System Identification and Control Using DNA Computing Algorithms

Ching-Huei Huang, Chun-Liang Lin and Horn-Yong Jan

***Abstract*—**A DNA computing algorithm (DNACA) with an electron-ion interaction potential (EIIP) decoding scheme is proposed to identify a class of transfer functions. The DNACA includes crossover, mutation, enzyme and virus operators providing a highly modular, flexible, and accurate self-organizing structure. Simulation study based on the De Jong's test functions show its superior performance when compared with the improved and standard genetic algorithms (GAs). The algorithm is also applied to control design with the simplest controller through special frameshift mutation such as enzyme and virus.

***Keywords*—**DNA computing algorithm, Electron-ion interaction potential, Systems identification

I. Introduction

RECENTLY developed DNA computing algorithms (DNACAs) have inspired new methods that can simultaneously solve the parameter and structure optimization problems. The DNACAs based on the concept of bimolecular evolution was first developed by Adleman in [1]. Maley further detailed this kind of algorithms in terms of chemical processes and computer programming [2].

The operational features inherent in DNACAs make the algorithms implementable in the future DNA computers which are over a billion times possible to implement more computationally efficient than the conventional computers. The massively parallel nature of DNA in those computers means that computation may be millions or billions of times beyond today's supercomputers [3-7].

In this paper, accuracy of the proposed DNACA with or without frameshift operators is verified first by numerical tests with the electron-ion interaction potential (EIIP) [8-10] decoding scheme.

Two of the competitive applications of DNACAs are on the system identification (ID) and control design. System ID is essential for control designs based on the process model, which go from PID controller [11, 12] to more sophisticated methods such as H_2 control [13] and is applicable to the robust control design [14-16]. With regard to system ID, DNA strings are created here to represent transfer function models in which codons are used to represent coefficients of the denominator and numerator polynomials. Using frameshift operators (enzyme and virus), the order and numeric values of the transfer function model can be refined to fit the objective of finding the simplest ID model. The proposed approach is verified by testing a group of the De Jong's functions given in [17]. Verification of the simulated results shows that the proposed method performed well, even for the system models with wide dynamic responses. Application to robust control design is also investigated and simulated result is presented.

Manuscript received January 31, 2009. This research was sponsored by National Science Council, Taiwan, R.O.C. under the Grant NSC 96-2628-E-005-086-MY2.

C. H. Huang is with the Department of Electrical Engineering, National Chung Hsing University, Taichung, 402 Taiwan, R.O.C. (e-mail: d9564201@mail.nchu.edu.tw).

C. L. Lin is with the Department of Electrical Engineering, National Chung Hsing University, Taichung, 402 Taiwan, R.O.C. (phone: +886-4-22851549 ext 708; fax: +886-4-22851410; e-mail: chunlin@dragon.nchu.edu.tw).

H. Y Jan was with the Institute of Electrical and Communications Engineering, Feng Chia University, Taichung, 407 Taiwan, R.O.C.

II. Biological Computation Algorithms

A DNA strand comprises of two complimentary strings in which four nucleotide base: adenine (A), cytosine (C), guanine (G), and thymine (T)—are arranged in various combinations. Nucleotides are paired along with two strings. One DNA string can be separated from the other through chemical process. As the other string is a perfect compliment, for the purpose of computation simplicity only one string is needed to further operations in next generation.

The biological computation algorithm calculates the fitness function and gets the best solution during evolution of generations in terms of DNA coding schemes. The property of electro-ion interaction potential (EIIP) decoding can be applied in biological computation process and designed to calculate the fitness values. The system computation processes are as follows.

A. The DNA Computation Algorithms

The design flow of the DNACA and decoding process with EIIP as shown in Fig.1 is described in the follows.

Step 1: Randomly generated a group of DNA sequences.

Step 2: Perform DNA coding scheme for each DNA sequence.

Step 3: The DNA codons within the DNA sequences are translated to amino acids.

Step 4: The amino acids are converted to the EIIP levels.

Step 5: Perform DNA computing process with crossover, mutation, enzyme and virus operations.

Step 6: Calculate the fitness value (the fitness function may incorporate with a penalty term).

Step 7: Store the results.

(Sorting and saving the best solution decoded with EIIP in the DNACA).

Step 8: Repeat.

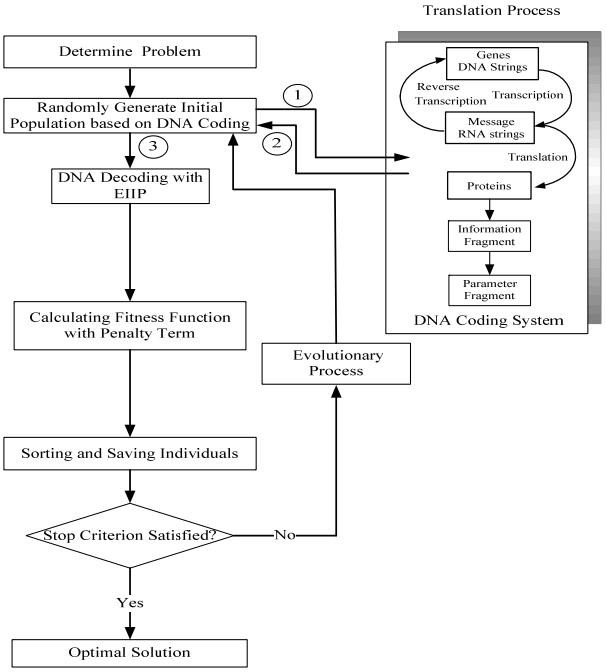


Fig. 1 Flowchart of DNACA for the solution search.

B. DNA-Coding scheme

Biologists have discovered 20 amino acids encoded by 64 codons, which are expressed in the versatile combinations of {A, C, T, G}. As in biological DNA coding scheme, a gene string for DNACAs starts with the codon ATG and ends with the codon TAA, TAG or TGA.

Electro-ion interaction potential (EIIP) applied here generates initial population and its subsequent offsprings. In the method presented, each amino acid was represented by a specified number, which was referred to as the unique electron-ion interaction potential and it was irrespective of its position in a DNA string in [18]. Moreover, these numbers are essential to build a physical and mathematical model which interprets protein sequences information using signal analysis methods.

Basically, the representation possesses three merits in biology: (i) measure the chemical properties of bases, (ii) preserve information about the properties of the bases, and (iii) directly relate to DNA chromosomes. The EIIP describes the average energy states of all valence electrons, especially amino acids; its value for 20 amino acids and five types EIIP wheel nucleotides are summarized in Table I and Fig. 2.

Table I. Translation of DNA strands code, amino acid codon and EIIP value.

