

## 基因演算法於控制系統的應用

### The Application of Genetic Algorithm in Control Systems

陳木松

Mu-Song Chen

大葉大學電機工程系  
Dept. of Electrical Engineering  
Da-Yeh University  
chenms@aries.dyu.edu.tw

廖鴻翰

Fong Hang Liao

大葉大學電機工程系  
Dept. of Electrical Engineering  
Da-Yeh University  
s857316@aries.dyu.edu.tw

#### 摘要

傳統的控制模式需要對受控系統有充份的了解，方可達到最佳化控制的目的。但隨著受控系統的日益複雜或控制的過程是非線性模式，最佳化控制器的設計更加困難。所以我們研擬應用基因演算法於控制器的設計，希望藉由演化的程序求解相關的控制參數。本文將應用適應性基因演算法調整互置機率與突變機率，並依適應值的變化情形，結合不同的互置方法作互置運算，以期提昇基因演算法的效率。本文的模擬測試以LQ(Linear-Quadratic)的問題，在不同的參數條件下進行系統最佳化的試驗。實驗數據顯示，由基因演算法所得的結果與已知最佳控制的結果相當符合。

關鍵字：適應性基因演算法，互置，控制系統

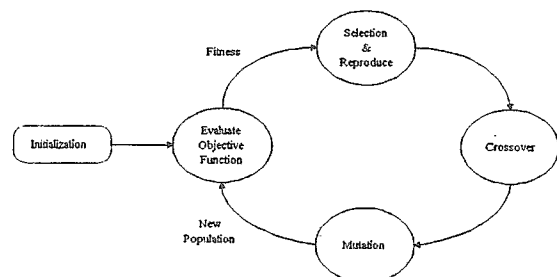
#### Abstract

The successful application of classical control design techniques usually requires an extensive knowledge of the system to be controlled, including an accurate model of its dynamics. However, as the system becomes more complex and nonlinear, the process of optimal design is more and more difficult. In this paper, we proposed to utilize Genetic Algorithms (GAs) as tools to model control processes for the LQ(Linear-Quadratic) control systems. The required control parameters of the systems are evolved by natural selection and evolution. To promote the efficiency, the adaptive GA is used to dynamically adjust the probabilities of crossover and mutation according to the fitness of individual. In addition, two crossover operators are mixed during evolution to maintain population diversity and a balance between exploration and exploitation. To verify the efficiency of the proposed method, the desired control sequence of 1-dimensional LQ control system were solved by optimal control theory as well as GA. The experimental results showed that the control sequences and steady states for plant are quite consistent with the results of optimal control.

Keywords: Adaptive Genetic Algorithms, Crossover, Control System

#### 1. 前言

傳統的控制方法大都是先求得系統的增益以控制系統的行為，然而對於高度非線性系統，其精確的數學模式不易獲得[1]，因此傳統的線性增益控制便無法達到高準確度及快速的控制。過去幾年間由於人工智慧的蓬勃發展，於是便有以模仿人類神經系統的神經網路[2]、模擬人類推理方式的模糊系統[3]、或二者的混合所架構的控制器。這類的控制器應用激發受控系統得到其輸入/輸出的關係，即可建立其網路架構或模糊規則庫，再經由疊代學習的方法，以掌握受控系統的行為。由於此一過程類似人類的學習模式所以亦稱為智慧型控制器。一般而言，智慧型控制器學習的方法大都是以梯度法[4]，漸次逼近函數最佳的極值點。然而梯度法的缺點是易陷於區域性極點，而且對於不連續函數其微分亦無法求得。本文研擬應用基因演算法(Genetic Algorithms, GAs)[5,6]於控制系統，希望藉由物種演化的原則：『物競天擇，適者生存』(Survival of the best)，以求出控制系統的最佳控制模式。GA是Holland教授所提出的，其最初的目的是要建立一個保有自然特性的人工系統，以解釋生物系統的進化過程。GA與傳統搜尋法不同處是，它採多點搜尋的方式在參數的空間中尋求最佳的解值，而不需要透過特定的轉移規則、目標函數的導數或相關之輔助訊息。而僅需藉由系統所定義目標函數值(Fitness)作為演化時互置及突變運算的依據，再經由預設的互置機率(Prob. of Crossover,  $P_c$ )及突變機率(Prob. of Mutation,  $P_m$ )進行演化，以保留優良的母代，同時適切地調整搜尋的方向及區域，使能逐漸找到最佳解。圖一所示為基因演算法的演化流程圖。



