

# Chapter 10

## Development of an Improved Genetic Algorithm for Resolving Inverse Kinematics of Virtual Human's Upper Limb Kinematics Chain

Gangfeng Deng, Xianxiang Huang, Qinhe Gao, Ying Zhan and Quanmin Zhu

**Abstract** Inverse kinematics is the key technique in virtual human motion control and it is difficult to obtain the solutions by using geometric, algebraic, or iterative algorithms. In this chapter, an Improved Genetic Algorithm (IGA) is proposed to resolve the inverse kinematics problem in upper limb kinematics chain (ULKC). First, the joint-units of ULKC and its mathematical models are constructed by using D–H method; then population diversity and population initialization are accomplished by simulating human population, and the adaptive operators for mutation are designed. The simulation results show that compared with the Standard Genetic Algorithm (SGA), the IGA can provide higher precise solutions in searching process and avoid “premature” stop or inefficient searching in later stage with high probability.

**Keywords** Upper limb kinematics chain · Inverse kinematics · D–H method · Genetic algorithm · Population initialization

### 10.1 Introduction

Control of joint angles is the key technique in virtual human motion control. The popular approaches include geometric control, behavior control, physical control, motion capture control, and synthesis control [1], which have different characteristics

---

G. Deng (✉) · X. Huang · Q. Gao · Y. Zhan  
Xi'an Research Institute of High Technology, Xi'an, 710025, People's Republic of China  
e-mail: 20378481@qq.com

Q. Zhu  
College of Chemical Engineering, China University of Petroleum, Qingdao, 266580,  
People's Republic of China

Q. Zhu  
Department of Engineering Design and Mathematics, University of the West of England,  
Coldharbour Lane, Bristol, BS16 1QY, UK

[2, 3]. The kinematics control, one of the geometric control methods, has been the most widely used in human motion control while the end-effector's position and orientation are determined.

Kinematics control algorithm contains forward kinematics (FK) algorithm and inverse kinematics (IK) algorithm. Calculating the position and orientation of the end-effector from the joint Cartesian space is known as FK problem and calculating the joint angles from position and orientation of the end-effector is called IK problem. Many traditional methods have been used to resolve the IK problem such as geometric [4], algebraic [5], and iterative methods [6]. However, these methods have their own demerits in solving the IK problem of the mechanical structure or the body structure.

In recent years, Fuzzy [7], Artificial Neural Networks (ANN) [8, 9] and Genetic Algorithm (GA)-based evolutionary approaches have been applied to solve the IK problem for many kinds of manipulators, and great progress has been achieved. Satish [10] uses a 3-layer perceptron neural network to resolve the IK problem in a two degree-of-freedom (DOF) serial chain manipulator, as the training data subdivide degree is only 1.125 degrees, the precision of solution is not good enough. Bassam [11] uses a 4-layer perceptron neural network to resolve the IK problem in 2DOF manipulator, which can avoid the emergence of redundant solutions and improve the precision by dividing the solution space into many sub-spaces, but the complexity of the algorithm is increased on one hand. Reference [12] researches the multi-layer perceptron and functional link artificial neural network, respectively, to compute 2 and 3DOF serial manipulators' inverse solution. When the robotic arm is 3DOF, the solution's precision is less than satisfactory. As the training sample's subdivision degree determines the precision of the solution in neural network, when the body has more than 3DOF, due to the limitation in the number of training samples, it is difficult to obtain a high precise solution. The GA is widely applied to solve the IK problem due to its excellent characteristics of evolutionary optimization; Saleh [13] obtains the IK solutions of the three joints which decide the end-effector position of PUMA560 robot by using the genetic algorithm based on the niche and clustering technique. The method can search four optimal solutions at a time but it is time-consuming—every generation computation spent 4 s; Banga [14] combines GA and Analytical Hierarchy Process (AHP) to resolve three nodes manipulator's IK problem, AHP is used to select a different fitness function to prefer a one robotic arm indicator (such as attitude, friction, adjust time), and GA optimizes the inverse solution according to the fitness function selected by AHP. Due to the fitness function selection mechanism, the method can meet multiple-objectives optimization, but it also increases the probability of obtaining suboptimal solutions. Due to the random selection of initial population and the large searching domain, the computational time of GA optimization is usually long and the algorithm is prone to "premature" and later search slowly. So, scholars study another approach, using GA to optimize the parameters of neural network, and then use this neural network to compute the IK solutions. Liu [15] uses GA to optimize weights and thresholds of BP neural network by applying partition coding method; then the BP neural network is used to compute the 2DOF robot's IK solution. Carlos [16] combines Fuzzy neural network (FNN) algorithm and GA for solving the inverse kinematics problem of a 2DOF

manipulator. This method uses GA to optimize the network structure, the set of rules and membership functions of the FNN, and then this FNN algorithm is used to compute the two joint angle's values of manipulator. These methods will accelerate the optimization process of the inverse solution when the algorithm completes the parameter optimization. The disadvantage is that the precision of the solution depends on the neural network, and the accuracy of the neural network's output depends on its structure, parameters, and training samples. When the dimension of the input and output increases, the number of parameters also will be exponentially increased. And it requires a lot of training samples, which would make the training inefficient or even unattainable. As a simple neural network algorithm, the method mentioned above is only suitable for solving the IK problem in less than 3 DOF; when DOF is equal to or larger than 3, the precision of solutions will be far from satisfactory.

The objective of this study is to resolve the IK problem of virtual human's ULKC which is the most complex, flexible, and commonly used agent of the human body which has 6DOF. This study proposes an improved GA method to resolve the IK problem of ULKC, which can avoid GA falling into "premature" stop or inefficient searching in later stage with high probability.

## 10.2 Mathematical Model of ULKC

In this section, we introduce the basic principle of D–H method, construct the kinematics chain of ULKC, and analyze the difficulty in obtaining solutions from the kinematics equation.

### 10.2.1 Introduction of D–H Method

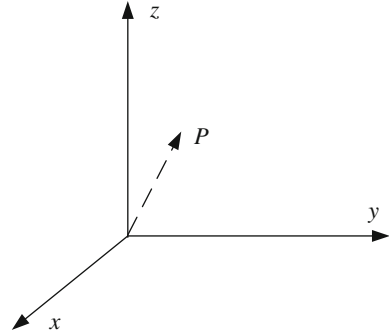
At present, the D–H method is usually used to construct the coordinate of kinematics chain whose mathematical model is used to represent by matrices. The D–H method is based on the transformation of the homogeneous coordinates in three-dimension space, which can link the movement, transformation, and mapping with matrix calculation.

