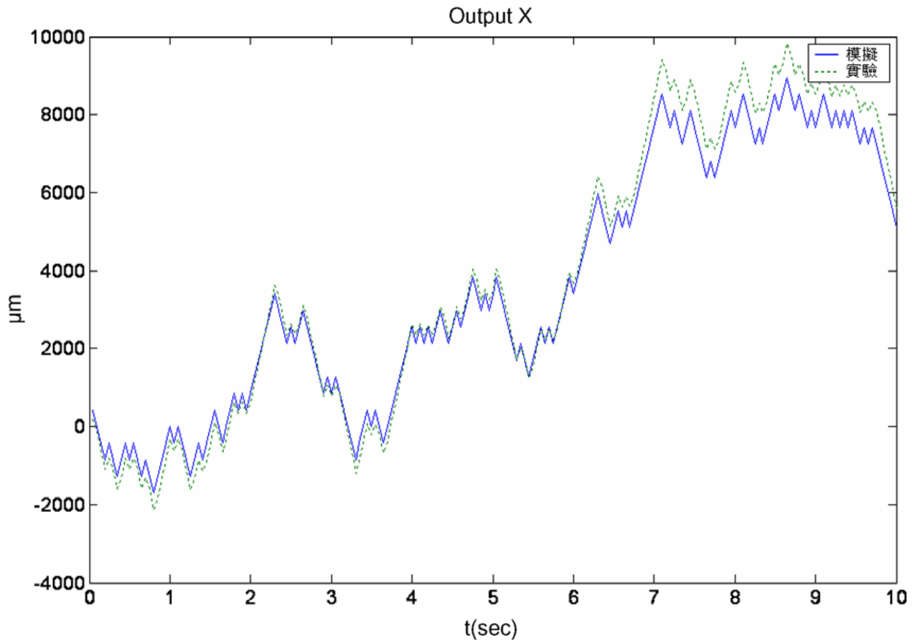


Fig. 8. Comparison between experimental and simulated displacement in *x*-axis.

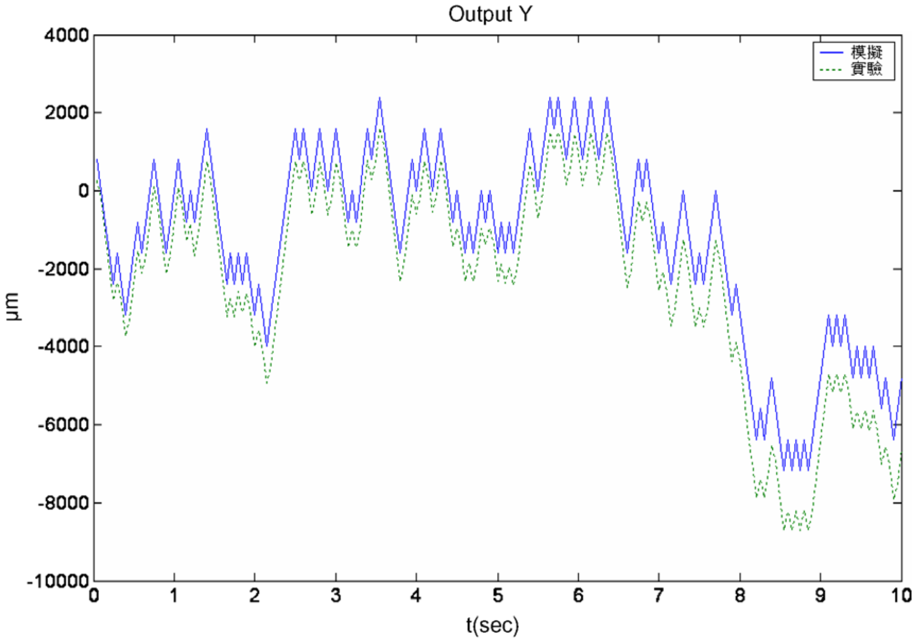


Fig. 9. Comparison between experimental and simulated displacement in *y*-axis.

4.3. Best-fit coordination/allocation by genetic algorithm

The on-line best-fit solution for coordination in Fig. 10 was newly inserted as the 5th step Eqs. (24), (25) in the following procedures (Lue et al., 2005):

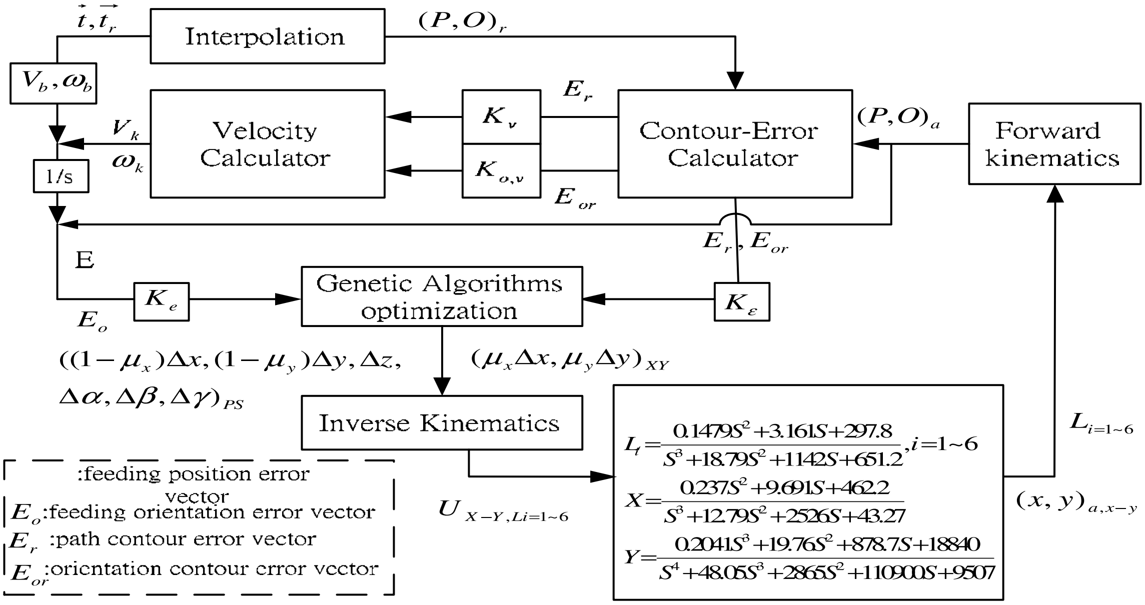

$\vec{t}, \vec{t_r}$
Interpolation
$(P,O)_r$
V_b, ω_b
V_k
ω_k
Velocity Calculator
K_v
$K_{o,v}$
E_r
E_{or}
Contour-Error Calculator
$(P,O)_a$
Forward kinematics
1/s
E
E_r, E_{or}
K_e
Genetic Algorithms optimization
K_ε
E_o
$((1-\mu_x)\Delta x, (1-\mu_y)\Delta y, \Delta z, \Delta\alpha, \Delta\beta, \Delta\gamma)_{PS}$
$(\mu_x \Delta x, \mu_y \Delta y)_{XY}$
Inverse Kinematics
$L_t = \frac{0.1479S^2+3.161S+297.8}{S^3+18.79S^2+1142S+651.2}, i=1\sim6$
$X = \frac{0.237S^2+9.691S+462.2}{S^3+12.79S^2+2526S+43.27}$
$Y = \frac{0.2041S^3+19.76S^2+878.7S+18840}{S^4+48.05S^3+2865S^2+110900S+9507}$
$L_{i=1\sim6}$
$U_{X-Y,Li=1\sim6}$
$(x, y)_{a,x-y}$
:feeding position error vector
E_o:feeding orientation error vector
E_r :path contour error vector
E_{or}:orientation contour error vector

Fig. 10. Block diagram of the GA optimized multi-axis machine tool system.