Code of amino acid	Three-letter code	One-letter code	Amino acid	Electron-ion interaction potential values
GCT GCC GCA GCG	ALA Ala (12)	A	Alanine	【2】0.0373
CGT CGC CGA CGG AGA AGG	ARG Arg (11)	R	Arginine	【4】0.0959
GAT GAC	ASP Asp (17)	D	Aspartic	【4】0.1263
AAT AAC	ASN Asn (15)	N	Asparagine	【0】0.0036
TGT TGC	CYS Cys (19)	C	Cysteine	【3】0.0829
GAA GAG	GLU Glu (18)	E	Glutamic	【1】0.0058
CAA CAG	GLN Gln (14)	Q	Glutamine	【3】0.0761
GGT GGC GGA GGG	GLY Gly (13)	G	Glycine	【0】0.0050
CAT CAC	HIS His (5)	H	Histidine	【1】0.0242
ATT ATC ATA	ILE Ile (2)	I	Isoleucine	【0】0.0000
TTA TTG CTT CTC CTA CTA	LEU Leu (9)	L	Leucine	【0】0.0000
AAA AAG	LYS Lys (16)	K	Lysine	【2】0.0371
ATG	MET Met (3)	M	Mothionine	【3】0.0823
TTT TTC	PHE Phe (1)	F	Phenylalaine	【4】0.0946
CCT CCC CCA CCG	PRO Pro (7)	P	Proline	【1】0.0198
TCT TCC TCA TCG AGT AGC	SER Ser (10)	S	Serine	【3】0.0829
ACT ACC ACA ACG	THR Thr (8)	T	Threonine	【4】0.0941
TGG	TRP Trp (20)	W	Tryptophan	【2】0.0548
TAT TAC	TYR Tyr (4)	Y	Tyrosine	【2】0.0516
GTT GTC GTA GTG	VAL Val (6)	V	Valine	【1】0.0057

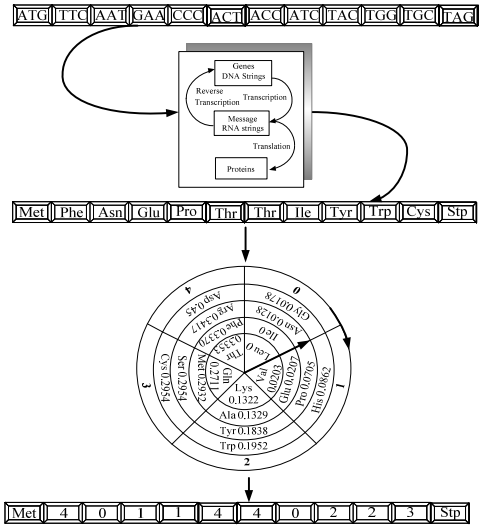


Fig. 2 Normalized EIIP wheel of amino acids.

First, consider a transfer function consisted of a gain, a numerator polynomial and a denominator polynomial. Each DNA string representing the transfer function is encoded as a vector in the following form:

$$\vec{S} = [\vec{S}_1 \quad \vec{S}_2 \quad \cdots \quad \vec{S}_k] \tag{1}$$

Individual DNA strings $\vec{S}_{\mathrm{IND}}$, IND = 1,…, k, are defined as

$$\vec{S}_{\mathrm{IND}} = [\vec{S}_{\mathrm{INF}} \quad \vec{S}_{\mathrm{PARA}}],\ \mathrm{IND}=1,\ldots,k,$$

$$S_{\mathrm{STA}} \triangleq ATG,\ S_{\mathrm{STP}} \triangleq \{TAG, TGA, TAA\},$$

$$\vec{S}_{\mathrm{INF}} = [\vec{S}_{\mathrm{INF\_G}} \quad \vec{S}_{\mathrm{INF\_D}} \quad \vec{S}_{\mathrm{INF\_N}}],$$

$$\vec{S}_{\mathrm{INF\_G}} = [S_{STA} \quad S_{\mathrm{INF\_G(1)}} \quad S_{\mathrm{STP}} \quad S_{\mathrm{STP\_INF}}],$$

$$\vec{S}_{\mathrm{INF\_D}} = [S_{\mathrm{STA}} \quad S_{\mathrm{INF\_D(1)}} \quad \cdots \quad S_{\mathrm{INF\_D(7)}} \quad S_{\mathrm{STP}} \quad S_{\mathrm{STP\_INF}}],$$

$\vec{S}_{\text{INF\_N}} = [S_{\text{STA}} \quad S_{\text{INF\_N(1)}} \quad \cdots \quad S_{\text{INF\_N(7)}} \quad S_{\text{STP}} \quad S_{\text{STP\_INF}}]$,

$S_{\text{STP\_INF}} \triangleq TGATGA$,

$\vec{S}_{\text{PARA}} = [S_{\text{PARA\_G}} \quad \vec{S}_{\text{PARA\_D}} \quad \vec{S}_{\text{PARA\_N}}]$,

$S_{\text{PARA\_G}} = [S_{\text{STA}} S_{\vec{C}_C} S_{\text{STP}} S_{\text{STP\_G}}]$,

$\vec{S}_{\text{PARA\_D}} = [\vec{S}_{\text{PARA\_D}_1} \quad \vec{S}_{\text{PARA\_D}_3}]$

$= [S_{\text{STA}} S_{\vec{C}_{N1\_1}} S_{\text{STP}} S_{\text{STA}} S_{\vec{C}_{N1\_2}} \cdots S_{\text{STA}} S_{\vec{C}_{N1\_m_1}} S_{\text{STP}} S_{\text{STA}} S_{\vec{C}_{N2\_1}} S_{\text{STP}} S_{\text{STA}} S_{\vec{C}_{N3\_1}} \cdots S_{\vec{C}_{N3\_m_3}} S_{\text{STP}} S_{\text{STP\_N}}]$,

$\vec{S}_{\text{PARA\_N}} = [\vec{S}_{\text{PARA\_N}_1} \quad \vec{S}_{\text{PARA\_N}_3}]$

$= [S_{\text{STA}} S_{\vec{C}_{D1\_1}} S_{\text{STP}} S_{\text{STA}} S_{\vec{C}_{D1\_2}} \cdots S_{\text{STA}} S_{\vec{C}_{D1\_m_1}} S_{\text{STP}} S_{\text{STA}} S_{\vec{C}_{D2\_1}} S_{\text{STP}} S_{\text{STA}} S_{\vec{C}_{D3\_1}} \cdots S_{\vec{C}_{D3\_m_3}} S_{\text{STP}} S_{\text{STP\_D}}]$,

$S_{\text{STP\_G}} \triangleq TGA$, $S_{\text{STP\_D}} \triangleq TAA$, $S_{\text{STP\_N}} \triangleq TAG$

where $\vec{S}_{\text{IND}}$ consists of the structure information $\vec{S}_{\text{INF}}$ and the parameter information $\vec{S}_{\text{PARA}}$; the structure information $\vec{S}_{\text{INF}}$ represents the coding information for the gain ($\vec{S}_{\text{PARA\_G}}$), denominator polynomial ($\vec{S}_{\text{PARA\_D}}$) and numerator polynomial ($\vec{S}_{\text{PARA\_N}}$) based on the combination of different numbers of amino acids; S_{STA} and $S_{\text{STP\_}S_s}$ with $S_s \triangleq$ INF, D, N are used to identify lead and end codons. Subscripts, f and s, are defined as the polynomials with the first or second-order polynomial.

Before executing the evolutionary process, the lead codon and the combination of end codons corresponding to each DNA string have to be excluded while single or multiple active points are randomly assigned for an operation.

Normalization is performed based on the categorization of EIIP. Accordingly, the decoded parameters are calculated, excluding lead and end codon, as follows

$$\frac{r\sum_{i=1}^{L} DNA_{AA\_EIIP}^{(i)} \cdot 5^{i-1} - M}{N} \tag{2}$$

where the constants r, M and N control the range and resolution of the parameters, $DNA_{AA\_EIIP}^{(i)}$ is the number of the i-th amino acid of $\vec{S}_{\text{PARA}}$, L is the length of the corresponding parameters in $\vec{S}_{\text{PARA}}$.

The DNACAs consist of several operators in which a crossover operation is used to generate a new strand that will retain beneficial features from the parent generation. This exploits current genetic potential. If the population doesn't contain enough encoded information to solve a particular problem, none of the mixing strands can produce a satisfactory solution.