(1) The posture description of kinematics chain

a. Position description

In the three-dimension Cartesian coordinate system, the position of point is usually described by a  $3 \times 1$  vector. Figure 10.1 shows the point  $P$  in the coordinate system  $\{A\}$ . Its homogeneous coordinates are  $P = (P_x, P_y, P_z, 1)^T$ .

**Fig. 10.1** Point  $P$  in the coordinate system



### b. Orientation description

In order to study the kinematics chain's movement and operation, both the object's position and orientation need to be described.  $4 \times 4$  homogeneous matrix  ${}^B_A\mathbf{R}$  represents the orientation of the coordinate system {B} relative to the coordinate system {A}.  $n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z$  are the direction cosine values that three unit vectors  $\vec{x}, \vec{y}, \vec{z}$  of the coordinate system {B} to the coordinate system {A}

$${}^B_A\mathbf{R} = \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.1)$$

${}^B_A\mathbf{R}$  represents the orientation of the coordinate system {B} to the coordinate system {A}.

Rotate  $\theta$  around the axis  $x, y,$  and  $z,$  respectively, and their corresponding rotation matrices can be expressed as follows:

$$\mathbf{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10.2)$$

$$\mathbf{Rot}(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10.3)$$

$$\mathbf{Rot}(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.4)$$

Spatially, when the coordinate systems {A} and {B} have the same orientation and different origins, the translation matrix is as follows:

$${}^B_A \mathbf{trans}(B_x, B_y, B_z) = \begin{bmatrix} 1 & 0 & 0 & B_x \\ 0 & 1 & 0 & B_y \\ 0 & 0 & 1 & B_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.5)$$

It represents the origin position of the coordinate system {B} in the coordinate system {A}.

c. The expression of position and orientation of objects in space

When coordinates system {A} and {B} have different origins and different orientations, the matrix  ${}^B_A \mathbf{T}$  represents the position and orientation of B in the coordinate system {A}

$${}^B_A \mathbf{T} = {}^B_A \mathbf{trans}(B_x, B_y, B_z) {}^B_A \mathbf{R} = \begin{bmatrix} n_x & o_x & a_x & B_x \\ n_y & o_y & a_y & B_y \\ n_z & o_z & a_z & B_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.6)$$

It is proved that every posture matrix can be decomposed to many rotation matrices and translation matrices.

d. The spatial transformation between different coordinate systems by the posture matrix

If  ${}^A P = ({}^A P_x, {}^A P_y, {}^A P_z, 1)^T$  and  ${}^B P = ({}^B P_x, {}^B P_y, {}^B P_z, 1)^T$  are homogeneous coordinates of the point  $P$  in the coordinates {A} and {B}, respectively, then,  ${}^A P = {}^B_A \mathbf{T} {}^B P$ .

The transformation between different coordinate systems can be conducted on the basis of the above formula.

(2) The nature of matrices in D–H method

a. The nature of translation matrix

Proposition 1: if  $\{\mathbf{trans}(a, b, c) | \{a, b, c \in \mathbf{R}\}\}$  represents the exchange group on matrix multiplication, then,

$$\mathbf{trans}(0, 0, 0) = \mathbf{I} \quad (10.7)$$

$$\mathbf{trans}(a, b, c) \cdot \mathbf{trans}(a, b, c)^{-1} = \mathbf{trans}(-a, -b, -c). \quad (10.8)$$

Property 1:

$$\mathbf{trans}(a_1, b_1, c_1) \cdot \mathbf{trans}(a_2, b_2, c_2) = \mathbf{trans}(a_1 + a_2, b_1 + b_2, c_1 + c_2). \quad (10.9)$$

Corollary 1:

$$\mathbf{trans}(a, b, c)^{-1} = \mathbf{trans}(-a, -b, -c). \quad (10.10)$$

b. The nature of rotation matrix

Proposition 2: if  $\{\mathbf{Rot}(x, \theta) | \theta \in \mathbf{R}\}$  represents the exchange group on matrix multiplication, so does to  $\{\mathbf{Rot}(y, \theta) | \theta \in \mathbf{R}\}$  and  $\{\mathbf{Rot}(z, \theta) | \theta \in \mathbf{R}\}$ .

Property 2:

$$\begin{aligned} \mathbf{Rot}(x, \theta_1) \cdot \mathbf{Rot}(x, \theta_2) &= \mathbf{Rot}(x, \theta_1 + \theta_2) \\ \mathbf{Rot}(y, \theta_1) \cdot \mathbf{Rot}(y, \theta_2) &= \mathbf{Rot}(y, \theta_1 + \theta_2) \\ \mathbf{Rot}(z, \theta_1) \cdot \mathbf{Rot}(z, \theta_2) &= \mathbf{Rot}(z, \theta_1 + \theta_2). \end{aligned} \quad (10.11)$$

Corollary 2:

$$\begin{aligned} \mathbf{Rot}(x, \theta)^{-1} &= \mathbf{Rot}(x, \theta)^T = \mathbf{Rot}(x, -\theta) \\ \mathbf{Rot}(y, \theta)^{-1} &= \mathbf{Rot}(y, \theta)^T = \mathbf{Rot}(y, -\theta) \\ \mathbf{Rot}(z, \theta)^{-1} &= \mathbf{Rot}(z, \theta)^T = \mathbf{Rot}(z, -\theta). \end{aligned} \quad (10.12)$$

c. The nature of the posture matrix

If the posture matrix  ${}^m_i\mathbf{T}$  represents the transformation from coordinate system  $\{m\}$  to coordinate system  $\{i\}$ , and the posture matrix  ${}^k_m\mathbf{T}$  represents the conversion from coordinate system  $\{k\}$  to coordinate system  $\{m\}$ ; then, the posture matrix, from coordinate system  $\{k\}$  to coordinate system  $\{i\}$ , can be computed by the following formula:

$${}^k_i\mathbf{T} = {}^m_i\mathbf{T} {}^k_m\mathbf{T}. \quad (10.13)$$

Then, the transformation from  $i$ th joint-unit to  $j$ th joint-unit can be described as:

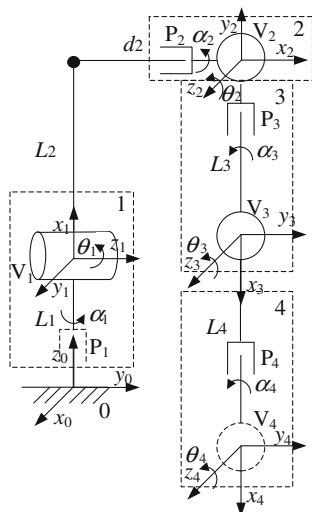
$${}^j_i\mathbf{T} = {}^{i+1}_i\mathbf{T} {}^{i+2}_{i+1}\mathbf{T} \cdots {}^j_{j-1}\mathbf{T}. \quad (10.14)$$

### 10.2.2 Structure of ULKC

The human ULKC contains root joint, chest joint, shoulder joint, elbow joint, wrist joint, and palm joints. Usually the palm is regarded as an independent research object as it has a lot of joints. Therefore, the wrist is the end-effector of the ULKC.