圖一、基因演算法的演算流程圖。

現階段應用基因演算法求解問題時，對於族群數目的多寡、演化的世代數(Generations)、編碼的方式(Coding strategy)、字串的長度、互置及突變的方式、及互置與突變機率等的決定，並無一定論，通常需視所欲求解問題的函數特性，再以嘗試與錯誤(Trial and Error)的經驗法則，決定出較好的參數組合。其中有關互置與突變機率的決定更是影響系統是否收斂的重要因素。當GA演化時如果 $P_c$ 值設的太大，則在進行族群互置時，會因為無法確認被複製的母代是否是優良或是較差的，所以易使系統陷入區域性極點。但是若將 $P_c$ 值設的太小，則原本適應性較佳的母代將無法完整地將其良好的特性留傳給子代，而且也可能使系統發生不穩定的現象而無法收斂。至於突變的目的是為防止系統過早收斂，同時也有助於增加子代的多樣性。就突變機率的設定，若 $P_m$ 值太低而系統又剛好陷落在全域的某一區域中(非最佳解的區域)，就容易造成系統的提早收斂。反之若 $P_m$ 的值太大，固然能使系統跳離區域極點，但是因為突變的次數太頻繁，很容易使系統又陷入另一不確定的區域之中，使得系統再次不穩定而無法收斂。所以本文的研究主要是應用適應性基因演算法則(Adaptive Genetic Algorithms, AGA)[7,8]動態地調整互置與突變機率，並依世代間最佳適應值的變化情形，以不同的互置運算子作為演化時互置運算依據，以提昇GA的效率。以下首先就AGA的方法提出說明。

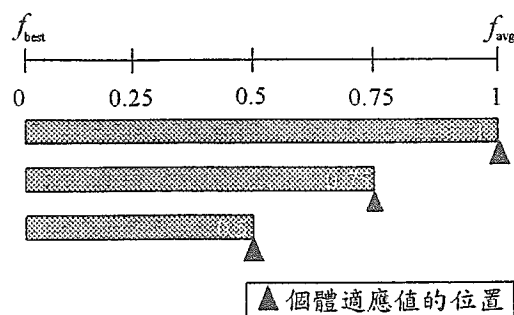
## 2. 適應性基因演算法

適應性基因演算法則(AGA)是以每一個體的適應值，動態地調整其互置機率與突變機率。假設在某一世代中其最佳適應值與平均適應值分別為 $f_{best}$ 及 $f_{avg}$ ，如果欲互置的個體適應值分別為 $f_i$ 及 $f_j$ ，則互置與突變機率決定如下：

$$\begin{cases} P_c = k_1 \frac{f' - f_{best}}{f_{avg} - f_{best}}, & f' \leq f_{avg} \\ P_c = k_2, & f' > f_{avg} \end{cases} \quad (1)$$

$$\begin{cases} P_m = k_3 \frac{f - f_{best}}{f_{avg} - f_{best}}, & f \geq f_{avg} \\ P_m = k_4, & f < f_{avg} \end{cases} \quad (2)$$

其中 $f'$ 為互置時兩個母代中適應性較佳者的適應值(就函數最小化而言，適應性較佳的個體其適應值較小)，而 $f$ 為突變時母代的適應值。由(1)、(2)式得知，AGA對於 $P_c$ 、 $P_m$ 值的決定是以 $f_{avg}$ 作為判斷的依據，亦即對於適應值低於平均適應值的個體(較佳的個體)給予不同等級的 $P_c$ 及 $P_m$ 值，再依據個體的適應值，依比例(接近最佳適應值的程度)分成不同的等級如下圖：



但是對於適應值高於平均適應值的個體，則給予預設的 $P_c$ 及 $P_m$ 值。就演化的理論而言，較佳的個體容易產生比較好的子代，所以AGA對於適應值較佳的個體，希望保留較多該個體的特質於其子代上，所以相對應的 $P_c$ 及 $P_m$ 值亦較小，以減少該個體被互置及突變的機率。反之對於適應值離最佳適應值較遠的個體，則給予較大的 $P_c$ 及 $P_m$ 值，以提高其被互置及突變的機率，讓其子代遺傳較少該個體的特質。雖然AGA亦是基於以上的原則推導互置/突變的機率，然而參數 $k_i$ 必須預先設定。依據[7]的建議 $\{k_1, k_2, k_3, k_4\} = \{0.75, 0.75, 0.5, 0.5\}$ ，而這樣的參數組合對於多極點函數的最佳化問題，無法有效地適應性調整 $P_c/P_m$ 。此外當 $f_{best} = f_{avg}$ (此時系統可能陷於區域極值)時，若依(1)、(2)式則互置與突變機率便無法計算。