The mutation operator capable is spontaneously generating new strands which provide a mechanism to maintain the population's diversity. In addition to the crossover and mutation operators, there are two frameshift operators, i.e. enzyme and virus, which will be explained in more details subsequently.

C. Crossover Operation with DNA sequences

The crossover operation causes recombination and information exchange of DNA sequences in a probabilistic way. Two DNA sequences are chosen from the current population and swap partial genes with each other. The variety of crossover algorithms work depending on the crossover rate (P_c) and the number of crossover points. Three types of crossover strategies have commonly been adopted [19].

The first one is two-point crossover operation, c_{tp}, that produces an intermediate population $P'^t = (a_1'^t, ..., a_\lambda'^t)$ from the original population $P^t = (a_1^t, ..., a_\lambda^t)$ in the t-th generation, is defined as

$$\begin{Bmatrix} a_i'^t \\ a_{(i+1)}'^t \end{Bmatrix} = c_{tp}\left(\begin{Bmatrix} a_i^t \\ a_{(i+1)}^t \end{Bmatrix} \right), \quad \forall i \in \{1, 3, \cdots, 2k+1, \cdots, \lambda - 1\}$$

$$= \begin{Bmatrix} \left[a_{(i+1),1}, a_{(i+1),2}, \cdots, a_{(i+1),\rho_1}, a_{i,(\rho_1+1)}, \cdots, a_{i,\rho_2}, a_{(i+1),(\rho_2+1)}, \cdots, a_{(i+1),L} \right] \\ \left[a_{i,1}, a_{i,2}, \cdots, a_{i,\rho_1}, a_{(i+1),(\rho_1+1)}, \cdots, a_{(i+1),\rho_2}, a_{i,(\rho_2+1)}, \cdots, a_{i,L} \right] \end{Bmatrix}$$

$a_i'^t, a_i^t \in S_k$, and $a_{(i+1)}'^t, a_{(i+1)}^t \in S_k$,

where $S_k = (x_1 x_2 x_3)$, $x_i \in \{A, G, T, C\}$,

and $1 \le \rho_1 < \rho_2 \le L$, ρ_1, ρ_2 indicate the crossover point.

The two-point crossover operation is a process of exchanging DNA information. The DNA sequence is divided into three sub-DNA sequences by the two points and crossovered each other by swapping the sequences between the first and third sub-DNA sequences. By this method, depending on the crossover points, a DNA sequence can be drastically changed. After crossover operation, the original DNA sequences between the two crossover points are exchanged as illustrated in Fig. 3.

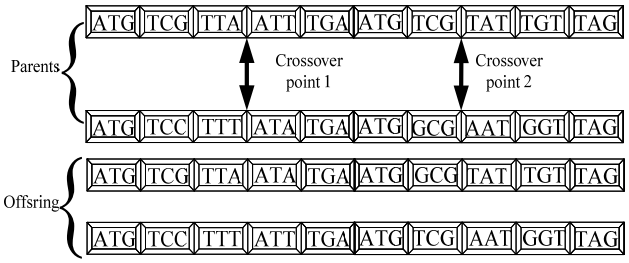


Fig. 3 Example of two-point crossover operation.

The second crossover strategy is multi-point crossover, c_{mp}, that produces an intermediate population $P'^t = (a_1'^t, ..., a_\lambda'^t)$ from the original population $P^t = (a_1^t, ..., a_\lambda^t)$, is defined as

$$\begin{Bmatrix} a_i^{'t} \\ a_{(i+1)}^{'t} \end{Bmatrix} = c_{mp}\left(\begin{Bmatrix} a_i^t \\ a_{(i+1)}^t \end{Bmatrix}\right), \quad \forall i \in \{1,3,\cdots,2k+1,\cdots,\lambda-1\}$$

$$= \bigcup_{k=1}^{L}\left\{\left(\begin{bmatrix} a_{i,k} \\ a_{(i+1),k} \end{bmatrix}, if\ \rho_k < \rho_{mp}\right) \vee \left(\begin{bmatrix} a_{(i+1),k} \\ a_{i,k} \end{bmatrix}, if\ \rho_k \geq \rho_{mp}\right)\right\}$$

$a_i^{'t}, a_i^t \in S_k$, and $a_{(i+1)}^{'t}, a_{(i+1)}^t \in S_k$

where $S_k = (x_1 x_2 x_3)$, $x_i \in \{A,G,T,C\}$, $1 \leq \rho_k, \rho_{mp} \leq \mathrm{L}$,

ρ_k indicates the position of multi-point crossover

ρ_{mp} denotes the threshold

In this crossover strategy, more than one crossover point is selected in a pair of chromosomes. The crossover operation is performed in bit level. The process of crossover positions in DNA sequence is selected randomly, distinctly from each other.

The third crossover strategy is uniform crossover, c_{un} that produces an intermediate population $P^{'t} = (a_1^{'t}, ..., a_\lambda^{'t})$ from the population $P^t = (a_1^t, ..., a_\lambda^t)$. A mask, a binary array with length L is generated, $0 \leq r_j \leq 1$, $(j = 1,2,\cdots,L)$ is generated randomly. If the j-th random number, r_j, is the j-th element in the binary array set as A(T). Otherwise, it is set to be C(G). The mask ma is defined as

$$ma = \bigcup_{j=1}^{L}\left\{\left([\mathrm{A(T)}],\ if\ r_j \geq \rho_{ma}\right) \vee \left([\mathrm{C(G)}],\ if\ r_j < \rho_{ma}\right)\right\}$$

The uniform crossover is defined as

$$\begin{Bmatrix} a_i^{'t} \\ a_{(i+1)}^{'t} \end{Bmatrix} = c_{un}\left(\begin{Bmatrix} a_i^t \\ a_{(i+1)}^t \end{Bmatrix}\right) \quad \forall i \in \{1,3,\cdots,2k+1,\cdots,\lambda-1\}$$

$$= \bigcup_{k=1}^{L}\left\{\left(\begin{bmatrix} a_{i,k} \\ a_{(i+1),k} \end{bmatrix}, if\ ma_k = \mathrm{C(G)}\right) \vee \left(\begin{bmatrix} a_{(i+1),k} \\ a_{i,k} \end{bmatrix}, if\ ma_k = \mathrm{A(T)}\right)\right\}$$

$a_i^{'t}, a_i^t \in S_k$, and $a_{(i+1)}^{'t}, a_{(i+1)}^t \in S_k$

where $S_k = (x_1 x_2 x_3)$, $x_i \in \{A,G,T,C\}$

Similar to the multi-point crossover strategy, the process of uniform crossover is performed bit by bit in a pair of DNA sequences. In the uniform crossover strategy, the crossover positions are predefined in a mask. All DNA sequences in a population are crossovered at the same positions. In other words, for the multi-point crossover strategy, each pair of DNA sequence is crossovered at different points because no predefined mask was used.

D. Mutation Operation with DNA sequence

The mutation operation is to change DNA sequences in a probabilistic way which is determined by the mutation rate P_m. Mutation injects new information into the generation that may be important but not contained in the initial population or lost in the selection process, it is helpful for escaping local optimal.

Mutation operation capable of spontaneously generating new strand provides a mechanism to maintain the population diversity as shown in Fig. 4. The Watson-Crick complementarity is commonly considered in mutation operation, i.e. $\mathrm{A}=\overline{\mathrm{T}}$; $\mathrm{T}=\overline{\mathrm{A}}$; $\mathrm{C} \equiv \overline{\mathrm{G}}$; $\mathrm{G} \equiv \overline{\mathrm{C}}$.