(1) Computation, conversion and storage of parameter information $S(u, v)$ for centroid trajectory $(P, O)_r$ of upper platform.
(2) Computation of forward kinematics using actual positions $L_{t,i=1–6}$ from optical scales to obtain actual position and orientation of the centroid $(P,O)_a$ and to calculate contour errors E_r, E_{or}.
(3) Formation of new reference position and orientation of the centroid:

$$P_f(n) = P_f(n-1) + TV \tag{18}$$

$$O_f(n) = O_f(n-1) + T\omega \tag{19}$$

From which the feeding errors are calculated as:

$$E_P(n) = E_P(n-1) + [P_f(n) - P_a(n)] \tag{20}$$

$$E_o(n) = E_o(n-1) + [O_f(n) - O_a(n)] \tag{21}$$

(4) Creation of driving commands using both contour errors and feeding errors:

$$U_P(n+1) = K_e E_P(n) + K_{er} E_r(n) \tag{22}$$

$$U_O(n+1) = K_o E_o(n) + K_{er} E_{r,o}(n) \tag{23}$$

Let $\Delta p_{n+1} = (\Delta x, \Delta y, \Delta z, \Delta\alpha, \Delta\ \beta, \Delta\gamma)_{n+1} = (U_P(n+1), U_O(n+1))$

(5) Finding best-fit solution for movement coordination/allocation between parallel and serial structure by genetic algorithm:

$$\begin{aligned}\Delta p_{n+1,PS} &= ((1-\mu_x)\Delta x, (1-\mu_y)\Delta y, \Delta z, \Delta\alpha, \Delta\beta, \Delta\gamma)_{n+1} \\ &= (U_{P,PS}(n+1), U_{O,PS}(n+1))\end{aligned} \tag{24}$$

$$\Delta p_{n+1,XY} = (\mu_x \Delta x, \mu_y \Delta y)_{n+1} = U_{P,XY}(n+1) \tag{25}$$

(6) After Inverse Kinematics computation the coordinated commands are calculated and issued to the respective driving motor.

$$\begin{aligned}&U_{Li,i=1–6} = \text{Inverse} - \text{Kinematics}(U_{P,PS}(n+1), U_{O,PS}(n+1)) \\ &U_{P,XY}(n+1) = \Delta p_{n+1,XY} = (\mu_x \Delta x, \mu_y \Delta y)_{n+1}\end{aligned} \tag{26}$$

(7) Repeat steps 2–6 until the trajectory is to the end.

4.4. Tracking with GA-selected system gains but no motion allocation

Three trajectories are used to evaluate the effects of GA best-fit solution. In first phase of evaluation, the systems are in the optimum gains configuration. But there is no motion allocation and the weighting ratio μ_x and μ_y are both set to 1 which means the XY-base is exploited to its utmost while the parallel structure is responsible only for $(0, 0, \Delta z, \Delta\alpha, \Delta\ \beta, \Delta\gamma)$.

Trajectory one evaluates the effects of position-related gains, trajectory two evaluates the effects of orientation-related gains and the trajectory three evaluates the effects of both position- and orientation-related gains.

4.4.1. Trajectory one (position trajectory)

Trajectory one (see Fig. 11) is of the following mathematical representation:

$X(t) = t \cdot \cos(t); Y(t) = t \cdot \sin(t);\ \ Z(t) = 3t + 260\ \ \alpha(t) = 0;\ \ \beta(t) = 0;\ \ \gamma(t) = 0\ \ T = 0\text{–}10$ s;sampling time = 0.02sec Position = [x(t),y(t),z(t), α(t),β(t),γ(t)] Specifications of variables used in GA are listed in Table 2.

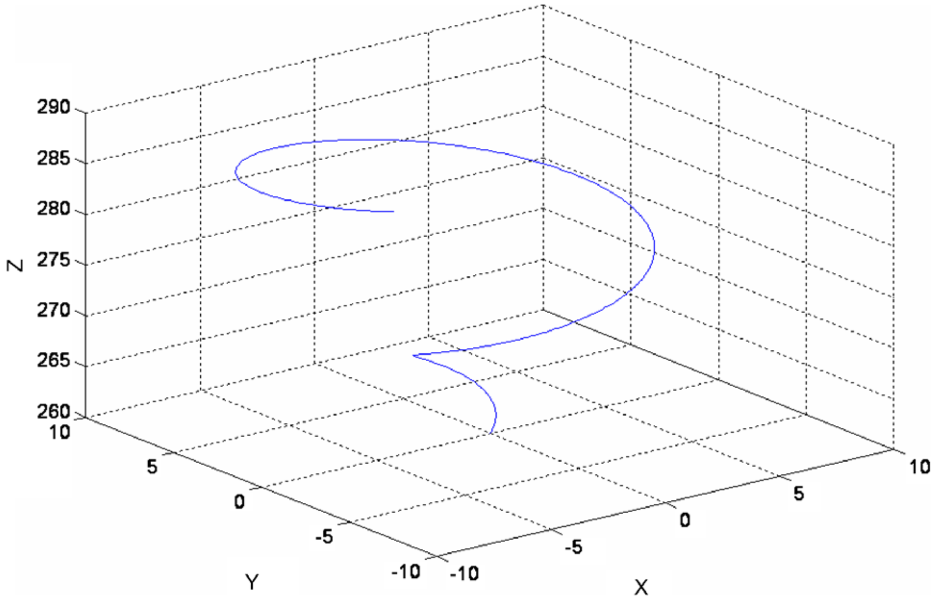


Fig. 11. Trajectory one, no orientation variation.

Table 2
Genetic Algorithm parameters for trajectory one

Variables	Values
String bits	K_e, K_ε, K_v are all 20 bits
Population	40
Generations	60
Reproduction strategy	Roulette wheel selection
Crossover strategy	Uniform
Crossover rate P_c	0.85
Mutation rate P_m	0.006
Range of system parameter K_e	0–5
Range of system parameter K_ε	0–5
Range of system parameter K_v	0–2000

Fig. 12 and Table 3 show the comparison of contour errors produced by US, CCS and CCPM for tracking trajectory one. Since US system has no cross-coupled compensation, it produces the greatest contour errors. CCS and CCPM are comparable systems while the latter offers slightly better results. Cross-coupling compensation of CCS and CCPM reduce contour errors but the errors fluctuate.

4.4.2. Trajectory two (orientation trajectory)

Trajectory two (see Fig. 13) is described by the following equation:

$X(t) = 0$; $Y(t) = 0$; $Z(t) = 130$ $\alpha(t) = 7\cos(t) \times \pi/180$; $\beta(t) = 7\sin(t) \times \pi/180$; $\gamma(t) = 0$ $T = 0–10$ s; sampling time = 0.02sec Position = $[x(t),y(t),z(t),\alpha(t),\beta(t),\gamma(t)]$

Specifications of variables used in GA are listed in Table 4.

The procedures in the GA computation are similar to those in position trajectory but only the orientation-related gains (i.e. $K_{e\alpha}$, $K_{e\beta}$,$K_{\varepsilon\alpha}$,$K_{\varepsilon\beta}$,$K_{V\alpha}$, $K_{V\beta}$) exist.