Human joints can be approximately treated as the rotary joint which is classified into P-type (shaft parallel to the link) and V-type (shaft perpendicular to the link). This study considers each joint-unit a combination of P-type and V-type. While the joint-unit only consists of P-type or V-type, this study adds another type which is

**Fig. 10.2** The joint-units of the ULKC



within the ZERO rotation range. The rotation shaft of the P-type joint is the  $x$ -axis of the local coordinate system of the joint-unit, and the rotation shaft of the V-type joint is the  $z$ -axis of the local coordinate system of the joint-unit, then the  $y$ -axis can be determined by the right-handed screw rule. According to this specification, the local coordinate systems of human ULKC's joint-units are established, as shown in Fig. 10.2.

The 0th joint-unit is the human root joint-unit whose coordinate is relatively fixed, and it is also the basic coordinate of the entire kinematics chain. The 1st joint-unit is the chest joint-unit which contains a real V-type joint and a complementary P-type joint; thus, it can only rotate around  $z_1$ . The 2nd joint-unit is the shoulder joint-unit, which consists of a V-type joint and a P-type joint. The 3rd joint-unit is the elbow joint-unit which contains a V-type joint and a P-type joint. The 4th joint-unit is the wrist joint-unit which consists of a complementary V-type joint and a real P-type joint. It should be noted that the joints  $P_2, V_2, P_3$  are derived from the shoulder joint which is a spherical joint. Placing  $P_3$  under the elbow unit is conducive to joint grouping and mathematical description of the kinematics chain.  $\alpha_i$  represents the rotation angle of  $P_i$ , and  $\theta_i$  the rotation angle of  $V_i$ .  $\alpha_1$  and  $\theta_4$  are always equal to 0.

According to the parameter values of adult body in GB10000-88 (Chinese), set values for  $L_1, L_2, L_3, L_4, d_2$  and the ranges of  $\theta_1, \theta_2, \theta_3, \alpha_2, \alpha_3, \alpha_4$ . They are listed in Table 10.1.

### 10.2.3 Posture Transformation of ULKC

With reference to Sects. 10.2.1 and 10.2.2, the transformation from first joint-unit to zeroth joint-unit of ULKC can be described as follows:

**Table 10.1** The parameters of the ULKC

Joint-unit	$L_i/\text{mm}$	$\theta_i/^\circ$	$\alpha_i/^\circ$	$d_i/\text{mm}$
1	19.5	0 ~ 60	0	0
2	31.5	-30 ~ 180	-180 ~ 45	19.85
3	33.3	-140 ~ 0	-45 ~ 90	0
4	25.3	0	-90 ~ 90	0

- (1) Translate  $L_1$  along the axis  $z_0$ , then the origin of the two coordinate systems are coincided;
- (2) Rotate  $-90^\circ$  around the axis  $x_0$ , then the shaft  $z_0$  and  $z_1$  are collinear;
- (3) Rotate  $-90^\circ$  around the axis  $z_0$ , then the shaft  $x_0$  and  $x_1$  are collinear.

Therefore, the transformation matrix is:

$${}^1_0\mathbf{T} = \mathbf{trans}(0, 0, L_1)\mathbf{Rot}(x, -90^\circ)\mathbf{Rot}(z, -90^\circ). \quad (10.15)$$

According to Eqs. (10.2–10.5),

$$\mathbf{trans}(0, 0, L_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10.16)$$

$$\mathbf{Rot}(x, -90^\circ) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10.17)$$

$$\mathbf{Rot}(z, -90^\circ) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.18)$$

Substitute Eqs. (10.16–10.18) into Eq. (10.15):

$${}^1_0\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.19)$$

Similarly, we can obtain the transformation matrices of other adjacent joints' local coordinate systems:



$${}^2_1\mathbf{T} = \begin{bmatrix} 0 & C\alpha_2 C\theta_1 - S\alpha_2 S\theta_1 - C\alpha_2 S\theta_1 - S\alpha_2 C\theta_1 & L_2 C\theta_1 \\ 0 & C\alpha_2 S\theta_1 + S\alpha_2 C\theta_1 & C\alpha_2 C\theta_1 - S\alpha_2 S\theta_1 & L_2 S\theta_1 \\ 1 & 0 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10.20)$$

$${}^3_2\mathbf{T} = \begin{bmatrix} C(\theta_2 - \pi/2) - S(\theta_2 - \pi/2)C\alpha_3 & S(\theta_2 - \pi/2)S\alpha_3 & L_3 C(\theta_2 - \pi/2) \\ S(\theta_2 - \pi/2) & C(\theta_2 - \pi/2)C\alpha_3 & -C(\theta_2 - \pi/2)S\alpha_3 & L_3 S(\theta_2 - \pi/2) \\ 0 & S\alpha_3 & C\alpha_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10.21)$$

$${}^4_3\mathbf{T} = \begin{bmatrix} C\theta_3 & -C\alpha_4 S\theta_3 & S\alpha_4 S\theta_3 & L_4 C\theta_3 \\ S\theta_3 & C\alpha_4 C\theta_3 & -S\alpha_4 C\theta_3 & L_4 S\theta_3 \\ 0 & S\alpha_4 & C\alpha_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.22)$$

In the equations above,  $S$  represents sine operator;  $C$  represents cosine operator.

According to Eq. (10.14), the posture transformation matrix of wrist joint (end-effector) relative to the root joint can be computed as follows:

$$\begin{aligned} {}^4_0\mathbf{T} &= {}^1_0\mathbf{T}_1 \mathbf{T}(\theta_1, \alpha_2) {}^3_2\mathbf{T}(\theta_2, \alpha_3) {}^4_3\mathbf{T}(\theta_3, \alpha_4) \\ &= \begin{bmatrix} {}^4_0n_x & {}^4_0o_x & {}^4_0a_x & {}^4_0B_x \\ {}^4_0n_y & {}^4_0o_y & {}^4_0a_y & {}^4_0B_y \\ {}^4_0n_z & {}^4_0o_z & {}^4_0a_z & {}^4_0B_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (10.23)$$

where,

$$\begin{aligned} {}^4_0n_x &= S(\theta_2 - \pi/2)C(\theta_3)(C\alpha_2 C\theta_1 + S\alpha_2 C\theta_1) - S\theta_3(S\alpha_3(S\alpha_2 S\theta_1 - C\alpha_2 C\theta_1) \\ &\quad - C(\theta_2 - \pi/2)C(\alpha_3)(C(\alpha_2)S(\theta_1) + S(\alpha_2)C\theta_1)) \end{aligned}$$