The operation of mutation m_p, that produces an intermediate population $P^{'t} = (a_1^{'t}, ..., a_\lambda^{'t})$ from the original population $P^t = (a_1^t, ..., a_\lambda^t)$, is defined as

$$a_i^{'t} = m_p(a_i^t) \quad \forall i \in \{1,2,\cdots,\lambda\}$$

$$a_{i,k}^{'} = \begin{cases} a_{i,k}, & for\ k \in \{1,2,\cdots,p-1,p+1,\cdots,L\} \\ \overline{a}_{i,k}, & for\ k = p \end{cases}$$

$a_i^{'t} \in S_k$, and $a_i^t \in S_k, \overline{a}_{i,k} \in S_k$

where $S_k = (x_1 x_2 x_3)$, $x_i \in \{A,G,T,C\}$, and $1 \leq p < L$

and p indicates the mutation point

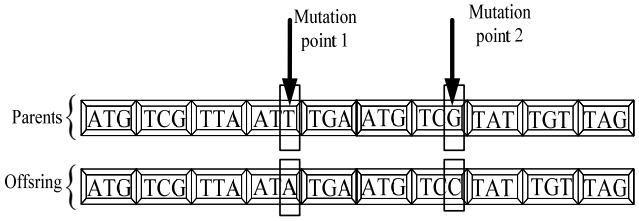


Fig. 4 Example of multi-point mutation operation.

E. Virus Operation with DNA sequences

From the biological viewpoint, the major cause of disease is usually that a virus has intruded into creatures, modifying the original codons of DNA strings and reproducing the infected strings. Through this mechanism, an inserted DNA string makes an incursion into its parent's string to form a new one. The direct consequence of this operation is that the resulting DNA string gains extra information in the next generation as displayed in Fig. 5.

To realize the effect, let the codons inducing virus influence be c_{vp} that produces an intermediate population $P^{'t} = (a_1^{'t}, ..., a_\lambda^{'t})$ from the original population $P^t = (a_1^t, ..., a_\lambda^t)$ in the t-th generation and is defined as

$$\{a_i^{'t}\} = c_{vp}\{a_i^t\} \quad \forall i \in \{1,2,3,\cdots,\lambda\}$$

$$\{a_{i+1}^{'t}\} = \{a_{i,1}, a_{i,2}, \cdots, a_{i,(\rho_1)}, \cdots, a_{i,(\rho_j)}, \cdots, a_{i,(\rho_2)}, \cdots, a_{i,L}\}$$

$a_i^{'t}, a_i^t \in S_k$, $a_{(i+1)}^{'t} \in S_k$ and $a_{i,(j)}^t \in S_k$

where $c_{vp} = \{a_{i,(j)}^t\} = \{a_{i,(\rho_1)}, \cdots, a_{i,(\rho_j)}, \cdots, a_{i,(\rho_2)}\}$, $1 \leq \rho_1 < \rho_j < \rho_2 \leq L$,

and $S_k = (x_1 x_2 x_3)$, $x_i \in \{A,G,T,C\}$

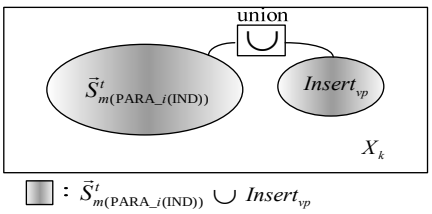


Fig. 5 Illustrations of virus operation.

F. Enzyme Operation with DNA sequence

The enzyme mutation works to separate and connect two DNA substrings while removing an intermediate fragment. The newly formed codon breaks away from the doped position and the reacted remainder of codons are joined together to form a new one. The direct consequence of this operation is that the resulting DNA string loses some information in the next generation, as displayed in Fig. 6. The net effect is to help the reacting molecules go through chemical changes more rapidly.

To realize the effect, let the codons eliminated by enzyme effect be c_{ep}; the start code, $a^t_{i,(\rho_1)}$; the stop code, $a^t_{i,(\rho_2)}$ that produces an intermediate population $P^{'t} = (a_1^{'t}, ..., a_\lambda^{'t})$ from the population $P^t = (a_1^t, ..., a_\lambda^t)$ in the t-th generation and

$$\{a_i^{'t}\} = c_{ep}\{a_i^t\} \quad \forall i \in \{1, 2, 3, \cdots, \lambda\}$$

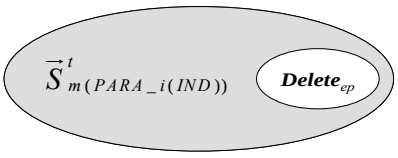
$$\{a_{i+1}^{'t}\} = \{a_{i,1}, a_{i,2}, \cdots, a_{i,(\rho_{1-1})}, a_{i,(\rho_{2+1})}, \cdots, a_{i,L}\}$$

$$a_i^{'t}, a_i^t \in S_k,\ a_{i+1}^{'t} \in S_k \text{ and } a^t_{i,(j)} \in S_k$$

where $c_{ep} = \{a^t_{i,(j)}\} = \{a^t_{i,(\rho_1)}, a^t_{i,(\rho_{1+1})}, \cdots, a^t_{i,(\rho_j)}, \cdots, a^t_{i,(\rho_{2-1})}, a^t_{i,(\rho_2)}\}$

$\forall j \in \{2, 3 \cdots, \lambda - 1\}$ and $1 \le \rho_1 < \rho_j < \rho_2 \le L$

$S_k = (x_1 x_2 x_3),\ x_i \in \{A, G, T, C\}.$



$\square : \vec{S}^{\,t}_{m(PARA\_i(IND))} \varnothing\ Delete_{ep}$

Fig. 6 Illustrations of enzyme operation.

III. ANALYSIS METHODS

A. Configurable ID model

There have been traditional or advanced approaches widely applied to deal with the system ID problem [20-22]. We consider here a novel application of DNACA to tackle the problem.

Consider a class of transfer functions modeled as follows

$$\hat{P}_E(s) = \hat{C}_C \frac{\prod_{i=1}^{n_n}(s + \hat{C}_{N1\_i}) \prod_{i=1}^{m_n}(s^2 + \hat{C}_{N2\_i} s + \hat{C}_{N3\_i})}{\prod_{i=1}^{n_d}(s + \hat{C}_{D1\_i}) \prod_{i=1}^{m_d}(s^2 + \hat{C}_{D2\_i} s + \hat{C}_{D3\_i})} \tag{3}$$

where $\tilde{n}_n = n_n + 2m_n$ and $\tilde{n}_d = n_d + 2m_d$ with $\tilde{n}_d \ge \tilde{n}_n$ assure the transfer function to be proper. In traditional system ID methodologies, the least mean square error scheme is adopted to construct a transfer function so that it ultimately mimics the dynamic behavior of the identified object. The resulting transfer function, however, might not possess the simplest structure, i.e. the one is not necessarily with the minimum order.

