

# 基因演算法應用於網際虛擬控制

許哲菘 研究生  
國立虎尾科技大學  
機械與機電研究所  
jss@mde.nfu.edu.tw

葉岱錦 研究生  
國立虎尾科技大學  
機械與機電研究所  
dji@mde.nfu.edu.tw

謝智翔 研究生  
國立虎尾科技大學  
機械與機電研究所  
zss@mde.nfu.edu.tw

嚴家銘 副教授  
國立虎尾科技大學  
機械設計工程系  
yen@nfu.edu.tw

## 摘要

本篇論文提出了一種具網際平台介面的 6R 機械手臂虛擬控制系統，利用實數編碼基因演算法(Real-coded Genetic Algorithms, RGA)計算各桿件的關節角度，解決逆向運動學 (Inverse Kinematics) 的議題。此合成系統透過最佳化方法，求取 6R 機械手臂端點通過路徑需求點過程，各連桿所須配合作動之關節角度，一旦系統達成適當精確度或最大世代數後，即終止運算。使用者能透過網際瀏覽器檢視所完成之機構模擬結果。在本篇論文的最後藉由兩組範例，說明本系統的操作流程。

**關鍵詞：**實數編碼基因演算法、逆向運動學、網際虛擬控制。

## Abstract

In this paper, a web-based platform for a 6R robotic manipulator simulation and control system is proposed. The inverse kinematics (IK) problems of this manipulator system were solved through the Real-coded Genetic Algorithms (RGA) to gain all the links' rotation angle functions. This synthesis of IK problem is equivalent to minimize the overall errors between the motion path of the end-effector and the targeted 3D curve due to the rotation angle combination of all the links. Once the RGA computation goal is reached according to either the precision of solution or the maximum generations, users can investigate the simulation result through the web browser. To the end of this paper, two cases were used to demonstrate the operation of this system.

**Keywords:** Real-coded Genetic Algorithms、Inverse Kinematics、Web-based Virtual Control.

## 1. 前言

在自動化控制的日益發展下，機械逐漸替

代人力，其中機械手臂在工業界的應用甚廣，它兼具高效率及工作穩定的兩大優點，在追求自動化生產的過程中扮演不可或缺的角色，例如：工作空間的軌跡控制及路徑規劃，零件組裝的定位誤差調整及力量控制等[1-4]。

在傳統解決逆向運動學的方法，大多使用向量矩陣法，利用丹赫座標表示法設定各連桿間的座標關係，透過迭代運算推導各連桿的關節角度；由於傳統矩陣法所求出的解大多不是唯一解，需要根據工作空間等條件做取捨，且過程非常繁複，而且會因為初始值的選取，造成局部解的問題[5][6]。因此，許多學者在尋找其他方法替代傳統解法，例如：利用機器視覺及最近鄰邊法[7]、路徑資料比對[8]及基因演算法；其中，基因演算法應用於逆運動學的研究廣泛，例如：利用小生境及群組改善基因演算法[5]、層級分析法結合基因演算法用於機械手臂模擬[9]、連桿設計改善力矩[10]...等，由於其泛用性，本研究採用基因演算法解決逆運動學的議題。

在實際產生機構或機器之前，會先進行虛擬作動模擬，以檢測其應用性；Java API 提供集合物件、GUI 元件、檔案處理、資料庫存取和網路等相關的類別和介面，非常適用於即時及互動的虛擬控制之介面開發[11]。早期，Java 透過 Java 3D 呈現 3D 動態模擬；由於 OpenGL 在 3D 圖形功能當中，色彩、光影、視角轉換等方面的特性，所以 Sun 公司開發 JOGL，使得利用 Java API 呈現得 3D 動態模擬更為完善[12][13]。

當控制機械手臂之連桿關節角，到達機械手臂目標點的位置函數時，稱為直接運動式；而方程式推導是透過正向運動學中，機械手臂目標點的位置與各軸關節角度的關係，透過實數基因演算法，演算出各連桿關節角，在藉由 Java 程式語言讀取各連桿關節角的數據及 \*.STL 圖檔，透果 JOGL 呈現 3D 模擬。最後，利用兩組範例，呈現模擬的結果，以證明此系統的應用。