$${}^4_0n_y = C(\theta_2 - \pi/2)C\theta_3 - S(\theta_2 - \pi/2)C\alpha_3 S\theta_3$$

$$\begin{aligned} {}^4_0n_z &= -S\theta_3(S\alpha_3(C\alpha_2 S\theta_1 + S\alpha_2 C\theta_1) + C(\theta_2 - \pi/2)C\alpha_3(S\alpha_2 S\theta_1 - C\alpha_2 C\theta_1)) \\ &\quad - S(\theta_2 - \pi/2)C\theta_3(S\alpha_2 S\theta_1 - C\alpha_2 C\theta_1) \end{aligned}$$

$$\begin{aligned} {}^4_0o_x &= -S\alpha_4(C\alpha_3(S\alpha_2 S\theta_1 - C\alpha_2 C\theta_1) + C(\theta_2 - \pi/2)S\alpha_3(C\alpha_2 S\theta_1 + S\alpha_2 C\theta_1)) \\ &\quad - C\alpha_4 C\alpha_3(S\alpha_3(C\alpha_2 S\theta_1 - C\alpha_2 C\theta_1) - C(\theta_2 - \pi/2)C\alpha_3(C\alpha_2 S\theta_1 + S\alpha_2 C\theta_1)) \\ &\quad - S(\theta_2 - \pi/2)C\alpha_4 S\theta_3(C\alpha_2 S\theta_1 + S\alpha_2 C\theta_1) \end{aligned}$$

$${}^4_0o_y = S(\theta_2 - \pi/2)S\alpha_3 S\alpha_4 - C(\theta_2 - \pi/2)C\alpha_4 S\theta_3 - S(\theta_2 - \pi/2)C\alpha_3 C\alpha_4 C\theta_3$$

$${}^4_0o_z = S(\theta_2 - \pi/2)C\alpha_4 S\theta_3(S\alpha_2 S\theta_1 - C\alpha_2 C\theta_1) - C\alpha_4 C\theta_3(S\alpha_3(C\alpha_2 S\theta_1 + S\alpha_2 C\theta_1))$$

$$\begin{aligned}
& +C(\theta_2 - \pi/2)C\alpha_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1) - S\alpha_4(C\alpha_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1)) \\
& -C(\theta_2 - \pi/2)S\alpha_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1)) \\
{}^4_0a_x & = S\alpha_4C\theta_3(S\alpha_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1) - C(\theta_2 - \pi/2)C\alpha_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1)) \\
& -C\alpha_4(C\alpha_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1) + C(\theta_2 - \pi/2)S\alpha_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1)) \\
& +S(\theta_2 - \pi/2)S\alpha_4S\theta_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1) \\
{}^4_0a_y & = S(\theta_2 - \pi/2)C\alpha_4S\alpha_3 + C(\theta_2 - \pi/2)S\alpha_4S\theta_3 + S(\theta_2 - \pi/2)C\alpha_3S\alpha_4C\theta_3 \\
{}^4_0a_z & = S\alpha_4C\theta_3(S\alpha_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1) + C(\theta_2 - \pi/2)C\alpha_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1)) \\
& -C\alpha_4(C\alpha_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1) - C(\theta_2 - \pi/2)S\alpha_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1)) \\
& -S(\theta_2 - \pi/2)S\alpha_4S\theta_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1) \\
{}^4_0B_x & = L_2S\theta_1 + L_3S(\alpha_2 - \pi/2)(C\alpha_2S\theta_1 + S\alpha_2C\theta_1) - L_4S\theta_3(S\alpha_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1) \\
& -C(\theta_2 - \pi/2)C\alpha_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1)) + L_4S(\theta_2 - \pi/2)C\theta_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1) \\
{}^4_0B_y & = d_2 + L_3C(\theta_2 - \pi/2) + L_4C(\theta_2 - \pi/2)C\theta_3 - L_4S(\theta_2 - \pi/2)C\alpha_3S\theta_3 \\
{}^4_0B_z & = L_1 + L_2C\theta_1 - L_3S(\theta_2 - \pi/2)(S\alpha_2S\theta_1 - C\alpha_2C\theta_1) - L_4S\theta_3(S\alpha_3(C\alpha_2S\theta_1 + S\alpha_2C\theta_1) \\
& +C(\theta_2 - \pi/2)C\alpha_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1)) - L_4S(\theta_2 - \pi/2)C\theta_3(S\alpha_2S\theta_1 - C\alpha_2C\theta_1).
\end{aligned} \tag{10.24}$$

It is difficult to compute the inverse solutions from such a complex posture transformation matrix like Eq. (10.24) by using the analytical method or geometric method. Also, as there are six variables  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ,  $\alpha_2$ ,  $\alpha_3$ , and  $\alpha_4$  in the matrix and the end-effector of the ULKC is 6 DOF, it is also difficult to compute the precise solutions quickly by simply using the neural network algorithm or the standard genetic algorithm.

## 10.2.4 Summary

In this section, D–H method is used to describe the kinematics equations of ULKC. The basic principle of D–H method is that transformation of points in different coordinate systems can be conducted according to transitivity of transformation matrices. ULKC is a kinematics chain. Its IK problem is to resolve the joint angles when the posture of end-effector is known. We derive the transformation matrices of adjacent local coordinates systems and the kinematics Eq. (10.24) of ULKC. It is difficult to resolve Eq. (10.24) by basic geometric method or analytic method due to its complex and nonlinear features. From the next section, GA will be improved to resolve Eq. (10.24).

## 10.3 Improvement of Standard Generation Algorithm

In this section, an improved GA is put forward with the assistance of an adaptive genetic operator.

### 10.3.1 Introduction of SGA

Genetic algorithm is an efficient and global searching algorithm based on natural selection and genetic theory. The possible solution in the problem domain is seen as an individual or chromosome in the population. The SGA is a group operation algorithm which only uses the standard genetic operators, such as selection operator, crossover operator, and mutation operator. It is the foundation of other improved GA.