The objective function for the current problem is defined as

$$J = \sum_{i=1}^{3} w_i J_i \tag{4}$$

where w_i is the weighting factor for the corresponding term:

$$J_1(\vec{S}_{\text{IND}}) : G_e^{Low} = \left| \frac{Mag(\hat{P}_E(j\omega_{Lf}))}{10^{-\exp(\log(\omega_{Lf}))}} - \frac{Mag(P_0(j\omega_{Lf}))}{10^{-\exp(\log(\omega_{Lf}))}} \right| \tag{5}$$

$$J_2(\vec{S}_{\text{IND}}) : M_e = \sum_{i=1}^{f_t} \left| \ln(Mag(\hat{P}_E(j\omega_i))) - \ln(Mag(P_0(j\omega_i))) \right| \tag{6}$$

$$J_3(\vec{S}_{\text{IND}}) : PA_e = \sum_{i=1}^{f_t} \left| phase(\hat{P}_E(j\omega_i)) - phase(P_0(j\omega_i)) \right| \tag{7}$$

$P_0(s)$ denotes the system to be identified, ω_{Lf} represents the lowest frequency, and $Mag(\cdot)$ and $phase(\cdot)$ indicate, respectively, the magnitude and phase of the frequency response; f_t is the total sampling number over the testing frequency interval. Here, J_1 denotes the magnitude error at the lowest frequency; whereas J_2 and J_3 calculate, respectively, the overall errors of the magnitude and phase over the whole spectrum.

B. Extension to Control Design

Consider the control system shown in Fig. 7 with its controller $\hat{C}_E(s)$ described in the form of (3). The following condition is a fundamental requirement for the nominal closed-loop stability:

$$\text{Re}\,\lambda_i(\Lambda(s)) < 0, \quad \forall i \tag{8}$$

where λ_i are the roots of the characteristic polynomial $\Lambda(s)$ of the nominal closed-loop system.

The condition for internal stability is ensured by incorporating the following constraint:

$$g_1 \triangleq \left\| \frac{W_{rc}(j\omega)\hat{C}_E(j\omega)}{1 + \hat{C}_E(j\omega)G_n(j\omega)} \right\|_\infty < 1 \tag{9}$$

where $G_n(s)$ is the nominal plant model which could be obtained by using the previous ID technique; $\|A(j\omega)\|_\infty = \sup_\omega |A(j\omega)|$; the weighting function $W_{rc}(s)$ works so that there won't be excessive control commands, especially at higher frequencies.

The conditions for robust stability and sensitivity reduction have been well known [5-7] and are given, respectively, as follows

$$g_2 \triangleq \left\| \frac{W_{rs}(j\omega)\hat{C}_E(j\omega)G_n(j\omega)}{1+\hat{C}_E(j\omega)G_n(j\omega)} \right\|_\infty < \delta(\omega) \tag{10}$$

$$g_3 \triangleq \left\| \frac{W_{rs}(j\omega)}{1+\hat{C}_E(j\omega)G_n(j\omega)} \right\|_\infty < 1-\delta(\omega) \tag{11}$$

where $\delta \in (0,1), \forall \omega \geq 0$, is used to control relative importance between stability and performance. The term $W_{rs}(s)$ satisfying $|\Delta G_n(j\omega)| \leq |W_{rs}(j\omega)|, \forall \omega$ is the weighting function used to bound the multiplicative uncertainty, which possesses a sufficiently high gain in low frequencies to get a disturbance suppression property and eliminate the steady state error. The multiplicative uncertainty is used here for its simplification and direct relationship with the complementary sensitivity function.

Control design simultaneously satisfying above constraints ensures

$$\left| \frac{W_{rs}(j\omega)\hat{C}_E(j\omega)G_n(j\omega)}{1+\hat{C}_E(j\omega)G_n(j\omega)} \right| + \left| \frac{W_{rs}(j\omega)}{1+\hat{C}_E(j\omega)G_n(j\omega)} \right| < 1, \forall \omega \geq 0$$

which is a sufficient condition assuring robust performance of the control system simultaneously subject to plant uncertainties and external disturbances.

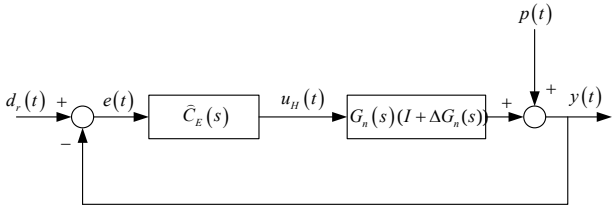


Fig .7 Typical closed-loop control system with uncertainties.

Objective function

The problems of control design can be viewed as requiring the discovery of a controller or a control strategy that takes the output variables or state variables of a problem as its inputs and produces the values of the control variable(s) as its outputs. The DNA programming is well suited to resolve control design problems where no exact solution is known and where an exact solution is not required [23].

Minimization of the following quadratic performance index is introduced here for the requirement of tracking accuracy:

$$J_1 = \frac{1}{T}\int_0^T e^T(t)e(t)dt \tag{12}$$

where T is chosen sufficiently large so that $e(t)$ for $T < t$ is negligible. The cost formulation covering the error energy over the whole time horizon of interest has been used extensively for both deterministic inputs and statistical inputs. However, system designs by this criterion tend to display a rapid decrease in the large initial error. Thus the systems may have poor relative stability. To complement the weakness, the following objective function in time domain is defined to directly reflect the transient performance:

$$J_2 = \varepsilon_1(1-M_o)^2 + \varepsilon_2(1-T_r)^2 + \varepsilon_3(1-E_{ss})^2 \tag{13}$$

where the weighting factors $\varepsilon_i \geq 0$ with $\sum_{i=1}^{3}\varepsilon_i = 1$; M_o with $0 \leq M_o \leq 1$ being the normalized maximum overshoot; T_r with $0 \leq T_r \leq 1$ being the normalized rise time; E_{ss} with $0 \leq E_{ss} \leq 1$ being the normalized steady-state error.

DNACA-Based control design

Although a complicated controller usually enables superior performance, a simpler structure with acceptable performance is more practically desirable. Therefore, objective functions considered should contain not only system performance indices but also structure information of the controller.

The fitness function for the current problem is defined as

$$f(\cdot) = F'(\cdot)p(\cdot) \tag{14}$$

where $F'(\cdot)$ is the unconstrained fitness term converted from the following objective function:

$$J_{all} = (1-\beta e^{-\varsigma g})J_p + \beta e^{-\varsigma g}J_s, \quad 0.5 \leq \beta \leq 1, \varsigma > 0 \tag{15}$$

where

$$J_p(\vec{S}_{\mathrm{IND}}) = \alpha_1 J_1(\vec{S}_{\mathrm{IND}}) + \alpha_2 J_2(\vec{S}_{\mathrm{IND}}), \ \alpha_1 + \alpha_2 = 1 \tag{16}$$

$$J_s(\vec{S}_{\mathrm{IND}}) = \frac{\tilde{n}_d^{\mathrm{IND}}}{\max\limits_{1 \leq \mathrm{IND} \leq \mu} \tilde{n}_d^{\mathrm{IND}}} \tag{17}$$

and g is the generation number, β represents the desired emphasis on controller's complexity, the penalty term $p(\cdot)$ is defined as

$$p(\cdot) = \left(1 - \frac{1}{3}\sum_{i=1}^{3}\frac{\Delta b_i(\cdot)}{\max\{\varepsilon_p, \Delta b_i(\cdot)\}}\right)\upsilon(\tilde{n}_d, \tilde{n}_n) \tag{18}$$

where

$$\upsilon(\tilde{n}_d, \tilde{n}_n) = \begin{cases} 1, & \text{if } \tilde{n}_d \geq \tilde{n}_n \\ 0, & \text{if } \tilde{n}_d < \tilde{n}_n \end{cases}$$

and $\Delta b_i(\cdot) = \max\{0, g_i(\cdot) - b_i\}, i = 1,2,3$ with $b_1 = 1$, $b_2 = \delta$ and $b_3 = 1-\delta$, ε_p is a small positive constant.