## 2. 機械手臂運動理論分析

研究機械手臂的運動時，不僅考慮本身，亦須考慮各物體間及物體與機械手臂間的關係，如位置及方位等。因此，藉由座標的變換，用以表示這些關係，進行運動學的演算[14]。

### 2.1 連桿座標系表示法

為了要描述各連桿相對於鄰近連桿的位置，加附一物體座標系在各連桿上。連桿座標系則以所在連桿號碼命名，如圖 1 所示[15][16]。

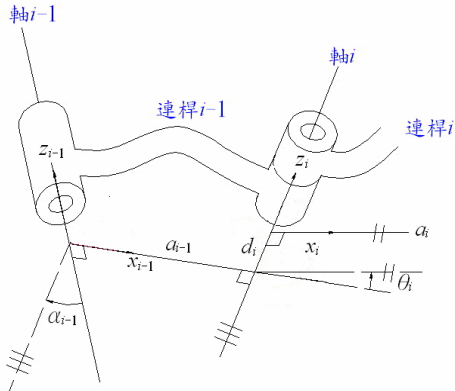


圖1. 連桿座標系{i}緊附在連桿之上

當決定座標系統時，須遵守四個原則[5][14]：

1. 座標系{B}的Z軸稱 $Z_B$ ，和關節軸B重合。
2. 座標系{B}的原點位於 $a_B$ 垂直交於關節B軸線處， $X_B$ 沿 $a_B$ 從關節處指向關節A。
3. 在 $a_B=0$ 的特殊情況下， $X_B$ 選為 $Z_B$ 和 $Z_A$ 平面的法線。
4.  $a_B$ 以右手定則繞 $X_B$ 。

依照上述原則，將連桿座標係依附在連桿後，則以下定義生效：

- ◆  $a_A$  = 從 $Z_A$ 沿 $X_A$ 量至 $Z_B$ 的距離。
- ◆  $\alpha_A$  = 從 $Z_A$ 繞 $X_A$ 量至 $Z_B$ 間的角度。
- ◆  $d_B$  = 從 $X_A$ 沿 $Z_B$ 量至 $X_B$ 的距離。
- ◆  $\theta_B$  = 從 $X_A$ 繞 $Z_B$ 量至 $X_B$ 間的角度。

為了方便求出各接頭的轉換矩陣，以逆運動學推導出和相鄰連桿座標係有關的轉換式，如圖一所示為座標系{h}、{u}、{j}之定義方式，{h}和{i-1}差一個旋轉量 $\alpha_{i-1}$ 、{u}和{h}差一個平移量 $\delta_{i-1}$ 、{j}和{u}差一個旋轉量 $\theta_i$ 、{i}和{j}差一個平移量 $d_i$ ，導出從{i-1}到{i}的轉換式如公式(1)：

$${}^{i-1}A = Rot(\hat{X}_i, \alpha_{i-1}) Trans(\hat{X}_i, \delta_{i-1}) Rot(\hat{Z}_i, \theta_i) Trans(\hat{Z}_i, d_i)$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & \delta_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