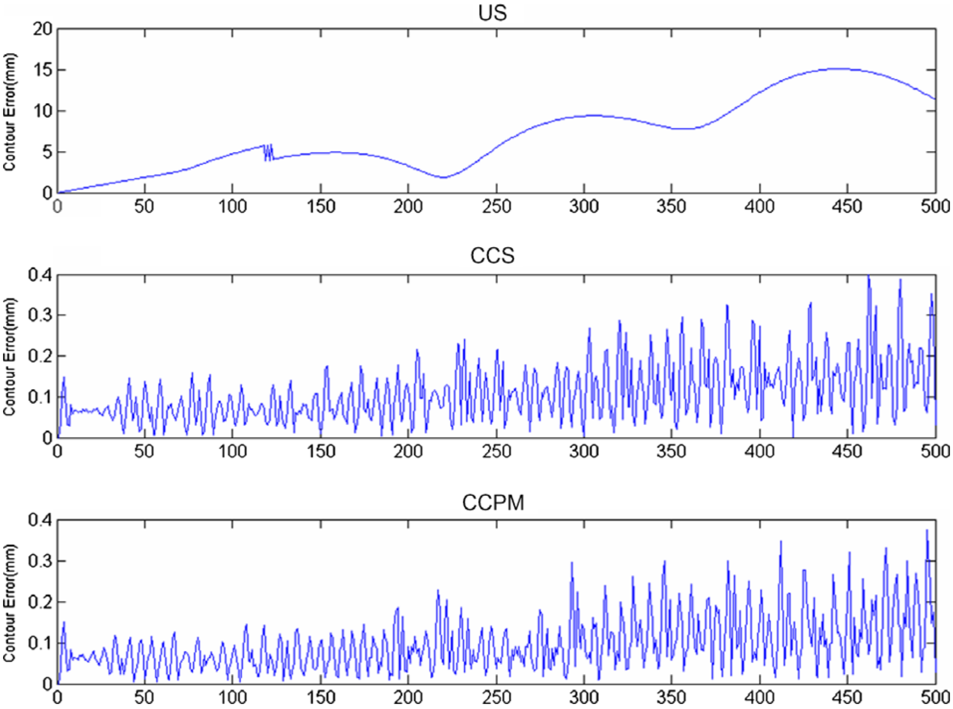


Fig. 12. Comparison of position contour errors produced by different leveled control system for tracking trajectory one.

Table 3
System gains and tracking results for trajectory one

		K_v	K_ε	K_e	IAE	|Er,max|
US	*x*	0	0	0.0004	7.0059	15.0577
	y			0.0001		
	z			0.0001		
CCS	*x*	0	1.9197	0.0004	0.1085	0.3994
	y		1.9286	0.0001		
	z		1.8181	0.0001		
CCPM	*x*	1967	1.9197	0.0004	0.0972	0.3737
	y	1700.2	1.9286	0.0001		
	z	1237.9	1.8181	0.0001		

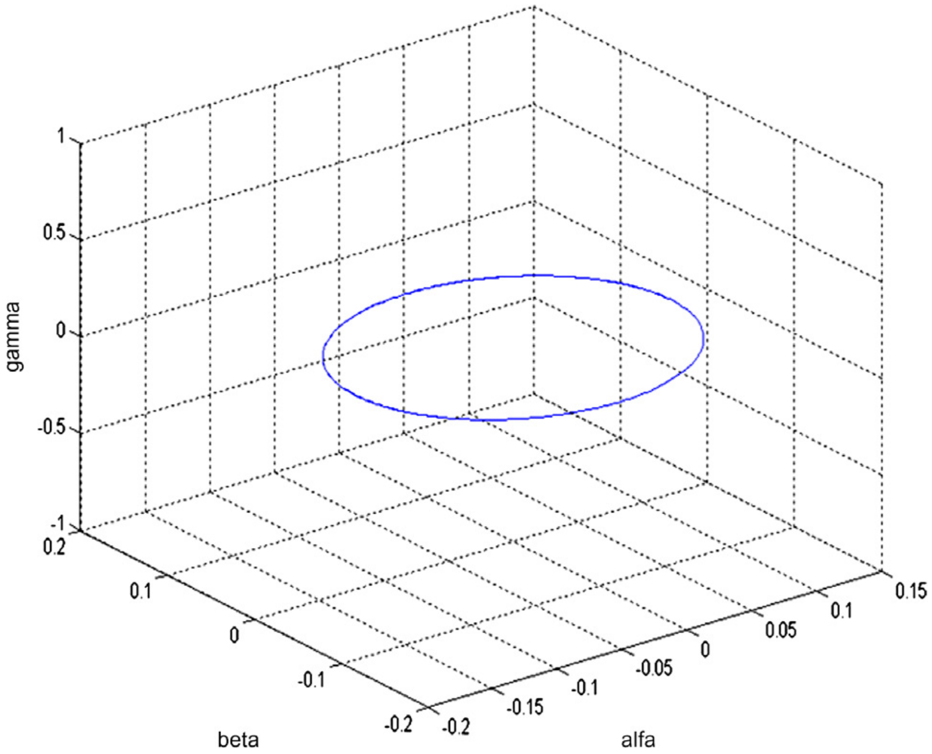


Fig. 13. Trajectory two, an orientation trajectory.

Table 4
Genetic Algorithm parameters for trajectory two

Variables	Values
String bits	K_e, K_ε, K_v are all 20 bits
Population	40
Generations	60
Reproduction strategy	Roulette wheel selection
Crossover strategy	uniform
Crossover rate P_c	0.85
Mutation rate P_m	0.006
Range of system parameter K_e	0–5
Range of system parameter K_ε	0–5
Range of system parameter K_v	0–100

Fig. 14 shows the contour errors produced by US, CCS and CCPM for tracking trajectory two. It is seen the US system produced the greatest contour errors but the there are no error fluctuations. Cross-coupled system suppressed the contour errors but the errors fluctuated while the trend of overall error profile remained the same. Interesting is that CCPM brings about the best contour precision and error profile did not exist. This is the effect of "speed pre-compensation" which has been proven a curvature cruncher. Table 5 gives numerical values of gains and errors, which the performance of CCS and CCPM can be readily recognized.

4.4.3. Trajectory three (position and orientation trajectory)

Trajectory three (Fig. 15), which contains both position and orientation variations, is described by the following equation:

$X(t) = t \cdot \cos(t)$; $Y(t) = t \cdot \sin(t)$; $Z(t) = t + 130$ $\alpha(t) = t \cdot \cos(t) \times \pi/180$; $\beta(t) = t \cdot \sin(t) \times \pi/180$; $\gamma(t) = 0$
T = 0–10 s; sampling time = 0.02 s Position = $[x(t), y(t), z(t), \alpha(t), \beta(t), \gamma(t)]$