The linear ranking approach can be used to convert J_{all} to F'. The formulation of (15) places a heavier weight on the control structure selection during the early generations, the weight shifts gradually to the parameters selection emphasizing on the system performance improvement. Balance between the two factors with the generation of evolution could be modified by appropriately adjusting the constant ς.

IV. Numerical Study

To examine applicability and efficiency of the proposed DNACA, the following De Jong's test functions are considered:

$$D_1 = \sum_{i=1}^{3} x_i^2, -5.12 \le x_i \le 5.12 \tag{19}$$

This is a simple 3-dimensional parabola with a spherical constant-cost contour.

$$D_2 = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2, \ -2.048 \le x_i \le 2.048 \tag{20}$$

This is a typical test function to deal with the optimization. The point (1, 1) is a minimum of zero and it is very difficult to perceive because a deep parabolic valley is along with the relation $x_2 = x_1^2$.

$$D_3 = \sum_{i=1}^{3} i x_i^4 + \text{Gauss}(0,1), \ -1.28 \le x_i \le 1.28 \tag{21}$$

This is a continuous, convex, unimodal, and 3-dimentional quadric which function with zero-mean Gaussian noise.

$$D_4 = 0.002 - \sum_{i=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}, -65.356 \le x_i \le 65.356 \tag{22}$$

This is a multi-modal function; it has 25 local minimums lying approximately at the points $a_{ij}, i = 1, 2 \ j = 1, \cdots, 25$.

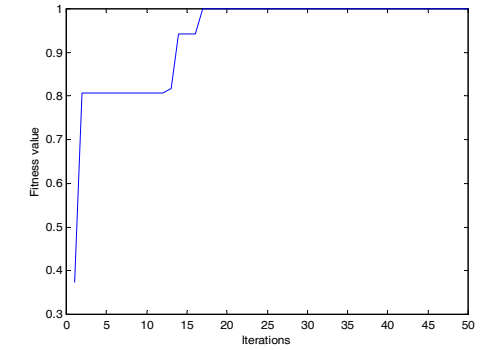
$$D_5 = \sum_{i=1}^{3} [x_i^2 - 10\cos(2\pi x_i) + 10], \ -5.12 \le x_i \le 5.12 \tag{23}$$

This is a generalized rastrigin's function. It possesses multiple local optimums that is suitable to examine performance of the optimization algorithms.

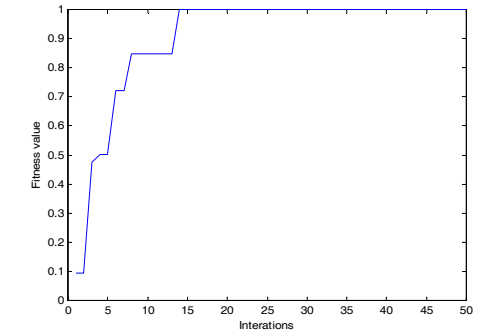
The corresponding fitness functions F_D^1 for $D_{1,2,3,5}$ and F_D^2 for D_4 are given, respectively, as

$$F_D^1 = \frac{1}{1 + D_i(x)}, \ i = 1, 2, 3, 5; \ F_D^2 = \frac{1}{(|1 - D_4| + 1)} \tag{24}$$

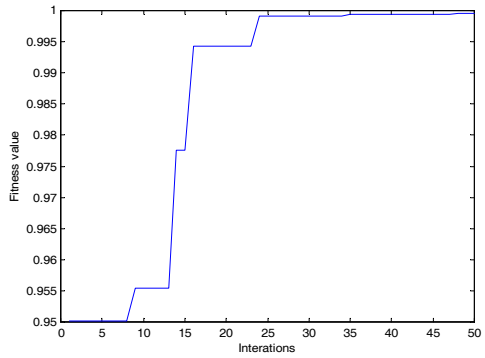
For each test function, the iteration and population numbers are set to be 50 and 10, respectively. The probability rates of crossover and mutation are 0.8 and 0.2, respectively. Both of the probability rates of enzyme and virus are 0.4. The results of the solution are searched by using the proposed DNACA as shown in Fig. 8 and Table II. Taking D_4 and D_5 as the example, the efficiency of DNACA with/without frameshift operators is illustrated as in Table III. Convergence of the fitness values for D_4 and D_5 is displayed as in Fig. 9. It is seen that the DNACA produces not only accurate solutions, it simultaneously prevents the redundant candidate strings from being repeated in the evolutionary process. These results demonstrate efficiency and performance of the presented algorithm.



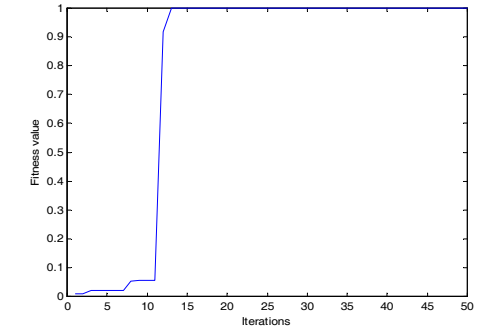
(a)



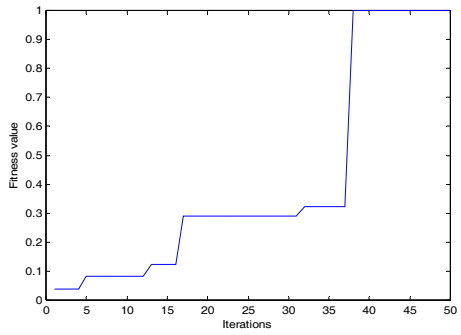
(b)



(c)



(d)



(e)

Fig. 8 Convergence of fitness values through DNACA. (a) D_1; (b) D_2; (c) D_3; (d) D_4; (e) D_5.

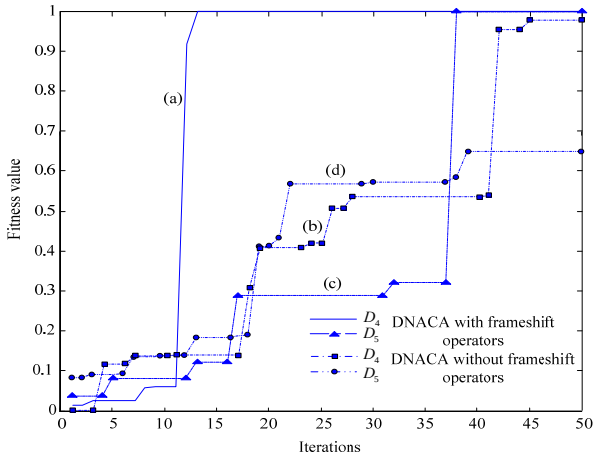


Fig. 9 Convergence of the fitness values for D_4 and D_5; (a,c) with frameshift operation; (b,d) without frameshift operation.

Table II. Results of the proposed DNACA, the standard GA [24] and the improved GA [25, 26] for the De Jong's function tests.

De Jong's test functions (converging generation)	Proposed DNACA	Standard GA	Improved GA
$D_1(x)$	1.0000(25)	1.0000(225)	1.0000(100)
$D_2(x)$	1.0000(20)	0.6393(>450)	0.9707(>045)
$D_3(x)$	0.9995(40)	0.8037(>450)	0.8349(>450)
$D_4(x)$	1.0000(25)	1.0000(200)	1.0000(>450)
$D_5(x)$	1.0000(45)	0.7297(>500)	1.0000(75)

Table III. Comparison of DNACA's efficiency with/ without frameshift operators for D_4 and D_5.