其中簡寫 c 為 cos、s 為 sin。

### 2.2 正向運動式推導

透過 2.1 小節所提的座標表示法，可設定出機械手臂各連桿間的座標關係，如圖 2、3。

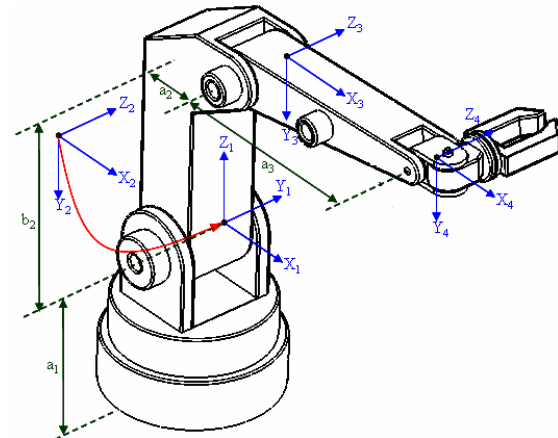


圖2. 機械手臂運動式及座標設定

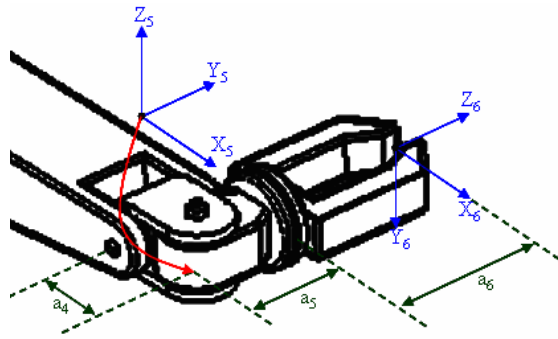


圖3. 夾爪運動式及座標設定

依據圖2、3的機械手臂座標設定，以公式(1)推導的方式，可輕易找出此機械手臂的運動式，如公式(2)：

表 1. 6R 機械手臂的連桿參數

座標系 i	$\alpha_{i-1}$	$\delta_{i-1}$	$d_i$	$\theta_i$
1	0	0	0	$\theta_1$
2	-90	0	0	$\theta_2$
3	0	$\alpha_2$	$b_2$	$\theta_3$

4	0	$\alpha_3$	0	$\theta_4$
5	90	$\alpha_4$	0	$\theta_5$
6	-90	0	$a_5 + a_6$	$\theta_6$

$$\begin{aligned}
 {}^0_1T &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -s_2 & -c_2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 {}^2_3T &= \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & b_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^3_4T = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 {}^4_5T &= \begin{bmatrix} c_5 & -s_5 & 0 & a_4 \\ 0 & 0 & 0 & 0 \\ s_5 & c_5 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2) \\
 {}^5_6T &= \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -s_6 & -c_6 & -1 & a_5 + a_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned}$$

${}^0_1A$  是地面與連桿1連接之接頭的轉換矩陣， ${}^1_2A$  是連桿1、2連接之接頭的轉換矩陣， ${}^2_3A$  是連桿2、3連接之接頭的轉換矩陣， ${}^3_4A$  是連桿3、4連接之接頭的轉換矩陣...，以此類推。其中  $s_1 = \sin \theta_1$ 、 $c_1 = \cos \theta_1$ ...，以此類推。而全部六個連桿之轉換矩陣的乘積公式，如公式(3)：

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

### 2.3 實數基因演算法應用於逆運動學

在 GA 的演算過程中，參數經過浮點數所組成稱為 RGA。RGA 不同於傳統 GA 演算時所需的編碼與解碼的動作，亦不受染色體長度的限制而影響精確度；RGA 的精確性及速度取決於參數的範圍設定，也就是說良好的參數範圍，將大幅提升其精確性及速度。

，但亦會影響求解過程的速度與精確性。

由於二進制基因演算法與RGA的優劣並無定論，演算過程中只須選用適當的演算子及輔助策略，皆可獲得良好的成效[17]。但在求

解速度的考量上，本研究是使用Prof. Noyan Turkkan 在 2004年10月所釋出的RGA模組進行求解演算，將其修改為適用於求解逆向運動學的問題，其演算程序分述如下所示：

#### (1)產生初始族群 (Initial Population)

因 RGA 採用隨機多點搜尋技巧，演算前須先產生一組初始族群 (initial population)，以便進行後續演化的運算過程。決定每代 (generation) 的族群數 (population size, PN) 後，採隨機方式產生族群內的所有個體，如果族群數夠多時，則族群內各個體應會均勻地分佈在設計空間中，可提高搜尋至全域最佳解的機會。