為解決上述的問題及提昇AGA的搜尋能力，我們提出混合式的互置運算法，亦即選取特性不相似的互置運算子，予以適當的組合，以提昇其對函數最佳化的效能。以下的敘述主要是針對實數字串編碼(亦即染色體的每一基因是以實數表示其解值)，今分析說明如后。

## 3. 混合式互置運算

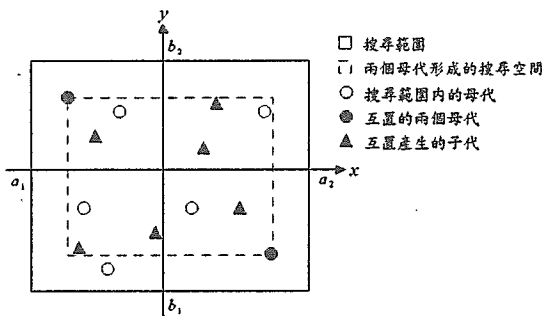
由於GA族群的組成是以在既定之參數範圍內，以隨機的方式產生，倘若族群個體間差異太小(即集中於部份區域內)，而且只採用一種互置法則，很容易使系統發生過早收斂的現象。這是因為在差異性小的個體間作基因的互換，並不容易產生與母代的特性相差較大的子代。就單點式的互置運算子，當兩母代間的差異不大時，將使得產生的子代就僅是母代特性的延續，所以新的子代不易跳脫原來母代所形成的超空間(Hyperspace)。因此欲防止GA陷於區域的極點，藉由互置的運算子的改變及適度的延伸子代的搜尋空間，必將有助GA演化的效率。以下將就本文所選用的二種互置方法(Arithmetical互置[9]和BLX- $\alpha$ 互置[10]混合使用)，及其配合AGA的 $P_c/P_m$ 所作的修正說明如下。

### Arithmetical Crossover

Arithmetical互置主要是將互置的母代取其加權比重產生新的子代。假設 $X = (x_1, x_2, \dots, x_n)$ 與 $Y = (y_1, y_2, \dots, y_n)$ 為欲互置的兩個母代，則產生後的子代為 $Z^k = (z_1^k, \dots, z_n^k)$ ,  $k=1, 2$ ，如下：

$$\begin{aligned} Z^1 &= \lambda_1 X + \lambda_2 Y \\ Z^2 &= \lambda_1 Y + \lambda_2 X \end{aligned} \quad (3)$$

根據上式可知，產生的子代是母代的線性組合(子代介於兩個母代所形成的搜尋空間)，其中 $\lambda_i$ 是常數。一般而言， $\lambda_i$ 是由 $[0,1]$ 間均勻亂數所產生而且受限於條件： $\lambda_1 + \lambda_2 = 1, \lambda_1 > 0, \lambda_2 > 0$ ，所以產生的子代將會介於兩個母代之間。此外當 $\lambda_1 = \lambda_2 = 0.5$ 時，Arithmetical互置亦稱為Average互置法。圖二為母代在不同 $\lambda_i$ 的情況下經由Arithmetical互置後，其子代可能分佈的情形。



圖二、Arithmetical互置法。

上圖中 $a_i, b_i$ 為母代的搜尋範圍。就演化的理論而言，我們希望每一世代的適應值能愈來愈好，然而Arithmetical互置是以母代的線性組合(由亂數決定)，產生新的子代。若是產生的子代遺傳到適應性較差的母代的基因片段(即子代的分佈集中於適應性較差的母代附近)，則系統將很容易陷入區域性的極點或提早收斂。此外Arithmetical互置所得的子代介於母代之間，這對於多極點函數的最佳化缺乏效率，所以選擇適當的 $\lambda_i$ 及使子代能傾向於較佳母代的一方，並將有助於的最佳極值的搜尋。由AGA的公式可知，適應值較佳的個體產生的 $P_c$ 值亦較小，所以我們擬將AGA所得之 $P_c$ 值取代 $\lambda_i$ ，並將(3)式修改如下：

$$\begin{aligned} Z^1 &= (1 - P_c)X + P_c Y \\ Z^2 &= 2 * rand() * (1 - P_c)X + P_c Y \end{aligned} \quad (4)$$