Condition	With frameshift operations	Without frameshift operations	With frameshift operations	Without frameshift operations
Function	D_4		D_5	
Fitness value	1,000	0.991	1,000	0.6465
Required generations for convergence	20	50	45	45
Length (codons)	8,3	6,6	1,2,2	6,6,6
DNA string and decimal values of parameters	CATTTCTCA GTTAAATA TGAAGTC(-3 1.4014) GCACAGTC T (31.6239)	ACGGCGCC G GCCCACGA A (-31.5653) AGGGCCAA A GCGCAGAT G (31.3896)	TTA(0) TTGGGA(0) TACTGG(0)	TTTTTTAGACC TTATAAG (-0.0413) ACCCAGAACT ATTGGTGG (-0.0282) TATTGGGAAG CTGCAGCA (-0.0164)

DNACA is then applied to identify the following system

$$P_0(s)=\frac{1.038\left(\frac{s}{300}+1\right)\left(\frac{s^2}{0.0911^2}+\frac{2\times1.17s}{0.0911}+1\right)\left(\frac{s^2}{0.7587^2}+\frac{2\times1.129s}{0.7587}+1\right)}{s\left(\frac{s}{4.012}+1\right)\left(\frac{s^2}{0.1068^2}+\frac{2\times2.712s}{0.1068}+1\right)\left(\frac{s^2}{2.168^2}+\frac{2\times1.59s}{2.168}+1\right)} \quad (25)$$

The aim is to find out a simplified transfer function to faithfully mimic the system behavior in the frequency domain. The frequency band under consideration is $[10^{-2},10^{3}]$ rad/s, four frequency intervals are considered individually including $[10^{-2}, 1]$, $[10^{-2}, 10]$, $[10^{-2}, 10^{2}]$ and $[10^{-2},10^{3}]$ (rad/s). For each section, the DNACA is used to optimize the structures and parameters of the identified transfer function. Every section encompasses the result obtained in the previous section as the basis for the optimization process.

The parameter settings for the DNACA are set as follows: the population size is 70, the resolution is 15 codons, the maximum generation is 800, the parameter range is [-200, 200], the rates of enzyme and virus are 0.4, respectively.

After 800 generations of evolution (i.e. every interval performed 200 generations of evolution), the final results are shown in Fig.10. For the first frequency interval, the DNACA generates a transfer function approximating the low frequency behavior of $P_0(s)$. As it is displayed in Fig. 10(a), there exhibits a slight discrimination in the Bode magnitude and the phase plots of the two systems. Clearly, the parameters and structure of the system model identified at this stage need to be refined. Moving further to the subsequent stages of evolution with the result obtained in the first stage as the initial model, the ID error attenuates with the increasing frequency interval of interest. The result reaches its optima over the whole frequency range after the fourth stage of ID, see Figs. 10(b)-10(d). The summarized details are shown in Table IV.

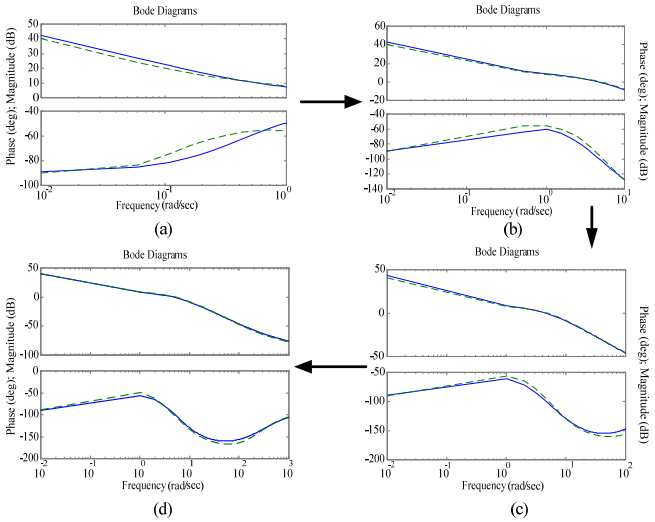


Fig. 10. Results of four step system ID; the real line indicates the original system response, the dashed line shows the identified system response within the following frequency intervals: (a) $[10^{-2}, 1]$; (b) $[10^{-2}, 10]$; (c) $[10^{-2}, 10^{2}]$; (d) $[10^{-2}, 10^{3}]$(rad/s).

Table IV. Results of ID in four frequency intervals

Performance indices / Estimation Model	Max. ID error (a)	Order of num/den polynomials (b)	$\exp^{-((a)x0.1(b))}$
$\left.\frac{1.97(s+5.67)(s+9.40)(s+18.4)}{s(s+1.32)(s+52.81)}\ \frac{(s+109.1)(s+193.6)(s+0.53)}{(s+54.45)(s+62.29)(s+0.79)}\right\|_{10^{-2}\sim 4}$	2.8030 db (at 0.06 rad/sec	12	0.0346
$\left.\frac{0.2332(s+0.5593)(s+192.1)}{s(s+2.199)(s+7.964)}\right\|_{10^{-2}\sim 10}$	2.8536 db (at 0.01 rad/sec)	5	0.2401
$\left.\frac{0.2332(s+0.5593)(s+192.1)}{s(s+2.199)(s+7.964)}\right\|_{10^{-2}\sim 10^{2}}$	2.8536 db (at 0.01 rad/sec)	5	0.2401
$\left.\frac{0.1355(s+0.414)(s+189.8)(s+194.8)}{s(s+6.236)(s+3.081)(s+103.1)}\right\|_{10^{-2}\sim 10^{3}}$	1.7315 db (at 260.01 rad/sec)	7	0.2975

Next, control design based on DNACA is examined. The performance index of the objective function J_2 is ignored in the following case.

Using Matlab to solve for the robust control problem defined by (10) gives

$$\widehat{C}_{H_{opt}} = \frac{4.973\times10^{-7}s^4 + 1.044\times10^{-6}s^3 + 5.071\times10^{-6}s^2 + 9.942\times10^{-6}s - 7.683\times10^{-9}}{s^5 + 2.234s^4 + 10.49s^3 + 21.38s^2 + 2.768s + 0.1801}$$

and the resulting $g_2 = 0.02$.

For the proposed DNACA, the population size is set as 50, crossover rate is 0.8, mutation rate is 0.2, the weighting factor ε_i is 0.2, both the enzyme and virus rates are 0.4, the maximum and minimum bounds of the information fragments are [1, 10]. A controller is obtained after 20 generations of evolution as

$$\widehat{C}_E(s) = \frac{3.9802(s+6.503)}{s+17.23}$$

The final result for the robust stability index is $g_2 = 0.03$. Convergent behavior of the controller's structure and g_2 are shown in Figs. 10 and 11. Although this value is slightly worse than the one obtained through using Matlab, the controller with lower order, while sacrificing performance a bit, is of more practical.

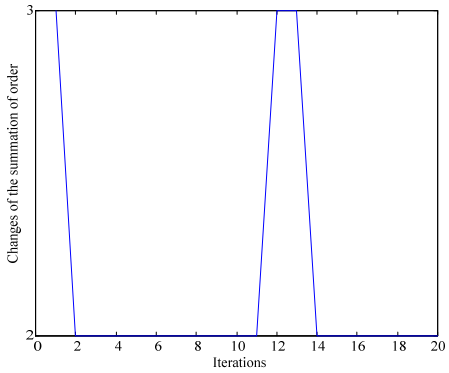


Fig. 10. Changes of the summation of denominator and numerator order.