#### (1) The mathematical model of SGA

The SGA can be expressed as:

$$\text{SGA}=(C, E, P_0, M, \phi, \Gamma, \Psi, T)$$

where,

- $C$  coding method;
- $E$  the fitness function of individual;
- $P_0$  the initialization population;
- $M$  the amount of population;
- $O$  selection operator;
- $\Gamma$  crossover operator;
- $\Psi$  mutation operator;
- $T$  the termination condition;

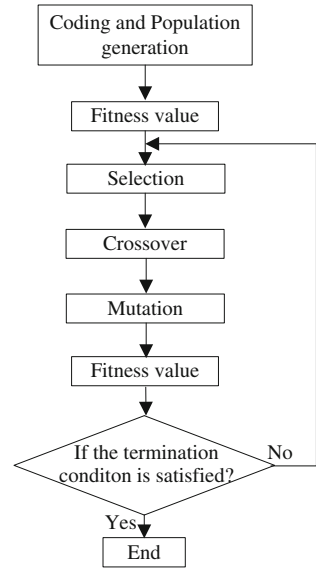
#### (2) The process of the SGA

##### a. Chromosome coding and decoding (Fig. 10.3)

In the SGA, the individuals in the population are represented by fixed-length binary strings, whose allele consists of the binary digit  $\{0, 1\}$ . The genes of each individual in the initial population can be generated via the random numbers distributed uniformly. For example,  $X=100010111011$  can be used to depict an individual, whose chromosome length is 16.

Coding: a parameter, with the range of  $[U1, U2]$ , can be expressed by the binary code symbol with the length of  $K$ , which generate  $2^k$  kinds of code. Thus the corresponding relationship between the parameters and coding is as follows:

**Fig. 10.3** The flowchart of SGA



$$\begin{aligned}
 000000 \dots 0000 &= 0 \rightarrow U_1 \\
 000000 \dots 0001 &= 1 \rightarrow U_1 + \delta \\
 000000 \dots 0002 &= 2 \rightarrow U_1 + 2\delta \\
 &\vdots \quad \quad \quad \vdots \\
 111111 \dots 1111 &= 2^k - 1 \rightarrow U_2
 \end{aligned} \tag{10.25}$$

where,  $\delta = \frac{U_2 - U_1}{2^k - 1}$ .

Decoding: if the code of an individual is  $b_k b_{k-1} b_{k-2} \dots b_2 b_1$ , the corresponding decoding formula is as follows:

$$X = U_1 + \left( \sum_{i=1}^k b_i \cdot 2^{i-1} \right) \cdot \frac{U_2 - U_1}{2^k - 1}. \tag{10.26}$$

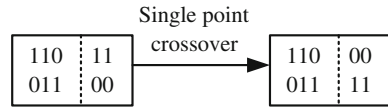
**b. The evaluation of individuals' fitness**

In the SGA, the probability of individual's genetic opportunity is decided by its fitness value. All the individual fitness values must be nonnegative for evaluating the genetic probability correctly. So, we need to determine the transformation principle from target function to fitness function, especially when the target value is negative.

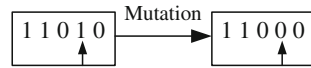
**c. Genetic operators**

Three kinds of genetic operators used by the SGA are as follows:

**Fig. 10.4** The process of crossover



**Fig. 10.5** The process of mutation



1. Use the proportional selection operator to select the father individuals. The proportional selection factor refers to the genetic probability. If the number of individuals is  $M$ , the fitness of  $i$ th individual is  $f_i$ , then, the probability of  $i$ th individual is:

$$P_i = f_i / \sum_{k=1}^M f_k. \tag{10.27}$$

If the probability of individual selection is given, uniform random numbers between  $[0, 1]$  are generated to determine which individual is to mate. The individuals, who have a large selection probability, can be repeatedly selected and their genes are expanded in the population, while the small ones would eliminate gradually.

2. The single-point crossover operator is applied to crossover operation in SGA. Only one cross point exists in the single-point crossover operator, the two individuals are chosen as cross objects, then, a crossover point is generated randomly. The chosen individuals exchange their genes at the cross point and generate two children individuals (Fig. 10.4).

3. The mutation operation uses the basic bit mutation operator and the uniform mutation operator. In order to avoid the “premature” convergence problem, there is a small probability for the individuals composed of binary gene code to shift the genetic code from 0 to 1 or 1 to 0 (Fig. 10.5).

4. The parameters of the SGA

There are four parameters of the SGA need to be set; they are  $M$ ,  $T$ ,  $P_c$ , and  $P_m$ .

$M$  represents the size of the population, which refers to the individual number of the population, usually ranging from 20 to 100;

$T$  represents the terminal generation times, which usually ranges from 100 to 500;

$P_c$  stands for the crossover probability, usually 0.4–0.99;

$P_m$  stands for the mutation probability, normally 0.0001–0.1.

### 10.3.2 Principles of Improvement

Since the intact structure and theory of GA was systemically proposed by Holland in 1975, numerous scholars have devoted to developing GA. The coding method, the control parameters, and crossover mechanism are deeply studied. Therefore, various

improved GA are proposed, such as Hierarchic Genetic Algorithm [18, 19], CHC Algorithm [20, 21], Messy Genetic Algorithm [22, 23], Niche Genetic Algorithm [24], Adaptive Genetic Algorithm [25, 26], and Parallel Genetic Algorithm [27–29], whose characteristics and performances are compared in article [30].

The present research focuses on the improvement of GA from the perspectives of population initialization and its diversity. An improved GA is put forward in this chapter with the assistance of adaptive genetic operator.

We operate on the individuals from the perspectives of selection, crossover, and mutation. The SGA contains some demerits such as “premature” and inefficient searching in later stage, which result from the following aspects, such as single population, random selection of initial population, and no-adaptive crossover and mutation probabilities. This research will improve the SGA on the aspects of population diversity, the selection strategy for initial population, and the adaptive genetic operators to overcome the “premature” and inefficient searching in later stage.

- a. Population diversity. In the SGA, all the  $N$  individuals in a population, **Chrom**, the differences among individuals will be reduced with the increase of generation times, which leads the searching process to the “premature” convergence. We introduce the population migration: the population **Chrom** is divided into  $p$  independent subpopulations, **SubPop** <sub>$i$</sub>  ( $i = 1, 2, \dots, p$ ); then each sub population has  $m = N/p$  individuals. The genetic operations are conducted in the isolated subpopulation in the normal condition; we let the individuals migrate to other subpopulations with the probability  $P_{\text{mig}}$ , while it meets the needs of certain conditions (such as get the certain generation times). It can ensure the diversity of individuals in population, and avoid the “premature” convergence.
- b. Population initialization. Randomly selecting the individuals will reduce the differences among individuals significantly after several genetic operations, which results in inefficient searching in later stage. To solve this problem, Rasit [17] proposes a method in which the initial population is generated by three trained Elman neural networks. Three outputs from three networks can be obtained at a time. After putting these outputs into the initial population, the GA is used to optimize the solution. The disadvantages of this method are: complex structure, which consists of 155 neurons in the hidden layer of the three Elman neural networks; the small number of sample (no more than 3); other individuals are randomly generated. Based on the research of the phenomenon of human populations, we found there is at least a significant difference in the human populations—color. The populations with different colors certainly have different chromosomes, and the colors of individuals within the same population are similar but different; on the other hand, there are no exactly same individuals in the same human population. These ensure the diversity of human genes, thus we propose three principles for initial population:
  1. There is at least one gene different between subpopulations;
  2. There is at least one gene similar (relative) between individuals within the same subpopulation;
  3. There are no two same individuals within the same subpopulation.