上式中 $X$ 是較佳的個體， $rand()$ 是 $[0,1]$ 間均勻分佈的亂數。所以經由以上的計算， $Z^1$ 會介於兩個母代之間但比較傾向於個體 $X$ ，而 $Z^2$ 亦會向個體 $X$ 靠近並且可能分佈於 $X$ 的鄰近區域，如此的目的是要增加在個體 $X$ 附近搜尋的機率。

### BLX- $\alpha$ Crossover

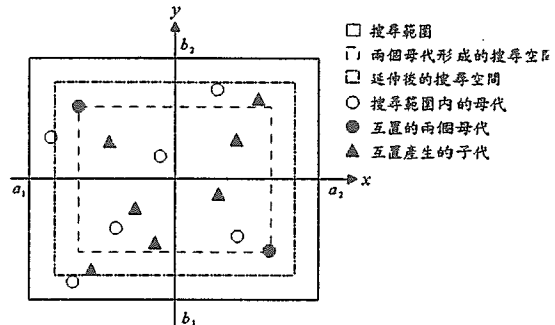
BLX- $\alpha$ 互置法是將母代所包含的區間作有條件的延展，以使產生的子代涵蓋原先母代的空間或其延伸的範圍。假設欲進行互置之兩母代分別為 $X = (x_1, x_2, \dots, x_n)$ 與 $Y = (y_1, y_2, \dots, y_n)$ ，產生的子代 $Z^k = (z_1^k, \dots, z_n^k)$ 則

$$z_i^k \in [\min_i - \alpha I, \max_i + \alpha I], \quad 0 \leq \alpha \leq 0.5 \quad (5)$$

其中

$$\begin{cases} \min_i = \min(x_i, y_i), \max_i = \max(x_i, y_i) \\ I = \max_i - \min_i \end{cases} \quad (6)$$

圖三為BLX- $\alpha$ 互置法所產生子代的分佈情形。



圖三、BLX- $\alpha$ 互置法。

BLX- $\alpha$ 互置使用固定的 $\alpha$ 值，若 $\alpha$ 太大，由於延伸的搜尋範圍過大會造成系統不易收斂。反之若 $\alpha$ 太小，則BLX- $\alpha$ 和Arithmetical互置無異。因此若能以 $P_c$ 適度地調整延伸的範圍，將會使GA的搜尋效率提高。假設 $X$ 與 $Y$ 為欲互置的二個母代， $P_c$ 為其相對應的互置機率。如果 $P_c$ 值較大表示兩個母代是屬於族群中比較差的個體，由於較差的個體本來就對族群進化的貢獻較小，因此其子代若能放棄此個體原來所在位置的特性，而置於延展的區間中，不但對於整個族群的進化不會有影響，更可以增加跳脫區域性極值的機率。當 $P_c$ 值較小時，其中一母代可能是較佳的個體，因此子代可保留母代較多好的特性，然而所延展的空間會比較小。但由於其空間仍有延展(只是比較小)，子代會比由Arithmetical互置產生的子代更具多樣性。根據以上的敘述，BLX- $\alpha$ 互置的延伸程度應與 $P_c$ 呈正比例即

$$z_i^k \in [\min_i - P_c I, \max_i + P_c I] \quad (7)$$

由以上的敘述可得知，改良後的BLX- $\alpha$ 互置法，對於個體集中於某一區域性極點附近時，會有較好的表現，但是當所有個體分散於搜尋空間時(如演化的初期、或因突變產生遠優於目前個體の子代)，會因其延展的特性，反而造成系統不易收斂，因此我們將分別取Arithmetical及BLX- $\alpha$ 互置的優點，提出混合式的互置法則。

### Hybrid Crossovers

因為在整個演化的過程中，對於目前系統的狀況，只能由個體的適應值得知，所以為判斷目前系統是否處於收斂的情況，可以經由比較上一代和目前這一代的最佳適應值，作為判斷的依據。當目前