#### (2)適應函數(Fitness Function)

以適應函數計算出適應值後，適應值高的基因被保留下來的機率會比適應值低的基因高，因此最後保留下來的物種將會是最佳的。本研究已知條件為 1 組點軌跡，而我們利用四連桿機構輸出的點軌跡作點對點的直線距離總和取其最小值，故適應函數設計如下：

$$Fit_{min} = \sum_{i=0}^{25} \sqrt{(ox - px)^2 + (oy - py)^2 + (oz - pz)^2} \quad (4)$$

ox:經過運算的 x 座標, px: 已知路徑的 x 座標, oy:經過運算的 y 座標, py: 已知路徑的 y 座標. oz:經過運算的 z 座標, pz: 已知路徑的 z 座標.

#### (3)交配機率

由於之前有提到參數會有超界的危險，所以本研究使用的方法為 Wright 的理論[3]，Wright 提出由兩個母代產生三個子代，再踢除超界的子代，保留兩個新子代，如公式(5)：

$$X_{01} = 0.5 \times X_{p1} + 0.5 \times X_{p2} \quad (5)$$

$$X_{02} = 1.5 \times X_{p1} - 0.5 \times X_{p2}$$

$$X_{03} = -0.5 \times X_{p1} + 1.5 \times X_{p2}$$

交配頻率則由交配機率而定，機率越高搜尋最佳值的速度也越快，太低則搜尋過程容易停滯。交配率值一般設為 0.5 至 1，本研究交配率為 0.8。

#### (4)突變

突變的好處是產生新品種，避免搜尋最佳值時不會陷入局大或局小值，突變機率值一般設在 0.01 至 0.08 若定得太高 RGA 不易收斂，本研究突變率為 0.05。突變方法採用非均勻突變(Non-Uniform Mutation)：

設  $\omega = (\omega_1, \omega_2, \dots, \omega_n)$  是一個染色體，元素  $\omega^k$  被選為突變， $\omega_k \in$ ，突變結果如公式(6)、(7)：

$$\omega^{t+1} = (\omega_1, \omega_2, \dots, \omega_k, \dots, \omega_n) \quad (6)$$

$$k \in \{1, 2, \dots, n\} \quad (7)$$

其中

$$\omega_k = \begin{cases} \omega_k + \Delta(t, U_k - \omega_k), & \text{如果隨機數取 } 0; \\ \omega_k - \Delta(t, \omega_k - L_k), & \text{如果隨機數取 } 1; \end{cases}$$

這裡函數  $\Delta(t, y)$  返回  $[0, y]$  區間的一個值，取如公式(7)

$$\Delta(t, y) = y(1 - r^{(1-t)^b}) \quad (7)$$

其中  $r$  是  $[0, 1]$  區間的隨機數， $t$  是群體的當時代齡， $T$  是最大代齡， $b$  是參數。(圖 4) 為基因演算法流程圖：

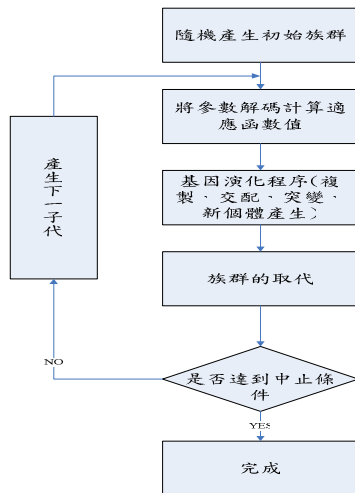


圖 4. 基因演算流程[17]

### 3. 網際動態模擬

為了能讓使用者透過瀏覽器觀看 6R 機械手臂的 3D 動態模擬結果，使用 Java Applet 及 JOGL 來達成。

#### 3.1 Java Applet

當 Java 程式碼編譯成位元組碼後，依據欲以網際網路執行，則需仰賴 Applet。

一般執行 Java 應用程式，應用程式裡有主程式 `main()` 讓 JRE 執行。Applet 與一般的 Java 應用程式不同，並沒有主程式，所以不能直接在作業系統上執行；由於 Applet 不是完整的 Java 應用程式，所以編寫模式也於其不同；Applet 的執行有一生命週期，主要是由下列四種方法組成：