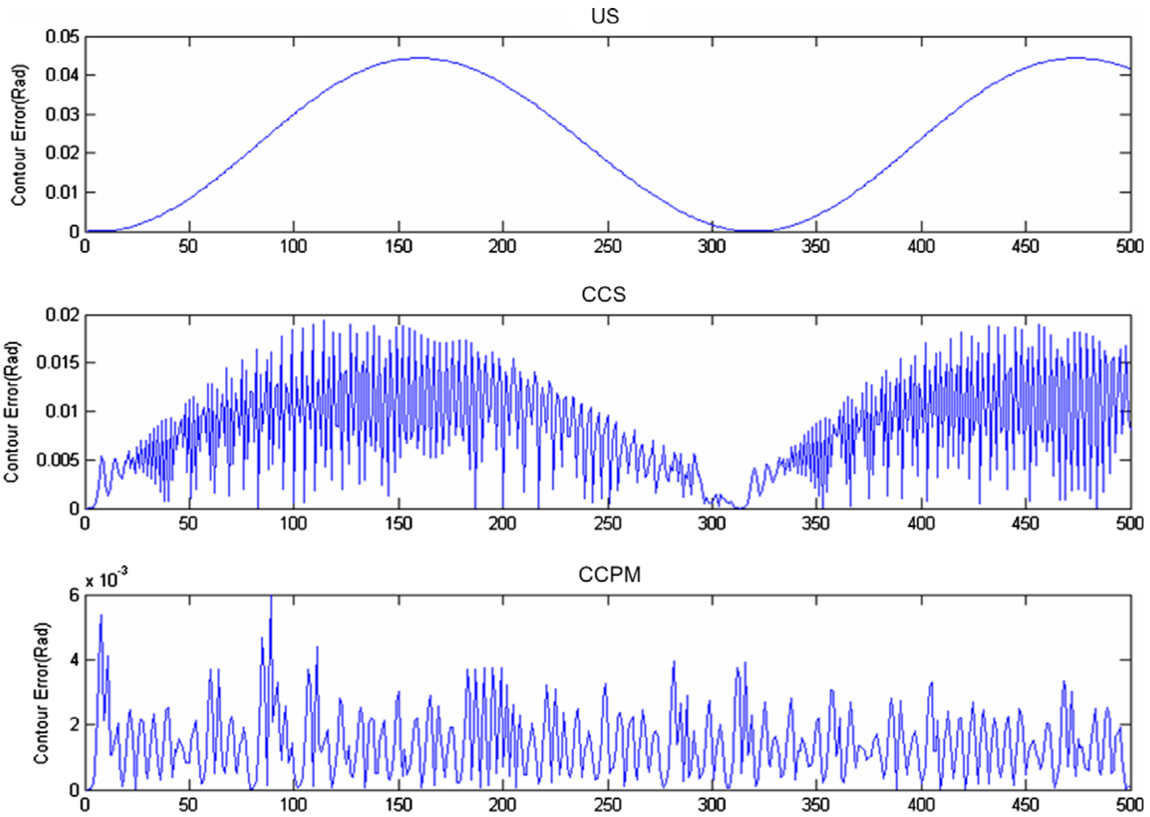


Fig. 14. Orientation contour errors produced by US, CCS and CCPM control system for tracking trajectory two.

Table 5
Optimum system gains and tracking results for trajectory two

		K_v	K_ε	K_e	IAE	|Er,max|
US	α	0	0	0.5	0.0231	0.0444
	β			0.595		
	γ			0		
CCS	α	0	1.9928	0.5	0.0082	0.0193
	β		1.8912	0.595		
	γ		0	0		
CCPM	α	39.583	1.9928	0.5	0.0014	0.006
	β	39.991	1.8912	0.595		
	γ	0	0	0		

Specifications of variables used in GA for trajectory three are listed in Table 6.

The procedures in the GA computation are similar to those in trajectory one or two but with all gains present (i.e. K_{ex},K_{ey},K_{ez}, $K_{\varepsilon x}$,$K_{\varepsilon y}$,$K_{\varepsilon z}$,K_{Vx},K_{Vy}, K_{Vz} and $K_{e\alpha}$,$K_{e\beta}$,$K_{\varepsilon\alpha}$, $K_{\varepsilon\beta}$,$K_{V\alpha}$,$K_{V\beta}$ are all non-zero).

Fig. 16 shows the tracking results of US, CCS and CCPM system. Again the system without cross-coupling compensation (US) has the greatest yet least dynamic contour errors. Cross-coupling compensation (CCS) suppressed the magnitude of contours errors but the errors were more dynamic, and the overall error envelope did not carry any similarity to that of the US system. Table 7 gives detailed information about error index IAE and the maximum contour error. It is seen that Pre-compensated cross-coupling compensation (CCPM) has slightly lower IAE for both orientation and position but only slightly lower maximum error for orientation than CCS. Fig. 17 tells a different story for orientation trajectory, in which all three tracking carried similar

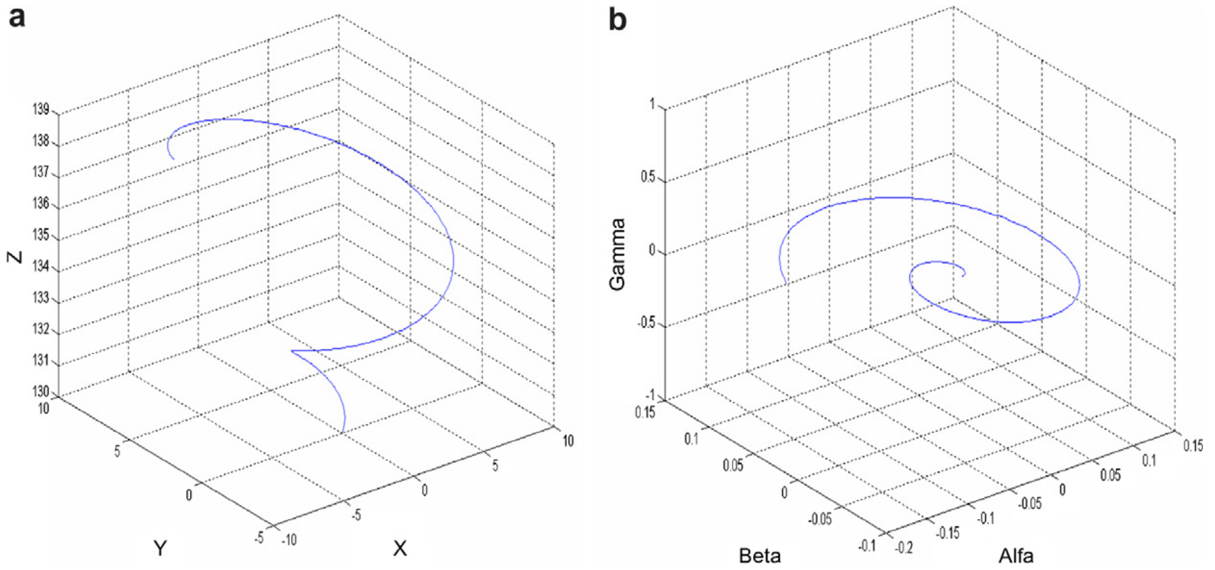


Fig. 15. a. Position trajectory of trajectory three. b. Orientation trajectory of trajectory three.

Table 6
Genetic Algorithm parameters for trajectory three

Variables	Values
String bits	K_e, K_ε, K_v are all 20 bits
Population	40
Generations	60
Reproduction strategy	Roulette wheel selection
Crossover strategy	Uniform
Crossover rate P_c	0.85
Mutation rate P_m	0.006
Range of system parameter K_e	0–5
Range of system parameter K_ε	0–5
Range of system parameter K_v	0–1000

error envelope. The US system subjected to the most dynamic orientation errors while the CCPM the least. This phenomenon is the reverse of the results shown in Fig. 16 for position trajectory.

4.5. Tracking with optimized motion allocation between redundant degrees of freedom

The tracking in the last section were done with optimum system gains and one pre-occupation: all x-, y-displacement were done by XY-base and the driving activity of parallel structure was hold minimum. This is based on the idea that conventional serial structure is simple and mature and hence superior in tracking planar trajectory. However, it is interesting to see if a motion allocation between parallel and serial structure makes any sense.

Genetic algorithm is newly implemented in Eqs. (24) and (25) to find the optimized motion allocation, i.e. the optimized μ_x and μ_y.

Table 8 lists the variables used in GA implementation for motion allocation.