的最佳適應值和上一代的最佳適應值相同時，表示系統可能正陷於區域性極值點附近，根據先前的敘述我們採用BLX- $\alpha$ 互置使其能有機會跳脫出區域極點，反之則使用Arithmetical互置。總結混合式互置運算的虛擬碼如下：

```

For i := 1 to Gen
  If current fitness is better than previous fitness
    use Arithmetical Crossover
  Else
    use BLX- $\alpha$  Crossover
  End
End
    
```

其中 Gen 代表演化的次數。

#### 4. 試驗結果與比較

在本節的測試我們以上述的研究方法求解LQ(Linear-Quadratic)[11]控制系統的最佳控制序列輸入。給予狀態方程式如下：

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ k &= 0, 1, \dots, N-1 \end{aligned} \quad (8)$$

我們希望求得控制輸入  $\mathbf{u}(k)=f(\mathbf{x}(k))$ ，使得系統的狀態  $\mathbf{x}(k)$  最快收斂至零，而且控制序列的輸入總和  $(\sum \mathbf{u}(k))$  是最小。所以最小化的函數可定義為：

$$J_1(\mathbf{x}, \mathbf{u}) = \mathbf{x}^T(N)\mathbf{S}\mathbf{x}(N) + \sum_{k=0}^{N-1} (\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k)) \quad (9)$$

根據 [12]，LQ問題的最佳控制輸入為  $\mathbf{u}(k)=-\mathbf{K}(k)\mathbf{x}(k)$ ，而增益  $\mathbf{K}(k)$  可由(10)、(11)式遞迴計算：

$$\mathbf{P}(k) = \mathbf{Q} + \mathbf{A}^T\mathbf{P}(k+1)\mathbf{A} - \mathbf{A}^T\mathbf{P}(k+1)\mathbf{B}[\mathbf{R} + \mathbf{B}^T\mathbf{P}(k+1)\mathbf{B}]^{-1}\mathbf{B}^T\mathbf{P}(k+1)\mathbf{A}^T \quad (10)$$

$$\mathbf{K}(k) = \mathbf{R}^{-1}\mathbf{B}^T(\mathbf{A}^T)^{-1}[\mathbf{P}(k) - \mathbf{Q}] \quad (11)$$

其中  $k=N-1, \dots, 1, 0$ ,  $\mathbf{P}(N)=\mathbf{S}$ 。最後所得(9)式的最小函數值為  $J_1^* = \mathbf{x}^T(0)\mathbf{P}(0)\mathbf{x}(0)$ 。就一維LQ的問題，(8)及(9)式可簡化為

$$x(k+1) = ax(k) + bu(k) \quad (12)$$

$$J_1(\mathbf{x}, \mathbf{u}) = sx^2(N) + \sum_{k=0}^{N-1} (qx^2(k) + ru^2(k)) \quad (13)$$

其中  $a, b, q, s, r$  為常數。本文的模擬試驗是在  $N=45, x(0)=100$  的條件下，以表一的十種情形進行

測試。經由(10)及(11)式的計算，最小的  $J_1^*(\mathbf{x}, \mathbf{u})$  整理表列於表二。

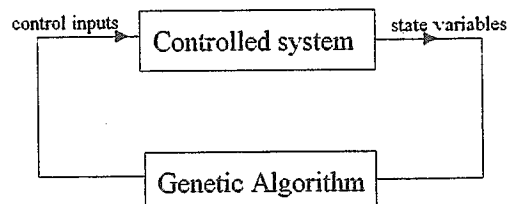
表一、一維LQ的問題的不同參數。

set	s	r	q	a	b
0	1	1	1	1	1
1	1	1	10	1	1
2	1	1	1000	1	1
3	1	10	1	1	1
4	1	1000	1	1	1
5	0	1	1	1	1
6	1000	1	1	1	1
7	1	1	1	0.01	1
8	1	1	1	1	0.01
9	1	1	1	1	100

表二、一維LQ的問題的最佳解。

set	$J_1^*$	$J_2^*$	$(J_2^*-J_1^*)/J_1^* \times 100\%$
0	1.618034e+04	1.618041e+04	4.57e-04%
1	1.091608e+05	1.091646e+05	3.52e-03%
2	1.000999e+07	1.001127e+07	1.27e-02%
3	3.701562e+04	3.702114e+04	1.49e-02%
4	2.875694e+05	2.879559e+05	1.34e-01%
5	1.618034e+04	1.619818e+04	1.10e-01%
6	1.618034e+04	1.620926e+04	1.79e-01%
7	1.000050e+04	1.000050e+04	3.1e-08%
8	4.310041e+05	4.310013e+05	1.68e-03%
9	1.000100e+04	1.017799e+04	1.79%