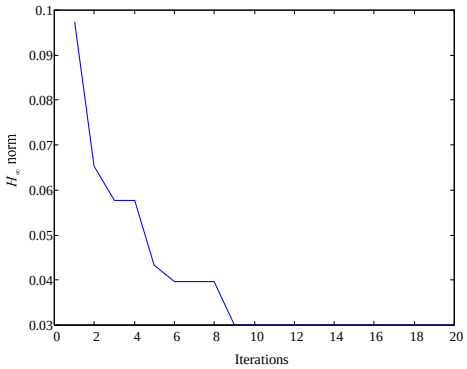


Fig. 11. Convergence of the H_∞ norm.

V. Conclusions

A DNA computing algorithm with new coding method based on EIIP of DNA codons is introduced. The EIIP is used instead of the traditional binary coding system. Capability for seeking the solutions of the multiple variable functions is confirmed by considering a class of De Jong's functions as the testing object. The proposed DNACA has also been successfully extended to deal with the system ID and control design problem. Numerical experiments confirm its excellence for these problems. The algorithm's multiple mutation mechanism allows it to simplify the model structure simultaneously while generating the optimal parameter set during the evolution process.

References

[1] L. M. Adleman, Molecular computation of solutions to combinational problems, *Science*, vol. 266, pp. 1021-1024, 1994.
[2] C. C. Maley, DNA computation theory, practice, and prospects, *Evolutionary Computation*, vol. 6, pp. 201-229, 1998.

[3] N. Forbes, *Imitation of Life: How Biology is Inspiring Computing*, MIT Press, Cambridge, 2004.
[4] C. C. Maley, DNA Computation theory, practice, and prospects. *Evolutionary Computation*, vol. 6, pp. 201-229, 1998.
[5] Y. Tsuboi, Z. Ibrahim, and O. Ono, DNA computing approach to knowledge representation. *Int. J. Hybr. Intell. Syst.*, vol. 2, pp.1-12, 2005.
[6] X. Yao, Y. Liu, and G. Lin, Evolutionary programming made faster. *IEEE Tran. Evo. Comp.*, vol. 3, pp. 82-102, 1999.
[7] J. Chen, E. Antipov, B. Lemieux, W. Cedeno, and D. H. Wood, DNA computing implementing genetic algorithms, *Proc. DIMACS Workshop on Evolution as Computation,* Princeton, pp. 39-49, 1999.
[8] V. Veljkovic, N. Veljkovic, J. A. Esté, A. Hüther, and U. Dietrich, Application of the EIIP/ISM bioinformatics concept in development of new drugs, *Cur. Med. Chem.,* vol. 14, no. 4, pp.441-53, 2007.
[9] C. Tintori, F. Manetti, N. Veljkovic, V. Perovic, J. Vercammen, S. Hayes, S. Massa, M. Witvrouw, Z. Debyser, V. Veljkovic,and M. Botta, Novel virtual screening protocol based on the combined use of molecular modeling and electron-ion interaction potential techniques to design HIV-1 integrase inhibitors, *J. Chem. Inf. Model,* vol. 5, pp.261-265, 2007.
[10] S. Bumble, *The Orchestral Analog of Molecular Biology*, Philadelphia College, USA, 2008.
[11] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1995.
[12] K. J. Astrom and T. Hagglund, *PID Controllers*, Instrument Society of America, 1995.
[13] M. J. Grimble, *Robust Industrial Control: Optimal Design Approach for Polynomial Systems*, Prentice Hall, 1994.
[14] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control*, John Wiley & Sons, Inc, 2003.
[15] K. Zhou, J. C. Doyle and K. Glover, *Robust and Optimal Control*, Prentice Hall, 1996.
[16] G. Luis, J. L. Manuel, and J. Lorenzo, Hardware/Software environment for process ID, robust controller design and hard real time implementation, *Proc. WSEAS International Conference on Telecommunications and Informatics*, pp. 503-508, Istanbul, 2006.
[17] D. Jong, K. A., *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, University of Michigan, Ann Arbor, MI, 1975.
[18] I. Cosic, V. Vojisavljevic, and M. Pavlovic, The relationship of the resonant recognition model to effects of low-intensity light on cell growth, *Int. J. Radiat. Biol*, vol. 56, pp.179-191, 1989.
[19] H. Adeli and S. L. Hung, *Machine Learning Neural Networks, Genetic Algorithms and Fuzzy System*, John Wiley & Sons, Inc, 1995.
[20] E. Altamiranda, R. Calderon, and E. C. Morles, An evolutionary adaptation strategy for dynamic systems ID, *WSEAS Trans. Syst. Contr.*, vol. 2, pp. 261-266, 2007.
[21] A. H. Mazinan and N. Sadati, Multiple modeling and fuzzy predictive control of a tubular heat exchanger system, *WSEAS Trans. Syst. Contr.*, vol. 3, pp. 249-258, 2008.
[22] H. Ghorbani, A. Ghaffari, and M. Rahnama, Constrained model predictive control implementation for a heavy-duty gas turbine power plant, *WSEAS Trans. Syst. Contr.*, vol. 3, pp. 507-516, 2008.
[23] C. Rentea, Neuro-controller design using genetic optimized pole placement method, *Proc. WSEAS International Conference on Automatic Control, Modeling and Simulation*, pp. 319-323, Prague, 2005.
[24] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, MA, 1989.
[25] H. F. L. Frank, H. K. Lam, S. H. Ling, and K. S. T. Peter, Tuning of the structure and parameters of a neural network using an improved genetic algorithm, *IEEE Trans. Neural Networks*, vol. 14, pp.79-88., 2003.
[26] Y. Hanaki, T. Hashiyama, and S. Okuma, Accelerated evolutionary computation using fitness estimation, *Proc. IEEE International Conference on Systems*, pp. 643-648, 1999.

Ching-Huei Huang was born in Zhanghua, Taiwan, ROC, in 1978. He received the B.S. degree in Department of Mechanical Engineering of Chung Cheng Institute of Technology, Taoyuan, Taiwan, ROC, in 2000. He received the M.S. degree in 2006, and he is currently working toward the Ph. D degree in the Department of Electrical Engineering at National Chung Hsing University, Taichung, Taiwan, ROC. His research interests are in evolutionary algorithms, optimal control, and robust control.

Chun-Liang Lin was born in Tainan, Taiwan, ROC, in 1958. He received the Ph.D. degree in aeronautical and astronautical engineering from the National Cheng Kung University, Tainan, Taiwan, ROC, in 1991. He was an Associate Professor and Professor in the Department of Automatic Control Engineering at Feng Chia University, Taichung, Taiwan, ROC, from 1995 to 2003. He is currently a Distinguished Professor in the Department of Electrical Engineering at National Chung Hsing University, Taichung, Taiwan, ROC. His research interests include guidance and control, intelligent control, and robust control. Dr. Lin received the Distinguished Research Award from National Science Council of Taiwan, ROC, in 2000 and 2003.

Horn-Yong Jan was born in Taoyuan, Taiwan, ROC, in 1978. He received the M.S. degree in automatic control engineering from Feng Chia University, Taichung, Taiwan, ROC, in 2003. He received his Ph.D degree from Institute of Electrical and Communications Engineering, Feng Chia University, Taichung, Taiwan, ROC, in 2007. His research interests are in bioinformatics, evolutionary algorithms, and motor control.