4.5.1. Trajectory one (position trajectory) tracked with motion allocation

Table 9 is the comparison between tracking without (μ_x,μ_y= 1) and with optimum motion allocation for trajectory one. It shows the motion allocation reduced the maximum contour error, but generated bigger

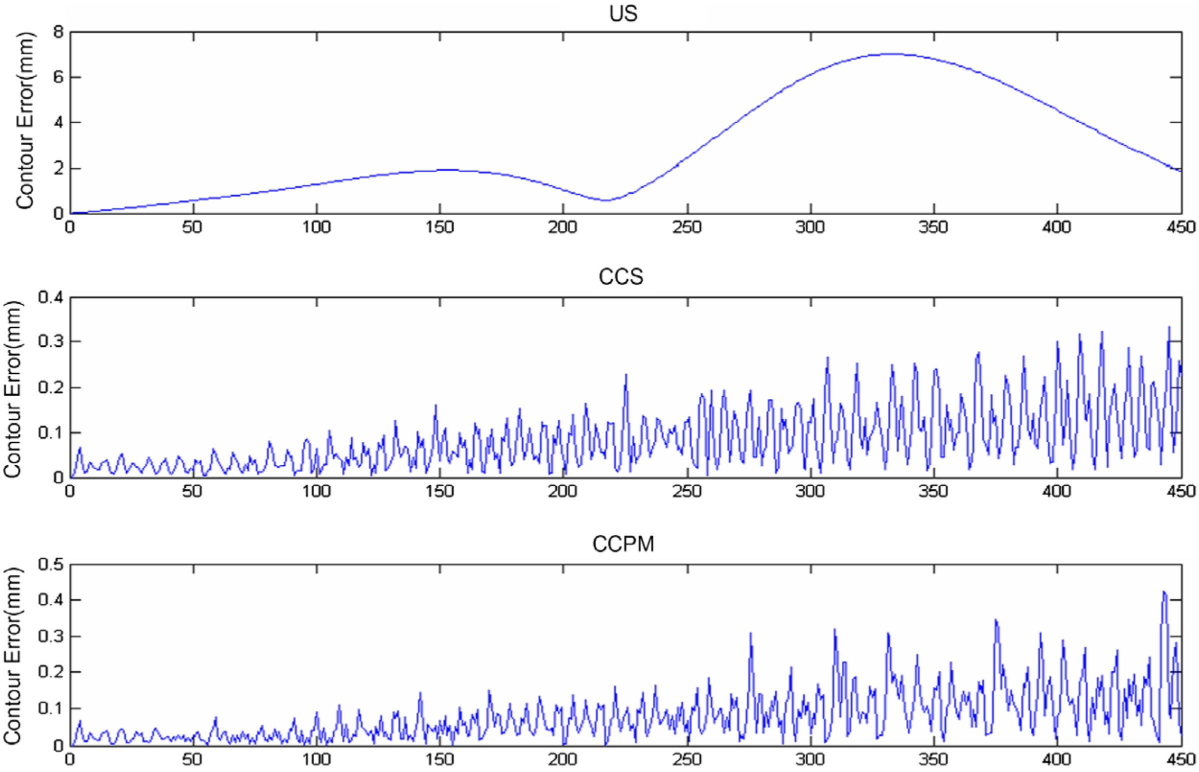


Fig. 16. Comparison of position contour errors produced by US, CCS and CCPM control system for tracking trajectory three.

Table 7
Optimum system gains and tracking results for trajectory three

		K_v	K_ε	K_e	IAE	|Er,max|
US	x	0	0	0.000498	2.8384	6.9873
	y			0.000473		
	z			0.000130		
	α			1.999600	0.0355	0.099
	β			0.290500		
	γ			0.000000		
CCS	x	0	1.9701	0.000498	0.0839	0.3345
	y		1.6925	0.000473		
	z		1.7192	0.000130		
	α		0.0071	1.999600	0.0355	0.093
	β		0.0194	0.290500		
	γ		0	0.000000		
CCPM	x	954.8721	1.9701	0.000498	0.0788	0.4247
	y	802.3184	1.6925	0.000473		
	z	760.4721	1.7192	0.000130		
	α	2.3056	0.0071	1.999600	0.0154	0.0704
	β	3.8527	0.0194	0.290500		
	γ	0	0	0.000000		

IAE. Fig. 18 shows the contour errors obtained without and with motion allocation. It is apparently seen that the error fluctuation is bigger for not allocated tracking. The IAE of allocated tracking is slightly higher, this is because of the smaller error fluctuation, i.e., the differences between upward and downward error peaks are smaller. It can be said that, thanks to the smaller error fluctuation, allocated tracking produced smoother

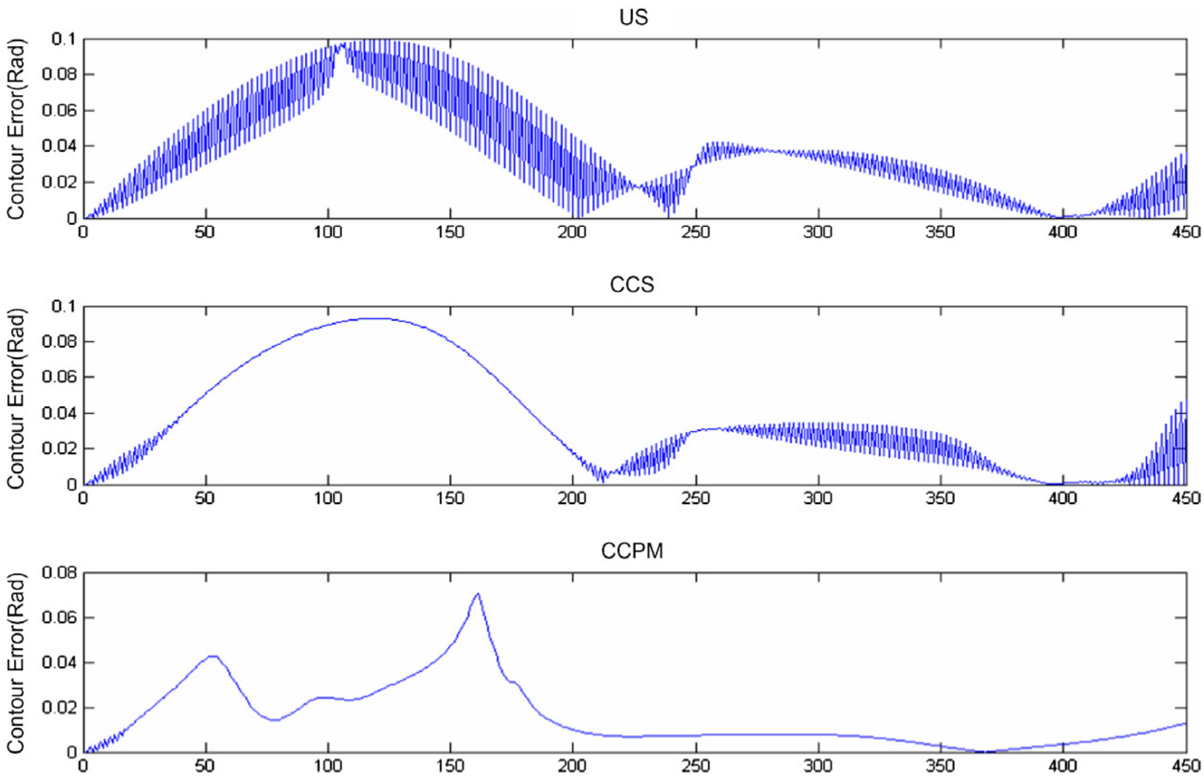


Fig. 17. Comparison of orientation contour errors produced by US, CCS and CCPM control system for tracking trajectory three.