- `init()`：當瀏覽程式或 `appletviewer` 載入和建立 Applet 時呼叫此方法，通知 applet 它已經加載到系統中。
- `start()`：當載入 Applet 或使用者進入含 Applet 的網頁時，呼叫此方法，Applet 開始執行。
- `stop()`：當使用者離開含 Applet 的網頁或

結束瀏覽程式時，呼叫此發法，停止 Java Applet 執行。

- `destroy()`：當離開瀏覽程式時，呼叫此方法，執行一些善後操作。

此四種方法是瀏覽程式執行 Applet 的初始、開始、停止和善後的四個過程[4]。

Applet 是透過瀏覽程式以執行，如果瀏覽程式支援 JVM 或安裝 Java Plug-in 外掛程式，便可以執行 Applet。

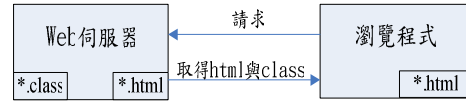


圖 5. Applet 呈現在瀏覽器上的過程[16]

如圖 2.3，在伺服端的 Applet 已編譯成 `*.class` 類別檔，在 HTML 網頁使用 `<applet>` 標籤嵌入 Applet 類別檔，當使用者瀏覽程式請求伺服端擁有 Applet 的網頁時，`*.class` 和 `*.html` 檔案都會下載到客戶端，然後在瀏覽程式執行 Applet，不過先決條件是使用端必須有該作業系統相依的 JRE。

#### 3.2 JOGL

JOGL 是 Sun 於 2003 年發布的對於 OpenGL 的 Java 類綁定，它聯繫 OpenGL 3D 圖形應用程式介面，能夠完全訪問 OpenGL 的函數，所以能達到與 OpenGL 同水準的 2D 與 3D 呈現。JOGL 亦可與 Java GUI 結合，達成圖型控制的目的。

在 OpenGL 最常使用的 GLU 與 GLUT 實用庫中的功能，Java 已可透過 JOGL 提供，例如：球體、圓柱、圓盤、鏡頭定位、圖片嵌入...等；不過，有些 OpenGL 的功能，若已 Java 便能達成，則 JOGL 便不移植此功能，如 GLUT 的視窗功能，在 Java 的 AWT 或 Swing 套件已有相關的類別庫可運用，因此並沒有移植到 JOGL 中。其中，`GLJpanel`、`GLCapabilities`、`GLAutoDrawable`、`GLCanvas` 為 JOGL 中重要的類別。

`GLJPanel` 是一個輕量級的 Swing 組件，它可以直接添加到 `JFrame`，它的用法與 `JPanel` 幾乎一樣，也因如此，`GLJPanel` 通常被慣用的 `JPanel` 取代。

`GLCapabilities` 是具有 OpenGL 繪圖功能的物件，它決定 JOGL 類別庫和 JVM 可以使用哪些 OpenGL 圖形特色。`GLAutoDrawable` 則可以支援 OpenGL 的著色支援。

GLCanvas 透過 JPanel 的包裝可以直接放在 JFrame 裡，在 JOGL 中，GLCanvas 如同畫布，而 JPanel 是畫板，JFrame 則是畫框，所有的圖形將會在 GLCanvas 上呈現。

GLCanvas 與 GLEventListener 監聽器物件通常都成對使用，這是為了響應畫布(canvas)變動及繪圖應用；除了畫布及監聽器外運用外，JOGL 圖形的呈現大部分需要一個機能用以更新畫布，這機能可透過 JOGL 的 FPSAnimator 實用類別達成。FPSAnimator 可以透過使用者設置的頻率呼叫畫布中的 display() 方法，所有的元素如圖 2.4 所示，