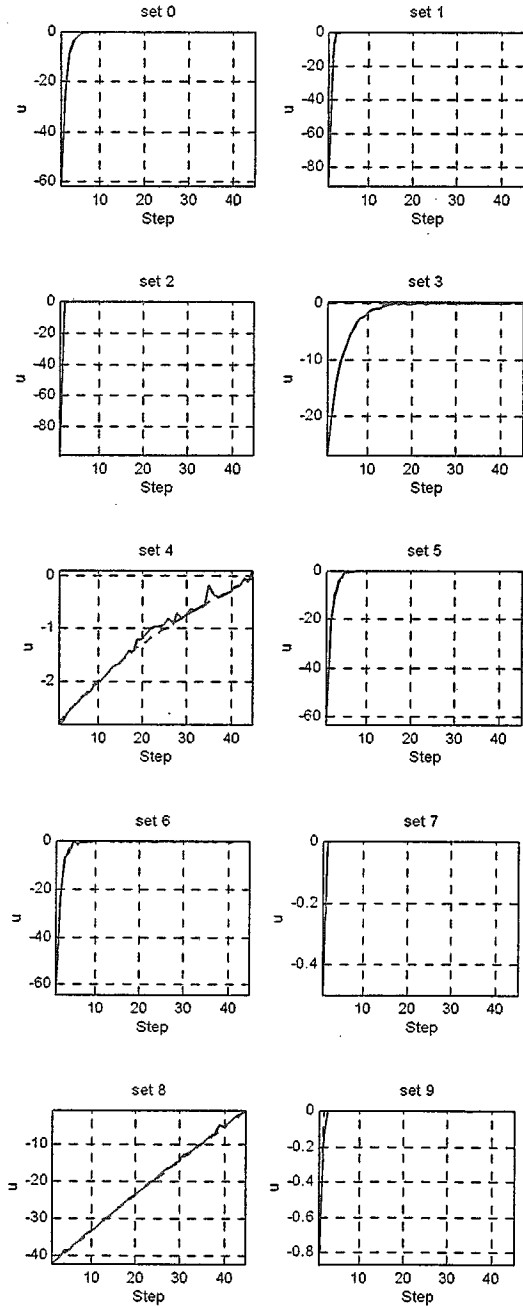
有關GA模擬試驗，受控系統的輸入將由GA演化而得，再經由狀態方程式((12)式)的更新，以(13)式為目標函數(Objective function)計算其適應值。其中適應值並將作為調整  $P/P_m$  的指標及混合式互置的依據。本系統的方塊簡圖如圖四。