Table 8
Genetic algorithm variables

Variables	Values
String bits	μ_x,μ_y are 10 bits
Population	20
Generations	10
Reproduction strategy	Roulette wheel selection
Crossover strategy	Uniform
Crossover rate P_c	0.85
Mutation rate P_m	0.006
Range of system parameter μ_x	0–1
Range of system parameterμ_y	0–1

Table 9
Contour error comparison between “without” and “with” motion allocation

	IAE	|Er,max|
CCPM	0.0972	0.3737
CCPM(GA)	0.11	0.264

trajectory. Fig. 19 shows the variation of motion allocation with time. It is to note that the calculations of μ_x and μ_y are on-line. Figs. 20, 21 are spatial trajectory tracked without and with motion allocation, respectively. It is seen that motion allocation produced smoother trajectory.

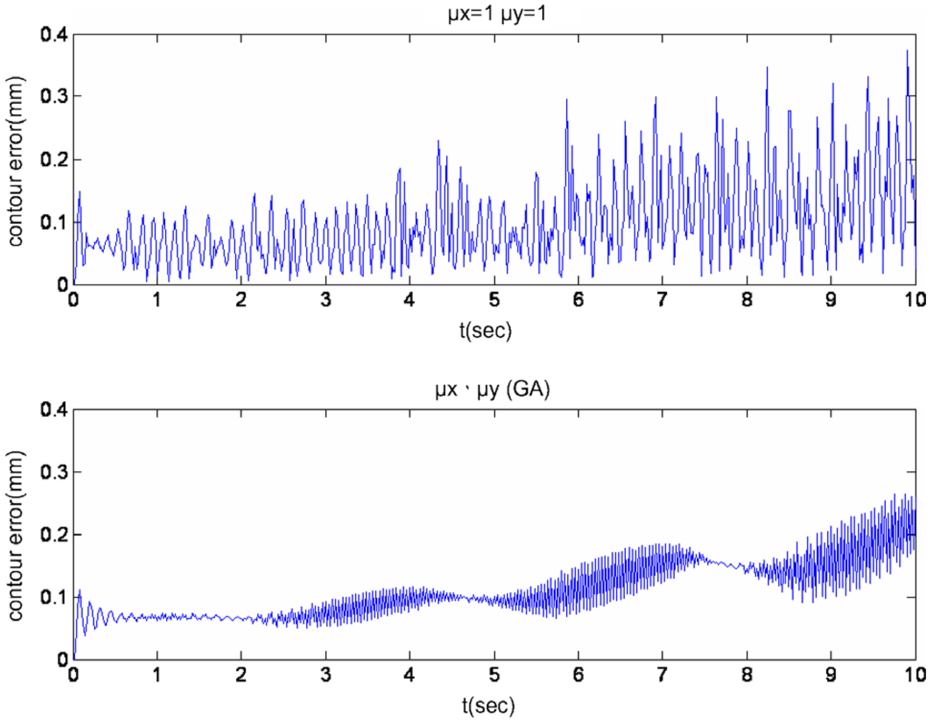


Fig. 18. Contour errors without (upper) and with (lower) motion allocation.

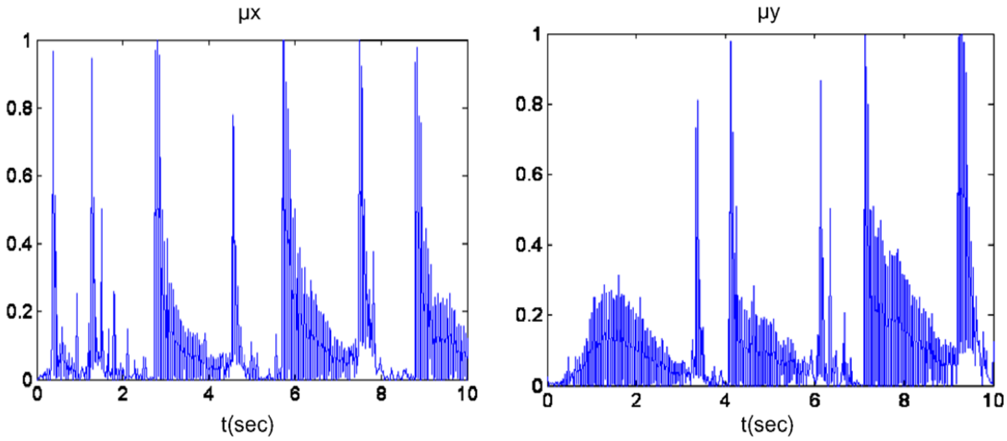


Fig. 19. Time history of μ_x,μ_y.

4.5.2. *Trajectory two (orientation trajectory) tracked with motion allocation*

Trajectory two is a pure orientation trajectory with $(\Delta x, \Delta y)$ always equal to zero. For trajectory of this kind, the linear motion allocation is irrelevant.

4.5.3. *Trajectory three (position and orientation trajectory) tracked with motion allocation*

Table 10 is the comparison between results without and with motion allocation. Again it is seen the motion allocation produced smaller maximum contour error but slightly bigger average contour error. Table 11 is the same comparison for orientation. As could be expected, the motion allocation in x, y has no effects on the orientation. Fig. 22 shows the contour errors without and with motion allocation. The optimum motion allocation

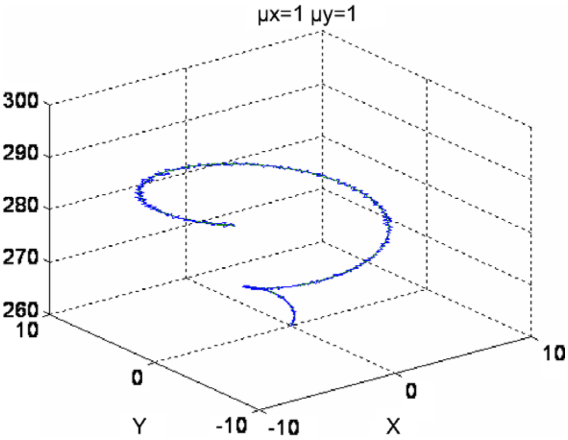


Fig. 20. Spatial trajectory tracked without motion allocation.

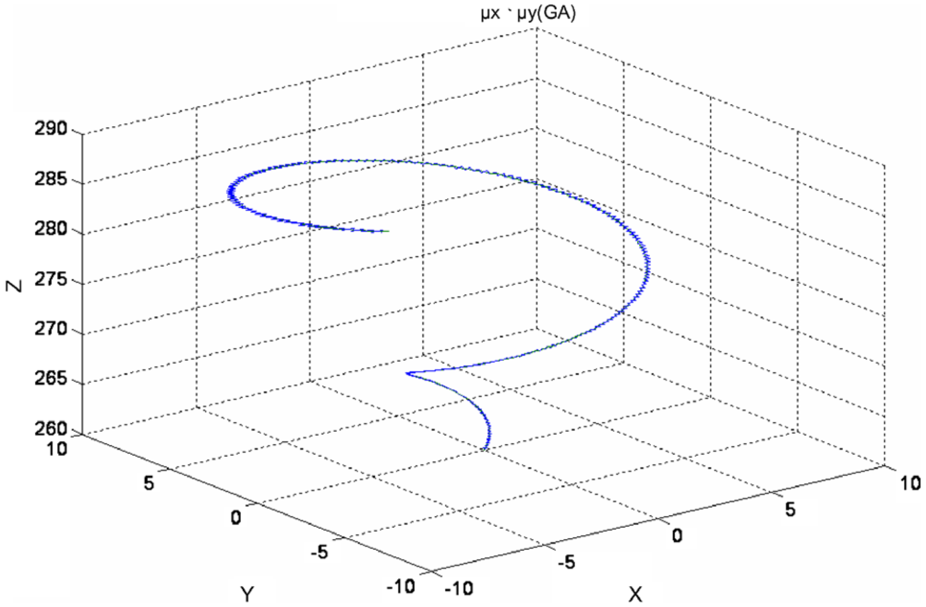


Fig. 21. Spatial trajectory tracked with motion allocation.