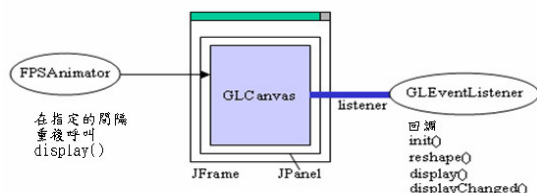


圖 6 GLJPanel 的回調[12]

GLEventListener 負責當接到 GLAutoDrawable 或 GLCanvas 的調用時，完成所有繪圖工作。它定義了四個方法：

- init(): 第一次初始化 JOGL 時，系統會調用 init() 這個方法作一些一次性的初始化工作，譬如設置線條顏色。
- reshape(): 當使用者改變窗口大小並因此而改變畫布大小時，reshape() 將做圖形比例、投影的調整。
- display(): 當開始做 JOGL 圖形呈現的動作時，會呼叫此方法以執行 OpenGL 3D 圖形系統操作；它會在 reshape() 呼叫後，重新呼叫。
- displayChanged(): 當顯示模式或顯示硬體改變時將調用此法。

本研究結合 Java Applet 的網際功能及 JOGL 的圖型顯示功能，完成一種能在網際呈現動態模擬的平台；只要透過繪圖軟體，例如：Solid-Work、Solid Edge、Inventor... 等匯出 \*.STL 圖檔規格，傳至網路空間，透過 Java 程式語言讀取圖檔的 URL 位址，便能在此平台上呈現圖型，如圖 7 所示。

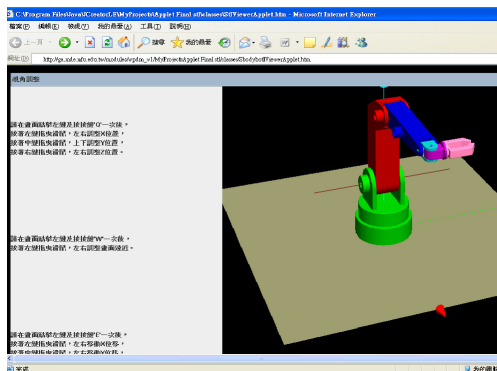


圖 7.機械手臂透過瀏覽器呈現

當讀取 RGA 演算出之到達目標點的各連桿關節角度後，使用者即可在此平台上觀看動態模擬的結果。

## 4.研究結果

本研究藉由兩個範例行審核，以證明此系統的應用。

### 4.1 範例一

本研究透過在 X-Y 平面規劃一 Cos 曲線路徑，在 X-Z 平面規劃一 Sin 曲線路徑，每一條路徑採 20 個斷點，透過 RGA 演算出各連桿的關節角度，演算處理的內定條件如下。