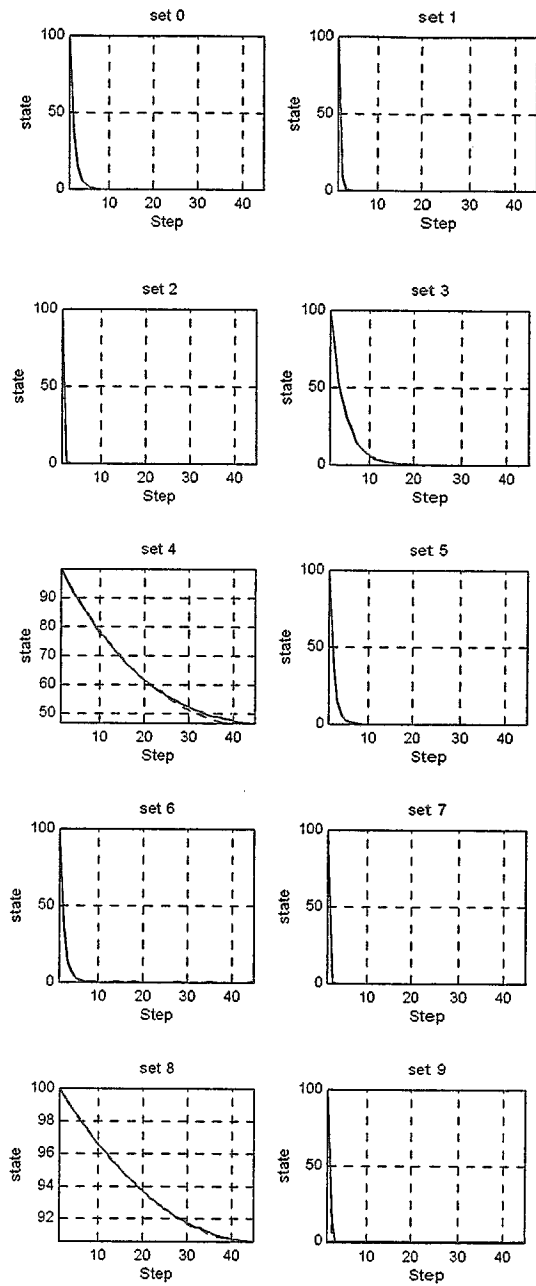
圖四、以GA演化LQ的控制輸入序列。

針對表一的十種情形，我們各以40個個體演化3000世代，以計算其控制序列的輸入。表二所列是最佳控制所得的  $J_1^*$  與GA的結果  $J_2^*$  的比較。由表二的數據可知， $J_1^*$  與  $J_2^*$  的相對誤差百分比除set 9外，GA的結果與最佳控制所得的結果相當符合。圖五

及圖六分別是在不同參數及最小化(13)式的條件下，最佳控制與GA對於受控系統的控制序列輸入與狀態變數的曲線比較。



圖五、一維LQ在不同參數條件下(表一)，控制輸入的結果比較(其中實線為最佳控制序列，虛線為GA演化的結果)。



圖六、一維LQ在不同參數條件下(表一)，狀態變數的結果比較(其中實線為最佳控制的結果，虛線為GA演化的結果)。

### 5. 結論

本文的研究提出以適應性基因演算法，及混合式互置運算法則以改進GA演化的效率，並將此一模式應用於一維LQ控制系統器的設計。實驗的數據顯示本文的結果與最佳控制計算所得的結果相當符合。這也同時顯示GA在類似問題應用上的潛力。本系統所採用的二種互置運算子並非僅限於本文所使用的類型與方法，若能對各種不同的互置運算子詳加研究，相信將必可找出更佳的組合及演化時系統考量的法則。此外對於演算法中族群數目的

多寡、演化的世代數、編碼的方式等的決定，亦是值得深入探討及日後繼續研究的方向。

## 誌謝

本文感謝國科會研究計畫 (NSC 87-2213-E-212-001)經費贊助，特此誌謝。

## References

- [1]J.J. Craig. Introduction to robotics. Addison-Wesley, 1989.
- [2]B. Kosko. Neural networks and fuzzy systems. Prentice Hall:Englewood Cliffs, N.J., 1991.
- [3]H.J. Zimmermann. Fuzzy set and its application. 3e, Kluwer Academic Publisher, 1996.
- [4]R. Battiti. First- and second order methods for learning:between steepest descent and newton's method. *Neural Computation*, Vol. 4, No. 2, pp. 301-310, 1992.
- [5]L. Davis. The handbook of genetic algorithms. Van Nostrand Reinhold, New York, 1991.
- [6]D.E. Goldberg. Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Publishing Company Inc., 1989.
- [7]M. Srinivas and L.M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. Systems, Man, and Cybernetics*, 24(4), pp. 17-26, 1994.
- [8]M. Srinivas and L.M. Patnaik. Genetic search:analysis using fitness moments. *IEEE Trans. on. Knowledge and Data Engineering*, 8(1), pp. 120-133, 1996.
- [9]F. Herrera, M. Lozano. and J.L. Verdegay. Tackling real-coded genetic algorithms: operators and tools for behavioral analysis. Dept. of Computer Science and Artificial Intelligence, 1995.
- [10]L.J. Eshelman and J.D. Schaffer. Real-coded genetic algorithms and interval schemata. *Foundation of Genetic Algorithms-2*. L.D. Whitley (Ed.), Morgan Kaufmann Publishers, 1993.
- [11]C.L. Phillips and H.T. Nagle. Digital control system analysis and design. Prentice Hall:Englewood Cliffs, N.J., 1984.
- [12]R. Bellman. Adaptive control process: A guided tour. Princeton Univ. Press, N.J., 1961.