Table 10
Contour errors without and with motion allocation

	IAE	|Er,max|
CCPM	0.0788	0.4247
CCPM(GA)	0.0903	0.2343

brings about quite different results of smaller error fluctuation, although the average error index IAE shows slight (15%) increase. Fig. 23 shows the fact of irrelevance of linear motion allocation on orientation tracking.

Fig. 24 is the time histories of on-line calculated μ_x, μ_y. This figure shows that the optimum motion allocation does not coincide with the simple idea of exploiting XY-base to its utmost (i.e. set μ_x,μ_y=1). Fig. 25 shows the spatial trajectory without and with optimized μ_x,μ_y. Fig. 26 shows the orientation trajectory without and with optimized μ_x,μ_y.

Table 11
Orientation errors without and with motion allocation

	IAE	|Er,max|
CCPM	0.0154	0.0704
CCPM(GA)	0.0154	0.0704

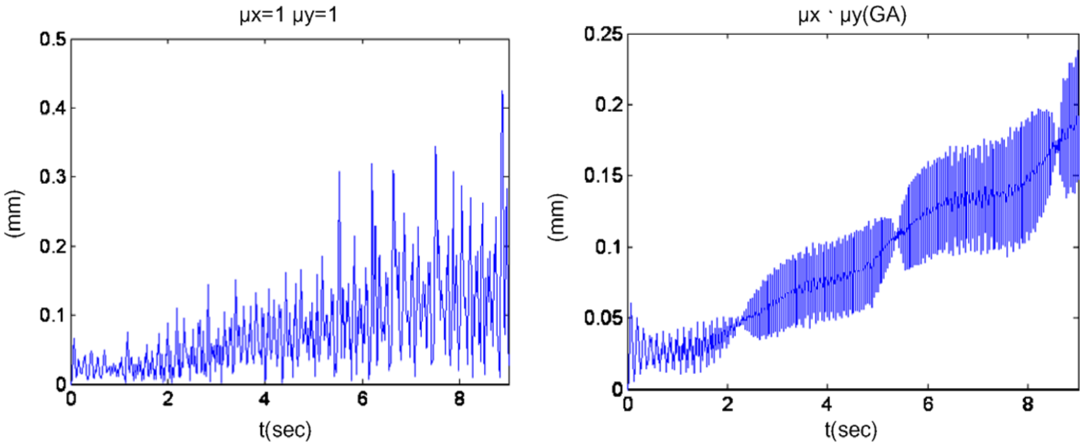


Fig. 22. Contour errors without (left) and with (right) motion allocation.

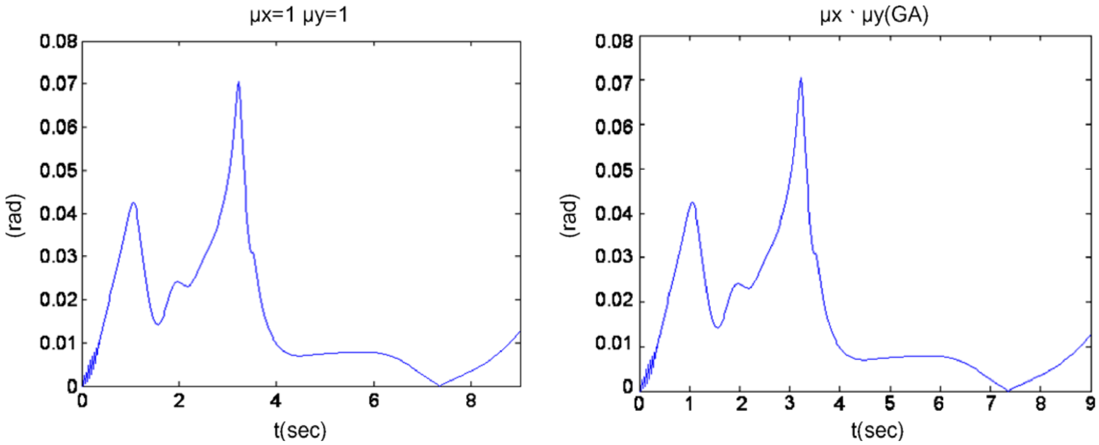


Fig. 23. Orientation contour error without (left) and with (right) optimized μ_x,μ_y.

5. Conclusions

The sophistication of modern multi-axis machine tools increases the number of system gains. Besides, modern multi-axis machine tools often possess redundant degrees of freedom and the adequate exploitation of different degrees of freedom posts a new challenge. The configuration of system gains and motion allocation of sophisticated manufacturing machine tools, either mathematically or by trial and error, are often very difficult. Genetic algorithm has the nature of solving complex, conflicting and mathematically difficult problems. Although the GA needs longer computation time and has the nature of unpredictability, its potential contribution to modern manufacturing system is still worth investigating.

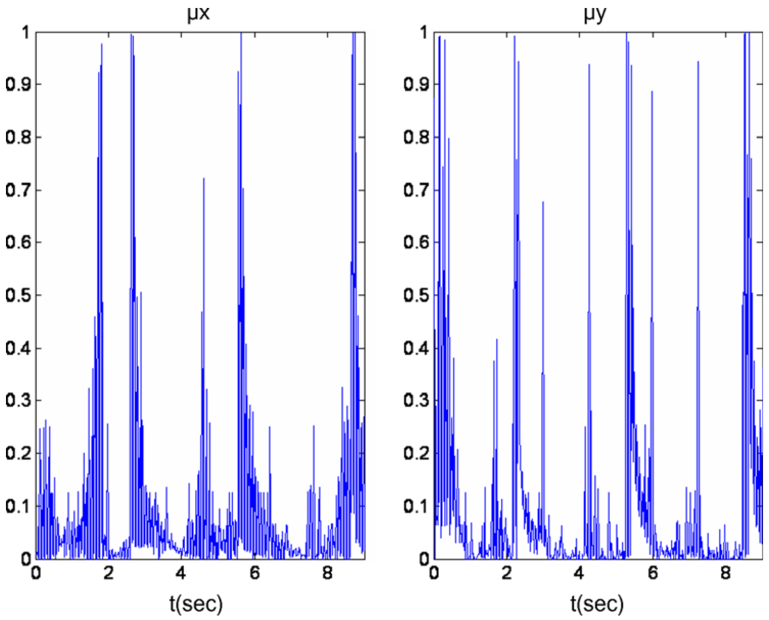


Fig. 24. Time history of μ_x(left), μ_y(right).