These three principles can be summarized as “the individuals are similar but different within the same sub population; the individuals are different but similar between sub populations.”

- c. The adaptive genetic operators. In order to improve the capability of GA to avoid the “premature” convergence and inefficient searching in later stage, the genetic operators should be adaptive. That means the crossover probability  $P_c$  and mutation probability  $P_m$  would be increased with the similarity of individuals increased. We design the adaptive operators as follows:

$$P_c = P_{c0} + \mu\beta_i \quad (10.28)$$

$$P_m = P_{m0} + \nu\beta_i \quad (10.29)$$

where,  $P_{c0}$  is the initial crossover probability and  $P_{m0}$  is the initial mutation probability;  $\beta_i$  is the proportion of the maximum fitness individuals account to the total number of individuals in  $i$ th generation;  $\mu$  and  $\nu$  are the two constant factors which are used to make  $P_c$  and  $P_m$  not more than 1.

The improved GA has the features of diversity population, the similar but different initial population, the adaptive genetic operators, which will be helpful to search the high-accuracy solutions quickly.

### 10.3.3 Process Flow of IGA

The process flow of the GA contains coding, population initialization, fitness value evaluation, selection, crossover, mutation, and judge the terminal condition. According to the above principles, we design the IGA as the following steps:

- a. Coding. The fixed-length binary string is used to represent the individual in the SGA, where allele is  $\{0, 1\}$ . Binary coding will greatly increase gene length, and must shift between the coding and decoding repeatedly. It is not suitable for the high dimension. We use the real coding to represent each individual whose chromosome concludes six genes  $\{\theta_1, \theta_2, \theta_3, \alpha_2, \alpha_3, \alpha_4\}$ . The variable domains are shown in Table 10.1.
- b. Population initialization. The population initialization contains population diversity and individuals' initialization. The population diversity is realized by setting subpopulations. We divide the population **Chrom** into  $p$  independent subpopulations **SubPop <sub>$i$</sub>**  ( $i = 1, 2, \dots, p$ ), each subpopulation has  $m$  individuals, then the entire population has  $N = pm$  individuals. The genetic operations are independently conducted within the subpopulations. We let the individuals of every subpopulation migrate to other subpopulations with probability  $P_{\text{mig}}$  every  $g$  generations, which will ensure the individual diversity within populations.

According to the principle “the individuals are similar but different within the same sub population, the individuals are different but similar between sub popula-

tions,” we initiate the population: using  $\theta_1$  as feature gene to distinguish the different subpopulations; we divide the interval  $(0, \pi/3)$  of variable domain of  $\theta_1$  into  $p$  equal length intervals, and the length of each interval is:

$$\sigma = \frac{\pi/3}{p}. \quad (10.30)$$

Then, the  $i$ th interval is  $((i - 1)\sigma, i\sigma)$  from which the first gene  $\theta_1$  of the  $i$ th subpopulation are generated, and the other five genes are generated from their own variable domain randomly. This method can ensure: a. the first genes of individuals from the same subpopulation are similar but different; b. the other five genes are all the same, which is a small probability event; c. the individuals from different subpopulations are not possibly the same.

- c. Fitness value evaluation. The only requirement for the fitness function is that, for each chromosome, we can calculate a corresponding comparable nonnegative result. According to the posture matrix  ${}^4_0\mathbf{T}'$  computed by individual and the target posture matrix  ${}^4_0\mathbf{T}$  except the last line (the last line of the all posture matrix is the same), we define the error function:

$$F(\theta_1, \theta_2, \theta_3, \alpha_1, \alpha_2, \alpha_3) = \frac{1}{3 \times 4} \sqrt{\sum_{i=1}^3 \sum_{j=1}^4 ({}^4_0\mathbf{T}'_{i,j} - {}^4_0\mathbf{T}_{i,j})^2}. \quad (10.31)$$

The fitness function is designed as follows:

$$\text{Fit}(F) = \frac{1}{1 + F(\theta_1, \theta_2, \theta_3, \alpha_1, \alpha_2, \alpha_3)}. \quad (10.32)$$

The individual's fitness value of  $\text{Fit}(F)$  will be larger when the error  $F(\theta_1, \theta_2, \theta_3, \alpha_1, \alpha_2, \alpha_3)$  is smaller.

- d. Selecting the father individuals. The randomized competitive selection method is used to select the father individuals. Each time, we select a pair of individuals by using roulette selection mechanism, and then make the two individuals compete. The one who has the higher fitness will be selected. Repeat this process until the population is full.
- e. Crossover. We use the two-point crossover method which needs to randomly select two integers  $k_1$  and  $k_2$ , and exchange the  $(k_1 + 1)$ th to  $(k_2)$ th genes of two father individuals with probability  $P_c$ ; the value of  $P_c$  is derived by Eq. (10.28).
- f. Mutation. Mutation means that one or several genes of each individual are changed with mutation probability  $P_m$ ; the  $P_m$  is derived by Eq. (10.29).
- g. Termination conditions. When the error computed by Eq. (10.31) is equal to or smaller than the given value  $\varepsilon$ , or the GA operation goes to the maximum generation times, the computational process comes to the end.



### 10.3.4 Summary

In this section, the principle, “the individuals are similar but different within the same sub population; the individuals are different but similar between sub populations,” is proposed to direct the initialization and diversity of the population. The population is divided into some independent subpopulations which are marked by the first gene. Thus, the chromosomes of individuals from different subpopulations are absolutely different, and the chromosomes of individuals from same subpopulations are similar. And the migration strategy is also used to keep diversity. Formulas (10.28) and (10.29) are designed to make the crossover operator and mutation operator adaptive. These methods can keep the IGA diversity and avoid “premature” convergence and inefficient searching in later stage. Besides, error function (10.31) and fitness function (10.32) are designed.

## 10.4 Simulation Studies

In this section, SGA and IGA are compared in terms of optimizing capability in solving the IK problem of ULKC. Then, the IGA is applied in the control of ULKC of a virtual human.

### 10.4.1 Simulation of SGA

We set 120 individuals in the population; crossover probability  $P_c = 0.7$ , mutation probability  $P_m = 0.131$ ; the termination condition was the error  $F \leq 0.01$  or the number of generation time was up to 200.