- (1) 族群數：2000，
- (2) 世代數：5000
- (3) 交配率：0.85
- (4) 突變率：0.05

完成演算的時間為 50 分鐘，誤差為 0.08mm，由於達到小數點第 2 位，符合準確度的標準，在 4500 代便停止運算，其誤差與時間的關係，如圖 8。

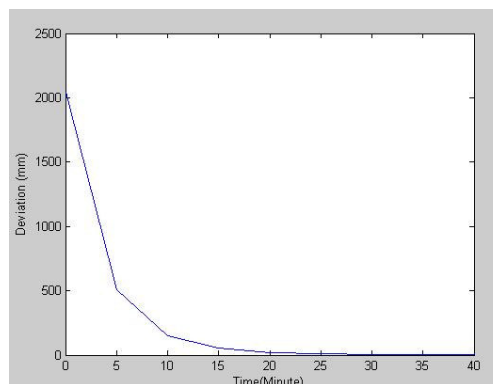


圖 8.誤差與時間圖

將 RGA 演算出的各連桿關節角度匯入平台後，其模擬結果如圖 9。



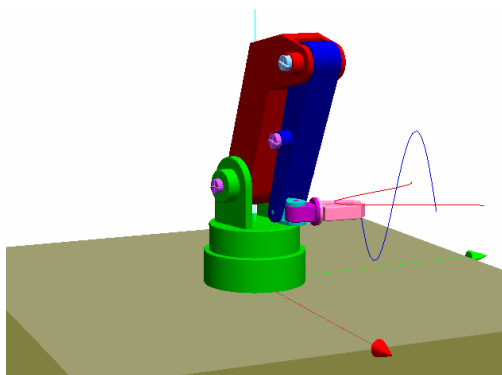


圖 9.Sin-Cos 曲線動態模擬

其中，紅色的軌跡線是 X-Y 平面的 Cos 曲線，藍色的軌跡線是 X-Z 平面的 Sin 曲線，透過此範例，可證明 RGA 可推算出在工作空間中，任一目標點的各連桿關節角度之值，平台亦可透過匯入關節角度，做準確的動態模擬。

## 4.2 範例二

此範例採用現有的文獻[20]，進行空間心臟線路徑模擬，其參數為  $r=a*(1-\sin\theta)$ ，此研究之參數  $a=200\text{mm}$  找出角度 0 到 360 度參數  $r$  值，並用算出  $r$  在  $x$ 、 $y$  座標平面上系統，之後將心臟線整個平移  $x$  方向 500mm、 $y$  方向 500mm，在心臟線平面與  $x$ 、 $y$  座標平面距離 600mm，而將心臟線平面與  $x$ 、 $y$  座標平面夾 45 度角，最後找出心臟線路徑點，顯示在圖 10

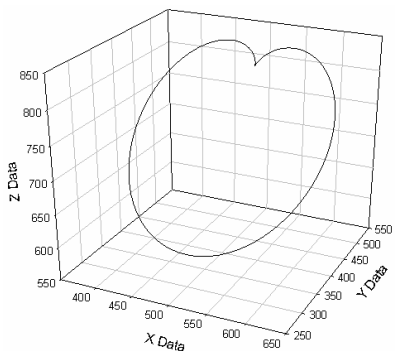


圖10.心臟線軌跡路徑(單位：mm)

將心臟線路徑點，透過 RGA 演算出各連桿的關節角度，演算處理的內定條件如下。

- (5) 族群數：2000，
- (6) 世代數：8000
- (7) 交配率：0.85
- (8) 突變率：0.05

完成演算的時間為 60 分鐘，誤差為 0.06mm，由於

達到小數點第 2 位，符合準確度的標準，在 7300 代便停止運算，其誤差與時間的關係，如圖 11

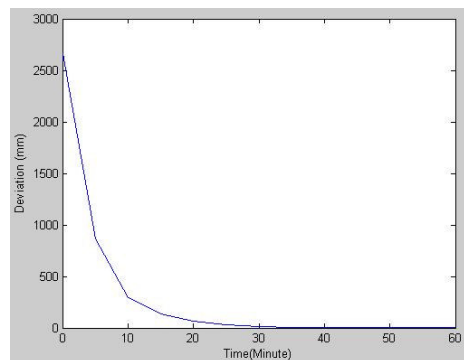


圖 11.誤差與時間圖

將 RGA 演算出的各連桿關節角度匯入平台後，其模擬結果如圖 12

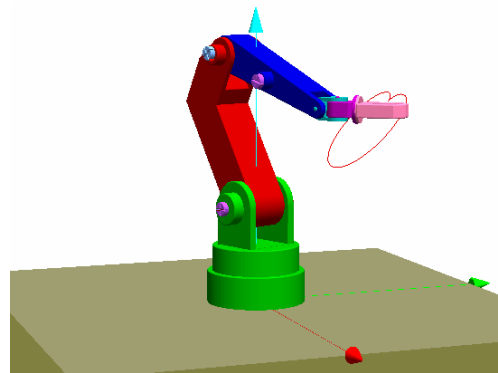


圖 12 心臟線動態模擬

由圖 12 檢視出此動態模擬系統與論文中的心臟線相符，如圖 9，因此可驗證此系統的應用性。

## 5. 結論

從範例一與範例二可檢視此系統的應用性；透過瀏覽器，可讓使用者能方便的檢視基因演算後所產生的各連桿關節角度，及其路徑軌跡的動態模擬結果。

由於許多學者早已將基因演算法應用在逆運動學上[7-9]，也證明此議題的可行性。而本研究在基因演算法的收斂時間方面，尚有改善空間，或許分散運算會是往後延伸的趨勢。

### 參考文獻

- [1] 王福壽，“六自由度混合式定位平台之設計分析與製造”，博士論文，國立成功大學，2008 年 1 月
- [2] 顏永明，“五軸 CGT 工具機之運動、工作空間與剛性分析”，碩士論文，國立中正大學，