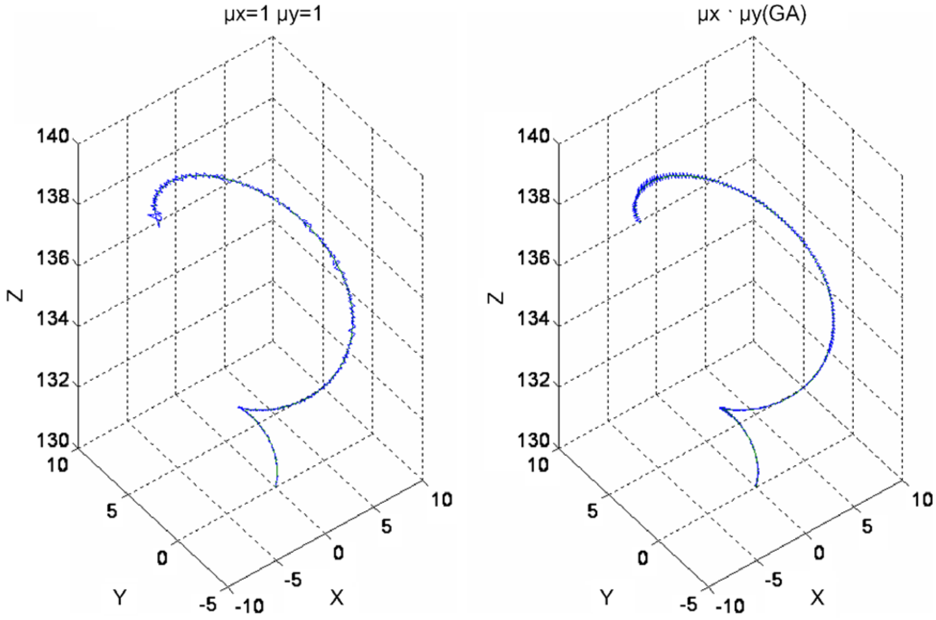
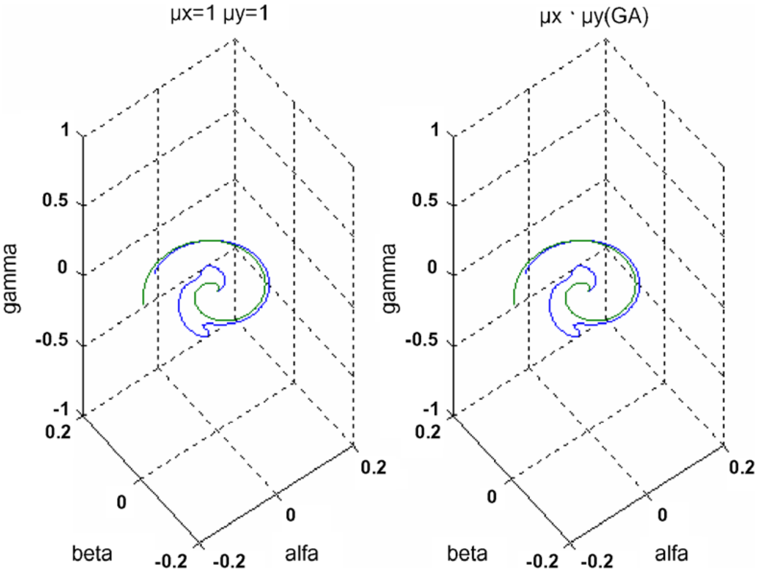


Fig. 25. The spatial trajectory without (left) and with (right) optimized μ_x,μ_y.

This work explored the genetic algorithms in finding the best-fit solution of system gains and motion allocation between redundant degrees of freedom for modern multi-axis machine tools. Standard GA was developed for a hybrid multi-axis system with 18 system gains. The following points can be concluded:

(1) An off-line optimum configuration of system gains by GA is successful and meaningful. The off-line configuration can be seen as a kind of system design which offers a good starting point for further on-line adaptation.


μx=1 μy=1

μx · μy(GA)

Fig. 26. Orientation contour errors without (left) and with (right) optimized μ_x,μ_y.

(2) In the best-fit solution of motion allocation, an on-line GA operation to the working frequency of 20 ms is reached. This time depends on the population size, string length, maximum number of generations, etc. By suitably configuring these factors, a reduction of selection time while maintaining an acceptable motion allocation quality is thinkable.

(3) The operation of redundant degrees of freedom of modern multi-axis machine tools is an algorithm dependant problem which does not coincide with the instinct idea of exploiting the best part of machine structure.

(4) A GA-selected best-fit motion coordination/allocation results in smaller absolute contour error but slightly bigger average errors, which reflects smaller error fluctuation and smoother tracked profile. A GA-selected best-fit motion coordination/allocation may require less finishing after machining.

In the experience gathered in this work, the unpredictability was not observed and the overall results are encouraging. An illustrating future is the integration of genetic algorithms with fuzzy logic system or neural networks. The two weakness of GA, longer computation time and the unpredictability of evolution, may be relieved or even removed if combined with fuzzy logic system or neural networks. The combination could lead to emerging new technology for machines used in manufacturing. It should also be noted that the problem of unpredictability, stability and safety of GA requires tremendously extensive studies in the future before it becomes fail-proof. This can be of future interests of studies.

Acknowledgement

The authors thank the National Science Council of Taiwan, R.O.C. for the support of this research under Contract No. NSC-93-2212-E-009-021.

References

Chin, J.-H., Cheng, Y.-M., & Lin, J.-H. (2003). Improving contour accuracy by Fuzzy logic enhanced cross-coupled precompensation method. *Robotics and Computer Integrated Manufacturing, 20*, 65–76.

Chin, J.-H., & Lin, S.-T. (1997a). "The path precompensation method for flexible arm robot". *Robotics and Computer-Integrated Manufacturing, 13*(3), 203–215.

Chin, J.-H., & Lin, T.-C. (1997b). "Cross-coupled precompensation method for the contouring accuracy of computer numerically controlled machine tools". *International Journal of Machine Tools & Manufacture, 37*(7), 947–967.

Chin, J.-H., & Lin, H.-W. (1999). "The algorithms of the cross-coupled precompensation method for generating the involute-type scrolls". *ASME J. Dynamic Systems, Measurement, and Control, 121*, 96–104.

Chin, J.-H., & Tsai, H.-C. (1993). A path algorithm for robotic machining. *Robotics and Computer-Integrated Manufacturing, 10*(3), 185–198.

Fogel, D. B. (1988). An evolutionary approach to the traveling salesman problem. *Biological Cybernetics, 60*, 139–144.

Goldberg, D. E. (1989). *Genetic algorithms: in search, optimization and machine learning*. Reading, MA: Addison-Wesley.

Hussain N, Al-Duwaish, and Zakariya, M., Al-Hamouz, 1998, "A genetic approach to the selection of the variable structure controller feedback gains", IEEE Int. Conf. on control applications Trieste, Italy, 1-4 Sept., pp.227-231.

Koren, Y. (1980). Cross-coupled biaxial computer control for manufacturing systems. *ASME Trans Journal of Dynamic System, Measurement and Control, 102*(4), 265–272.

Koren, Y., & Lo, C.-C. (1991). Variable gain cross coupling control for contouring. *Annals of the CIRP, 40*, 371–374.

Lue, C.-W., Cheng, Y.-M., & Chin, J.-H. (2005). System structure and contour tracking for a hybrid motion platform. *The International Journal of Advanced Manufacturing Technology, 26*, 1388–1396.

Man, K. F., Tang, K. S., & Kwong, S. (1996). Genetic algorithms: concepts and applications. *IEEE Transactions On Industrial Electronics, 43*(5), 519–533.

Sarachik, P. J., & Ragazzini, R. (1957). A two dimensional feedback control system. *Transactions AIEE, 76*, 55–61.

Scott, A. (1990). An introduction to genetic algorithms. *AI Expert, 4*(3), 49–53.

Srinivas, M., & Patnaik, L.-M. (1949). Genetic algorithms: a survey. *IEEE Computer*, 17–26.

Tarng, Y.-S., Chuang, H.-Y., & Hsu, W.-T. (1999). Intelligent cross-coupled fuzzy feedrate controller design for CNC machine tools based on genetic algorithms. *International Journal of Machine Tools & Manufacture, 39*, 1673–1692.

Yamamoto, K., & Inoue, O. (1995). New evolutionary direction operator for genetic algorithms. *AIAA J., 33*(10), 1990–1993.

Zakariya, M., Al-Hamouz Hussain, N., & Al-Duwaish (1989). A new variable structure DC motor controller using genetic algorithms". *IEEE Transactions On Industrial Electronics, 1*(2), 1669–1673.