For example, a target posture matrix was generated randomly. See the following matrix:

$${}^4_0\mathbf{T} = \begin{bmatrix} 0.87814 & -0.02665 & 0.47767 & 46.3554 \\ 0.07128 & 0.99459 & -0.07554 & 54.6941 \\ -0.47307 & 0.10039 & 0.87529 & 33.5743 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.33)$$

The result of the SGA operation is shown in Fig. 10.6.

The solid line represents the error of the optimal individual in the population, and the dotted line represents the average error of all the individuals. The computation fell into “premature” convergence when the 31st generations, and the error was 0.0329 when generation time was 200, which did not reach the target value 0.01.

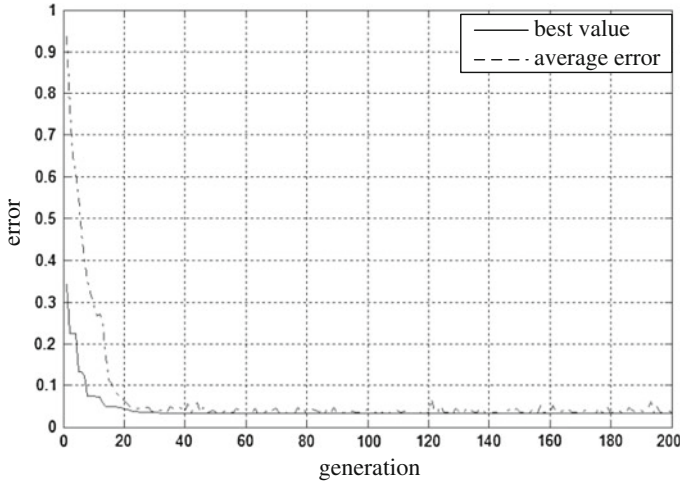


Fig. 10.6 Computational curves of SGA

### 10.4.2 Simulation of IGA

We divided the population into three subpopulations, each with 40 individuals;  
 We let the individuals migrate to other subpopulations with the probability of 0.2 every 20 generation times;  
 Crossover probability and mutation probability are:

$$P_{ci} = 0.6 + 5\beta_i \tag{10.34}$$

$$P_m = 0.125 + 2\beta_i \tag{10.35}$$

The termination condition was the error  $F \leq 0.01$ ;  
 The target posture matrix was expressed as matrix (10.33).  
 The result of the IGA operation was shown in Fig. 10.7.  
 The error was 0.0094 when the generation time was 30, which reached the target value 0.01, and then the computation came to an end. The best individual was:  
 $\text{Chr\_best} = \{34.380 \ 79.038 - 76.716 - 124.272 \ 29.286 - 0.738\}$   
 According to Chr\_best, we obtained the actual posture matrix:

$${}^4_0T' = \begin{bmatrix} 0.87611 & -0.01764 & 0.48180 & 46.2860 \\ 0.06416 & 0.99471 & -0.08025 & 54.1649 \\ -0.47783 & 0.10122 & 0.87260 & 33.3975 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10.36}$$

Comparing the  ${}^4_0T'$  and  ${}^4_0T$ , we saw that the solution was highly accurate.

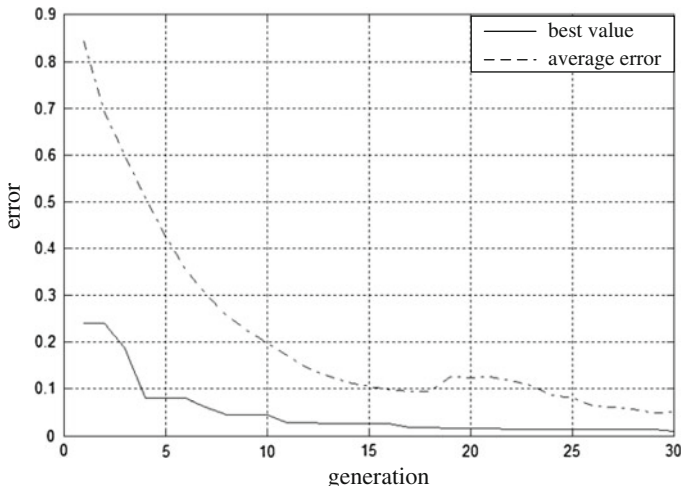


Fig. 10.7 Computational curves of IGA

Table 10.2 Results of simulation

No.	The generation times when error is up to 0.01	
	IGA	SGA
1.	38	41
2.	28	–
3.	43	–
4.	21	25
5.	25	40
6.	30	55
7.	35	–
8.	32	32
9.	59	–
10.	31	86

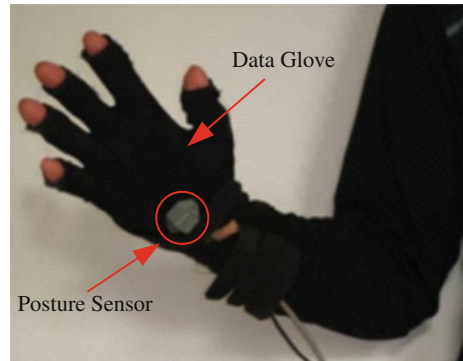
Note The line means the error has not reached 0.01 after 200 generation times

### 10.4.3 Comparison of SGA and IGA

We compared the performance differences of SGA and IGA by 10 times simulation to resolve the ULKC IK problem. The results are shown in Table 10.2.

By inspection of Table 10.2, it can be seen that all 10 computations based on IGA meet the target error 0.01 within 200 generations. And SGA cannot meet the target error after 200 generations for four times. The results showed that compared with SGA, IGA falls into the “premature” convergence with low probability.

**Fig. 10.8** Hardware of the system



#### 10.4.4 Application

We applied the IGA in a virtual maintenance system. Posture sensor was used to measure the wrist joint's posture of operator relative to its root joint, and the position of operator's root was fixed. The postures of root and sensor of operator are mapped into the postures of root joint and end-effector of virtual human in the virtual circumstance. The data glove was used to control the virtual human's behaviors like grasp and release. The hardware is shown in Fig. 10.8.

The termination condition is  $F < 0.05$  or the generation is 100 times. We set the scan period 100 ms.

The operator stretched or shrank his hand in the real world. The posture sensor measured the posture of wrist joint relative to the root; thus the posture of end-effector relative to the root joint of virtual human was obtained. Compute the inverse solution with IGA and get the six joint angles. Then the form of virtual human's ULKC was displayed on the screen. When the hand of virtual human touched the part, the data glove was used to control the hand and catch it. The operator and virtual maintenance scene are shown in Fig. 10.9.

The computer measurements indicated that it cost 0.068 s to compute one genetic operation with Intel Core i3 (clocked at 2.93 GHz) processor. And in most cases, the IGA approach can obtain the required solution within 20 generations. The experimental results showed that the IGA approach was reliable in application.