- 2006 年 7 月。
- [3] 尤崎帆, “機械手臂運動誤差之分析與路徑規劃”, 碩士論文, 逢甲大學, 2003 年 1 月。
- [4] 董志強, “智慧型材料驅動機械手之設計規劃”, 碩士論文, 國立中央大學, 2007 年 7 月。
- [5] Saleh Tabandeh, Christopher Clark, William Melek, A Genetic Algorithm Approach to solve for Multiple Solutions of Inverse Kinematics using Adaptive Niching and Clustering, 2006 IEEE Congress on Evolutionary Computation, pp.1815-1822.
- [6] 郭俊良、王培士編譯, 機器人的機構與控制, 全華科技圖書股份有限公司, 第 72—117 頁, 1988 年 10 月。
- [7] Hannes Schulz, Lionel Ott, Jürgen Sturm, and Wolfram Burgard, Learning Kinematics from DirectSelf-Observation Using Nearest-Neighbor Methods, Advances in Robotics Research, Springer-Verlag Berlin Heidelberg, 2009, pp.11-19.
- [8] Katharina Soller, Dominik Henrich, Intuitive Robot Programming of Spatial Control Loops with Linear Movements, Advances in Robotics Research, Springer-Verlag Berlin Heidelberg, 2009, pp.147-157.
- [9] V. K. Banga, Y. Singh, and R. Kumar, Simulation of Robotic Arm using Genetic Algorithm & AHP, International Journal of Intelligent Systems and Technologies 2;2, www.waset.org Spring 2007, pp.93-98
- [10] C. Reyes and F. Gonzalez, Mechanical Design Optimization of a Walking Robot Leg Using Genetic Algorithm, Departamento de Ingeniería Industrial y de Sistemas, Universidad Nacional de Colombia, Bogotá D.C., Colombia, pp.275-284.
- [11] 李武鈺、嚴家銘、林瑞璋、郭明達, 多軸機械臂虛擬控制—使用 Java 3D, 第十九屆機械工程研討會論文集, 2002 年。
- [12] Jim X. Chen, Chunyang Chen, Foundations of 3D Graphics Programming, Springer-Verlag, London Limited, ISBN: 978-1-84800-283-8 2nd edition .
- [13] 梁志宏, 淺析虛擬實踐模的實現, 中國集體經濟下, 2007 年第 08 期。
- [14] 蔡自興, 機器人學, 北京市: 清華大學出版社, 2000 年 9 月初版, pp.29-88。
- [15] K. Abdel-Maleka, S. Othmanb, Multiple sweeping using the Denavit–Hartenberg representation method, Computer-Aided Design 31, 1999, pp.567–583
- [16] 連鼎立, 線切割放電加工機之 NC 程式製作, 碩士論文, 國立成功大學, 2006 年 6 月。
- [17] 林仕亭, 基因演算法應用於六連桿橢圓運動機之運動合成, 碩士論文, 國立中興大學機械工程所, 2008 年, pp.9-24。
- [18] Alden H. Wright, 1991, Genetic Algorithms for Real Parameter Optimization, 1st ed, pp. 1-12, Morgan Kaufmann Publishers.
- [19] 陳會安, Java 2 程式設計範例教本, 學貫行銷股份有限公司, 2004 年 11 月二版。
- [20] 魏禎佑, 開迴路撓性機械臂的逆動力學分析與控制, 碩士論文, 國立虎尾科技大學, 2008 年 5 月。