#### 10.4.5 Summary

In this section, SGA and IGA have been tested to resolve the IK problem of ULKC. The same target posture matrix has been resolved by both SGA and IGA which consists of the same number of individuals. The simulation curves are shown in Figs. 10.6 and 10.7, respectively. Obviously, the simulation result of IGA is better



**Fig. 10.9** Operator and virtual human

than that of SGA. A number of simulations have been conducted for the comparison, which is shown in Table 10.2. It can be concluded that the optimizing capability of IGA is much better than that of SGA. The IGA has been tested to control the ULKC of virtual human and to provide demonstrations for the other potential applications.

## 10.5 Conclusions

In this research, we have proposed a new IGA approach to resolve IK problem and obtained the precise inverse solution of ULKC. Compared with SGA, the IGA can avoid “premature” convergence and inefficient searching occurred with much high probability in the later stage of the computation. We have utilized D–H method to structure the mathematic model of ULKC, and improved the GA’s population diversity, population initialization, and adaptive genetic operators. Therefore the proposed procedure provides high precise solutions in a quick search route. The simulation bench tests have demonstrated the concepts and algorithms. In addition, the IGA can be applied to resolve many other chain’s inverse kinematic problems.

**Acknowledgments** This work is partially supported by the National Nature Science Foundation of China under Grant 61102170 and 61273188, and the National Advanced Research Foundation of China under Grant 9140A27040112JB47081.

## References

1. Wang X, Sun S-Q, Chai C-L (2009) An overview of 3D human motion editing and synthesis. *J Image Graph* 14(12):233–242
2. Liu G-D, Pan Z-G, Cheng X, Li L, Zhang M-M (2010) A survey on machine learning in the synthesis of human motions. *J Comput Aided Des Comput Graph* 22(9):1619–1627

3. Xia S-H, Wei Y, Wang Z-Q (2010) A survey of physics-based human motion simulation. *J Comput Res Devel* 47(8):1354–1361
4. Chen P, Liu L, Yu F (2012) A geometrical method for inverse kinematics of a kind of humanoid manipulator. *Robot* 34(2):211–216
5. Wang H, Cai Y-F, Zhang W-G (2011) Analytical algorithm of inverse kinematics model for 7DOF manipulator. *J JingSu Univ (Nat Sci Ed)* 32(3):254–259
6. Zhang X, Wang Z-Y, Wang Z-S (2009) A real-time inverse kinematics algorithm for human motion modeling. *J Comput Aided Des Comput Graph* 21(6):853–860
7. Qu S-C, Liu X-B, Wang Q-Q (2013) A brief fuzzy controller for an intelligent tracking system. *Int J Model Ident Control* 19(2):171–178
8. Zhang B-G, Zhang R-Z, Wang G (2012) Breakout prediction for continuous casting using genetic algorithm-based back propagation neural network model. *Int J Model Ident Control* 16(3):199–205
9. Golea N, Debbache G, Golea A (2012) Neural network-based adaptive sliding mode control for uncertain non-linear MIMO systems. *Int J Model Ident Control* 16(4):334–344
10. Satish K, Kashif I (2012) Implementation of artificial neural network applied for the solution of inverse kinematics of 2-link serial chain manipulator. *Int J Eng Sci Technol* 4(9):4012–4024
11. Bassam D, Shadi K, Mohamed A (2010) Applying neural network architecture for inverse kinematics problem in robotics. *Softw Eng Appl* 3:230–239
12. Santosh KN, Swetalina P, Subudhi PRS (2012) A novel application of artificial neural network for the solution of inverse kinematics controls of robotic manipulators. *Intell Syst Appl* 9:81–91
13. Saleh T, Christopher C, William M (2006) A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering. *IEEE Proc Evol Comput* 1815–1822
14. Banga VK, Singh Y, Kumar R (2007) Simulation of robotic arm using genetic algorithm and AHP. *World Acad Sci Eng Technol* 5:95–101
15. Liu D-H, Yuan S-C, Wang J-Y (2008) Neural networks based on the genetic algorithm and its application in mechanical engineering. *J XiDian Univ* 35(1):152–156
16. Carlos K, Maria LC (2012) Robot arm fuzzy control by a neuro-genetic algorithm. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.135.4167>
17. Rasit K (2011) A neuro-genetic approach to the inverse kinematics solution of robotic manipulators. *Sci Res Essays* 6(13):2784–2794
18. Wang J, Li J-H, Ni N (2011) Restricted searching area hierarchical genetic algorithm for UAV path planning. *J Detect Control* 33(4):39–43
19. Cheng W, Jin Q-R (2013) Optimum base station frequency allocation based on hierarchical genetic algorithms. *Comput Digital Eng* 41(2):168–170
20. Qi Y, Qin H-L, Shen S-T, Li Y-H (2004) A study of the optimization for fuzzy diagnostic rules based on the reformative CHC algorithm. *Acta Aeronautica Et Astronautica Sinica* 25(04):362–367
21. Huang W-P, Wang X-Y (2007) Structural damage diagnose is based on improved CHC algorithm. *J Vib Measur Diagn* 27(3):232–235
22. Goldberg DE, Korb B, Deb K (1989) Messy genetic algorithms: motivation, analysis and first results. *Complex Syst* 3:493–530
23. Zhang Z-Y, Xie G, Xie K-M (2005) Definition of node numbers of hidden layer of feed-forward neural network by messy genetic algorithm. *J TaiYuan Univ Technol* 36(4):392–394
24. Saleh T, Christopher C, William M (2006) A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering. *IEEE Congr Evol Comput Sheraton Vancouver Wall Centre Hotel* 1815–1822
25. Lv X-Q, Chen S-G, Lin J (2013) Adaptive genetic annealing algorithm of solving 0/1 knapsack. *J Chongqing Univ Posts Telecommun (Nat Sci Ed)* 25(1):138–142
26. Chen K, Hu X-G (2013) Method of relay routing based on genetic adaptive ant colony system algorithm. *J Cent South Univ (Sci Technol)* 44(2):572–579

27. Topping B-H, Sziveri J, Bahreinejad A (1998) Parallel processing, neural networks and genetic algorithms. *Adv Eng Softw* 29(10):763–786
28. Matsumura T, Nakamura M, Okech J (1998) A parallel and distributed genetic algorithm on loosely-coupled multiprocessor system. *IEICE Trans Fundam Electron Commun Comput Sci* 81(4): 540–546
29. Mayer MK (1999) A network parallel genetic algorithm for the one machine sequencing problem. *Comput Math Appl* 37(3):71–78
30. Liu G, Cao Y (2007) Performance comparison of several improved genetic algorithm. *Microcomput Inf* 30:190–192